FREIE UNIVERSITÄT BERLIN
FACHBEREICH FÜR MATHEMATIK UND INFORMATIK

Aspect-Oriented Sentiment Analysis of Customer Reviews Using Distant Supervision Techniques

DISSERTATION

zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)

vorgelegt von

Dipl. Inform. Jürgen Broß

Berlin 2013

Gutachter:	Prof. Dr.-Ing. Heinz F. Schweppe Databases and Information Systems Group Institut für Informatik Freie Universität Berlin
	Prof. Dr. Artur Andrzejak Parallel and Distributed Systems Group Institut für Informatik Universität Heidelberg
Tag der Einreichung:	5. März 2013
Tag der Disputation:	11. Juli 2013

Dedicated to
the memory of my father,
Klaus F. Broß
(1945-2001)

Abstract

The opinions and experiences of other people constitute an important source of information in our everyday life. For example, we ask our friends which dentist, restaurant, or smartphone they would recommend to us. Nowadays, online customer reviews have become an invaluable resource to answer such questions. Besides helping consumers to make more informed purchase decisions, online reviews are also of great value to vendors, as they represent unsolicited and genuine customer feedback that is conveniently available at virtually no costs. However, for popular products there often exist several thousands of reviews so that manual analysis is not an option.

In this thesis, we provide a comprehensive study of how to model and automatically analyze the opinion-rich information contained in customer reviews. In particular, we consider the task of *aspect-oriented sentiment analysis*. Given a collection of review texts, the task's goal is to detect the individual product aspects reviewers have commented on and to decide whether the comments are rather positive or negative. Developing text analysis systems often involves the tedious and costly work of creating appropriate resources — for instance, labeling training corpora for machine learning methods or constructing special-purpose knowledge bases. As an overarching topic of the thesis, we examine the utility of *distant supervision techniques* to reduce the amount of required human supervision.

We focus on the two main subtasks of aspect-oriented review mining: (i) identifying relevant product aspects and (ii) determining and classifying expressions of sentiment. We consider both subtasks at two different levels of granularity, namely expression vs. sentence level. For these different levels of analysis, we experiment with dictionary-based and supervised approaches and examine several distant supervision techniques. For aspect detection at the expression level, we cast the task as a terminology extraction problem. At the sentence level, we cast the task as a multi-label text categorization problem and exploit section headings in review texts for a distant supervision approach. With regard to sentiment analysis, we present detailed studies of sentiment lexicon acquisition and sentiment polarity classification and show how pros and cons summaries of reviews can be exploited to reduce the manual effort in this context. We evaluate our approaches in detail, including insightful mistake analyses. For each of the tasks, we find significant improvements in comparison to relevant state-of-the-art methods. In general, we can show that the presented distant supervision methods successfully reduce the required amount of human supervision. Our approaches allow to gather very large amounts of labeled data — typically some orders of magnitude more data than possible with traditional annotation. We conclude that customer review mining systems can benefit from the proposed methods.

keywords: sentiment analysis, customer review mining, opinion mining, aspect-oriented review mining, distant supervision, weakly labeled data, indirect crowdsourcing

Danksagung

An dieser Stelle möchte ich all jenen Menschen danken, die mich in den verschiedenen Phasen dieser Dissertation helfend, motivierend und kritisierend unterstützt haben.

Besonderer Dank gilt:

- meinem Doktorvater Prof. Dr. Heinz Schweppe, der mich über die gesamte Promotionszeit rückhaltlos unterstützte, mir mit seiner großen Erfahrung immer hilfreich zur Seite stand und stets wusste die richtigen Fragen zu stellen. Ich möchte mich bedanken für die langen Jahre der sehr guten Zusammenarbeit in vielen interessanten Projekten, den großen Freiraum in Forschung und Lehre, sowie die stets angenehme Arbeitsatmosphäre in der gesamten Arbeitsgruppe.
- Prof. Dr. Artur Andrzejak für die Übernahme des Zweitgutachtens.
- meinen aktuellen und ehemaligen Kollegen am Lehrstuhl für Datenbanken und Informationssysteme für die schöne gemeinsame Zeit und die vielen anregenden Diskussionen. Insbesondere geht Dank an Dr. Joos-Hendrik Böse, Dr. Manuel Scholz, Dr. Katharina Hahn, Daniel Bößwetter, Sinikka Schröter, Sebastian Müller, Paras Mehta, Daniel Kressner und Marcel Drachmann. Danke auch an Heike Eckart für die stets unkomplizierte Unterstützung bei jeglichen administrativen Angelegenheiten.
- den vielen Studenten mit denen ich an verschiedensten Projekten zusammengearbeitet habe für den Innovationsgeist, die Kreativität und die immerwährende Diskussionsbereitschaft. Hier gilt besonderer Dank Alan Akbik, Heiko Ehrig, Thilo Mühlberg und Till Stolzenhain.
- Dr. René Grüneberger für die äußerst gründliche Redigierung meines Dissertationstextes. Ebenso bedanke ich mich für das Korrekturlesen bei Malte Kämpf, Paras Mehta und Marcel Drachmann. Dank geht auch an Prof. Dr. Agnès Voisard für einige wichtige Hinweise zur korrekten Verwendung des amerikanischen Englisch.
- Familie und Freunden für das Verständnis, dass die Arbeit an der Dissertation oft auch Abendstunden und Wochenenden in Beschlag nahm. Besonderer Dank gilt meiner Mutter für die Liebe und unbedingte Unterstützung in allen Lebenslagen.

Un ringraziamento particolare va alla mia ragazza Celeste. Senza il tuo amore, il tuo appoggio e la tua pazienza questo lavoro non sarebbe stato possibile. Grazie

Contents

List of Figures

List of Tables

1. Introduction

1.1. Motivation

> We had the pleasure to spend 7 days in Umbria and La Corte del Lupo. The B&B is indeed very well located if you wish to explore both the north and the south of this region. During our stay we visited Assisi, Perugia, Montefalco, Orvieto, Cortena, Gubbio and Arezzo - all located between 30 minutes and 1h 30minutes drive from Pertana. [...]
>
> Breakfast was included in our room price. Most products used are self made (choco pasta, marmelade, ...) or local products (salami, cheese). Coffee and tea as well as some juices are served, cereals too. They also serve meals in the evening. Don't mistake this with a top restaurant, but the meals are very nice and "honest" and it's 4 course including wine and coffee for only 20 € per person. [...]
>
> Service in general was very helpful - thanks Andrea & Max. The views from the garden are lovely. The place itself was a little difficult to reach, as our sat-nav didn't pick it up and some of the roads leading towards Nocera Umbra are new. [...]
>
> I would definetely recommend this place to anyone who wants to stay out of towns, in a pretty landscape but not too far from the many lovely hilltowns in Umbria.[1]

No, citing the above piece of text is not a new idea of funding a dissertation project by placing advertisements in publications. In fact, the text is an excerpt of a typical customer review found on the website of a popular travel portal. Like these, thousands of customer reviews are written and published on the Web each day. The subjects are manifold, ranging from reviews of electronic products, books, or movies to reviews of hotels or restaurants. Indeed, every ratable product or service may be addressed — for example, students also rate their professors and lecturers[2,3].

In general, opinions and experiences of other people have always been of interest for a great share of us. We ask our friends, relatives, acquaintances (people whom we trust) which dentist, hairdresser, or restaurant they would recommend us. We are interested in their opinions towards political issues and we consult professional product reviews[4] when buying a new digital camera. We normally strive for making informed decisions and our decision-making processes are often influenced by the opinions of others.

In the last decade the World Wide Web has emerged as another important source for this kind of information. People more and more tend to share their views, opinions, and experiences online — for example, by writing blogs, posting comments on a microblogging service (e.g., Twitter), using a social networking service (e.g., Facebook or Google+), publishing a product review, or commenting in discussion forums and other types of social media. Undoubtedly, an **increasing share of public discourse and popular opinion is taking place on the Web**. With regard to online customer reviews, this vast amount of unsolicited and genuine feedback represents a valuable resource for both, consumers and vendors:

As a consumer, we benefit from online customer reviews by making more informed purchase decisions. For popular products/services we have the diverse experiences of thousands of other con-

[1] Excerpt of a customer review for the agriturismo "La Corte del Lupo" by user "SuperH0liday" on tripadvisor.com (http://www.tripadvisor.com/ShowUserReviews-g666701-d1649130-r67507290 — accessed 12/2012)

[2] http://www.ratemyprofessors.com/

[3] http://www.meinprof.de/

[4] e.g., "Stiftung Warentest" in Germany or "Comsumer Reports" in the US

sumers directly at our fingertips. Nowadays, if we want to purchase a new laptop, plan our next vacation, or search for a good recipe of Tiramisù, we typically consult online reviews and ratings prior to making our decision. While we ourselves can observe this behavior, market researchers also report that online product research has become an integral part of the consumers' "purchase experience" [149, 170, 187].

As a vendor, we are naturally interested in our customers' opinions. In this context, social media in general, and online customer reviews in particular, represent an increasingly important source of information. Reviews represent genuine customer voices that primarily can help us to understand our customers' likes and dislikes. We can quickly learn about problems with our products or services and react accordingly — for example, by improving the product or adjusting our marketing campaign. We further know that customer reviews influence the opinions and ultimately may affect the purchase decision of other consumers [86, 166]. We are thus strongly interested in what customers are saying about our products or our brand as a whole. Being able to monitor and analyze social media is thus a cornerstone for implementing an online reputation management strategy[5]. Besides learning about our own customers, we may also be interested in learning what people think about our competitors' products. In general, the analysis of user-generated content in social networks, blogs, customer reviews, etc. must be regarded as an additional, important source of knowledge for business intelligence applications. For example, compared to traditional (structured) surveys the analysis of genuine, unsolicited user feedback comes with the advantage of being available in real-time at quasi no costs.

We summarize that there is a genuine **information need with regard to discovering the public and individual opinions towards a given subject** (e.g., a product, service, or brand). Traditional information retrieval systems, in particular web search engines, are of little help towards satisfying this information need. These systems cannot distinguish documents on a dimension related to opinion or sentiment. For instance, an information need such as "retrieve all documents that speak in disfavor of product X", cannot be adequately answered with a conventional Web search engine[6].

In the introductory example, we have cited an excerpt of a single review. In fact it is one of about 60 other reviews for this particular B&B on this particular travel portal. For other accommodations, located in more popular destinations, we frequently observe several thousands of reviews on a single site. On the one hand, we have vast amounts of information at our fingertips. On the other hand, **we are confronted with an increasing information overload**. For example, when searching for an appropriate accommodation, it is nearly impossible to read all the reviews of relevant hotels. Furthermore, processing even a small share of the (often contradictory) information is by itself a difficult cognitive challenge. In consequence, with the ever increasing amount of public discourse and popular opinion taking place on the Web, there is a **strong demand for automatically analyzing and summarizing the opinions expressed in natural language text**. Researchers and practitioners typically denote this task as **opinion mining** or **sentiment analysis**.

Naturally, sentiment analysis is not constrained to the analysis of customer reviews. In fact, sentiment analysis is studied and applied in very different scenarios and domains, such as political debates [23, 247, 454], financial news [48, 100], and as part of recommender systems [186, 226] or multiple-perspective question answering systems [352, 353, 363]. Recently, large-scale sentiment analysis of user comments in microblogging services (e.g., Twitter) has received increasing attention [47, 140]. A quite popular application of sentiment analysis took place during the 2012 Summer Olympics in London: A light show on the London Eye was driven by the Twitter users' sentiments about the Olympics[7].

In summary, application scenarios for sentiment analysis are manifold. Depending on the spe-

[5] see for example Beal and Strauss [33] or http://www.nytimes.com/2009/07/30/business/smallbusiness/30reputation.html

[6] Handling this type of information need is the objective of so-called sentiment retrieval systems [161, 242, 288].

[7] http://www.bbc.co.uk/news/uk-england-london-18918318

cific domain or scenario, different tasks and subtasks in sentiment analysis become important. Most obviously, the nature of textual data varies between different domains — for example, we typically observe more formal language in newswire text compared to the rather informal language or slang in microblogging posts. In consequence, the complexity of analysis, and with that, also the concrete methods for sentiment analysis may differ widely. In this thesis **we explicitly set focus on the specific application scenario of analyzing sentiment in online customer reviews**. While we provide a comprehensive overview of *review mining* in general, we are primarily interested in **aspect-oriented customer review mining**. The task's main goal is to automatically determine and assess expressions of sentiment towards individual aspects of a product. For example, we may find that most hotel guests were pleased with their room in general, but many complained about the slow Wi-Fi connection and the loud air conditioning system. A system capable of such a fine-grained analysis allows to generate a detailed summary of the customers' opinions and thus can help to alleviate the information overload we outlined earlier.

Regarding the most common approaches to sentiment analysis, we may differentiate between supervised machine learning and rule/dictionary-based approaches. In either case a great amount of manual effort is necessary to build adequate systems. Whereas supervised machine learning approaches typically involve the expensive task of creating labeled training corpora, rule/dictionary-based systems rely on the availability of comprehensive lexicons and manual fine-tuning of rule sets. As an overarching topic of this thesis, we will **examine the utility of weakly labeled data for reducing the costs involved with training machine learning models or creating lexical resources**. Weakly labeled data refers to training data where the class labels were determined heuristically and not by human supervision. We will show that user-generated content (e.g., customer reviews) often represents an appropriate data source for extracting weakly labeled training corpora. In the ideal case, the benefits with weakly labeled data are twofold: First, we can reduce costs of acquiring training data to nearly zero and thus can more easily scale our applications to other domains or languages. Second, the size of weakly labeled corpora frequently exceeds the size of conventional, manually labeled training corpora by some orders of magnitude. We can thus hope to build more accurate machine learning models.

1.2. Problem Statement and Contributions

The main goals of this thesis can be summarized as follows: We primarily examine the task of aspect-oriented customer review mining. Given a collection of review documents, the goal is to algorithmically detect and analyze all expressions of sentiment towards the different aspects of a reviewed product or service. This problem setting involves mainly two subtasks.

- Product aspects: Given a specific type of product or service (e.g., digital cameras or hotels), we want to automatically derive the most relevant product aspects for this particular type. Which aspects characterize a product? Which aspects are most commonly discussed in customer reviews of this product (e.g., picture quality, battery life, ease of use)? Knowing the relevant product aspects, we must further develop methods to detect mentions of them in natural language text.

- Sentiment expressions: Reviewers refer to product aspects in different contexts. They may use factual language and simply describe some aspects (e.g., "the camera has a 3x optical zoom") or they may express their opinion towards an aspect (e.g., "the 3x optical zoom works perfectly"). We are primarily interested in the latter case. Our goal is to automatically detect expressions of sentiment in customer reviews. We further aim at analyzing the polarity of these expressions. We want to know whether an utterance is predominantly positive (e.g., "works perfectly") or negative ("is totally crap").

In this thesis, we address both subtasks on different levels of granularity. We consider a fine-grained analysis on expression/phrase level as well as a more coarse-grained analysis on paragraph or sentence level. For both tasks we examine dictionary-based as well as machine learning approaches. As an overarching topic, we will study the utility of weakly labeled data for each of the different subtasks and methodologies. It is not our ambition to build and describe a fully functional, production ready review mining system. Few or no additional insights would be obtained by implementing such a system. Instead, we put emphasis on directly studying, implementing, and evaluating the relevant subtasks/subcomponents of such a system. In detail, our contributions are as follows:

1.2.1. Models and Text Corpora

- As a starting point, we provide a detailed analysis about how sentiments are expressed in customer reviews at different levels of granularity. We develop a *discourse oriented model* that can be implemented at the sentence or paragraph level. We further devise an *expression level model* that addresses the phrase level. We compare our models with various other models found in the literature.

- We implement both models by creating precise annotation schemes and guidelines. Based on this, we create four different, manually annotated text corpora. In particular, we consider the relatively diverse domains of hotel and digital camera reviews. For each application domain, we annotate two corpora (i.e., according to the two different levels of granularity). Each corpus comprises several hundred annotated customer reviews.

- The text corpora primarily serve us for evaluation purposes, but also help us to understand the problem setting more thoroughly. We present a detailed corpus analysis that provides insight into which linguistic phenomena are relevant and need to be tackled when performing sentiment analysis in customer reviews.

- We present a hierarchical model (denoted as *product type taxonomy*) that allows to integrate product aspects at different levels of granularity. The model distinguishes a *mention level* and a *concept level* and allows to encode *semantic relations* (e.g., "part-of" or "type-of") between different aspects.

1.2.2. Product Aspect Extraction

We distinguish between fine-grained product aspect detection at the mention level and coarse-grained detection at the concept level.

Fine-grained Analysis (Expression Level)

- For fine-grained analysis, we cast the task of determining a set of relevant product aspects as an instance of a *terminology extraction* problem. Given a collection of reviews for a specific product type, we automatically (unsupervised) generate a dictionary of associated product aspects. We propose a pipeline architecture that involves steps such as candidate acquisition, candidate filtering, variant aggregation, candidate ranking, and candidate selection. The components for each step are exchangeable. We propose and experiment with varying algorithms (e.g., different ranking metrics or acquisition heuristics).

- We consider an approach that uses weakly labeled data to improve the candidate filtering step. We show that the data helps to increase the overall extraction accuracy.

- We examine a dictionary-based approach to detect product aspects in customer reviews on the mention level. Based on this approach, we can extrinsically evaluate the performance of our terminology extraction pipeline.
- We manually refine the automatically created dictionaries and create knowledge bases according to the *product type taxonomy* model introduced earlier. We show that manual refinement is relatively effortless and allows to further increase the performance of fine-grained aspect detection.

Coarse-grained Analysis (Sentence Level)

- Also for coarse-grained analysis, the first step is to acquire a set of domain relevant product aspects. We propose and examine an approach that is based on *probabilistic topic modeling*. Thus, instead of relying on the knowledge of domain experts, we follow a data-driven approach. We consider the method as being an integral part of a semi-automatic process where final results depend on some manual refinement. We experiment with different document representations and test varying parameter settings.
- We formalize the task of discovering coarse-grained product aspects as an instance of a hierarchical multi-label classification problem. We consider the task at the sentence level, that is, we classify sentences according to a set of predefined aspect-related topics. We compare a dictionary-based approach (utilizing our product type taxonomy) with a supervised machine learning approach. For the latter, we cast the task as a multi-label text categorization problem and experiment with varying feature sets.
- For the same task, we also consider the utility of weakly labeled data. We propose a method that extracts weakly labeled corpora by interpreting section headings as labels. We find that classifiers trained on the automatically generated corpora show a similar classification accuracy compared to classifiers trained on manually labeled data.

1.2.3. Sentiment Analysis

Fine-grained Analysis (Automatic Sentiment Lexicon Creation)

- *Sentiment lexicons* provide information about the sentiment status (e.g., the polarity or associated emotion) of individual words and phrases. Whereas the major share of existing work considers general purpose lexicons, we postulate and show that domain adapted dictionaries are better suited for sentiment analysis in customer reviews.
- We propose a fully automatic method that exploits short summaries in customer reviews (the pros and cons listings) to generate highly accurate, domain-specific sentiment lexicons. We compare our method to four other state-of-the art approaches, including two thesaurus-based, automatic approaches, as well as two handcrafted lexicons. As our method is fully automatic, it scales out well to other domains. Besides being domain-specific, the lexicon is even capable of defining polarity as a function of concrete *sentiment targets*. For example, we may have multiple entries for the word "short", e.g., "short battery life" is negative, whereas "short shutter lag" is interpreted positive.

Coarse-grained Analysis (Sentiment Polarity Classification)

- We provide a detailed analysis of supervised methods for the task of sentence level *polarity classification* in customer reviews. We consider binary (positive vs. negative) and ternary classi-

fication (positive vs. negative vs. factual/objective). We discuss and experiment with varying feature sets and machine learning configurations.

- Again, we identify the pros/cons summaries of reviews as an adequate source for weakly labeled training data. We propose highly accurate methods for extracting and filtering such data. We can show that these automatically created corpora can perfectly substitute (expensive) manually labeled training data for the task of binary polarity classification.
- The task of ternary polarity classification typically involves a *subjectivity detection* step (i.e., distinguishing between objective and subjective passages). Using the weakly labeled data, we examine an approach based on *one-class classification* with support vector machines.

1.3. Thesis Outline

We structure the thesis into four larger parts, namely Part I "Background", Part II "Models, Datasets, and Problem Analysis", Part III "Tasks and Approaches", and Part IV "Discussion" (see Fig. 1.1, which illustrates the thesis structure).

Part I covers two chapters that introduce the two main topics of this thesis: Chapter 2 provides an overview of sentiment analysis in general and customer review mining in particular. We introduce the relevant terminology in the field of sentiment analysis, discuss the main tasks and challenges in customer review mining, and present some existing systems, including commercial products. Chapter 3 gives an overview of the work with weakly labeled data. We take a holistic view and embed this topic as part of a more general question, namely how to reduce the manual effort entailed with creating labeled training copora.

Part II represents the linguistically oriented part of this thesis. Measuring and algorithmically treating the expression of opinions, require formalisms that can approximate this complex phenomenon of natural language. Part II covers our attempt to do so. Chapter 4 formalizes the expression of opinion for the domain of customer reviews. We elaborate on a coarse-grained *discourse oriented model*, as well as on a fine-grained *expression level model*. We implement both models as annotation schemes which we describe in Chapter 5. The chapter further presents basic statistics of our annotated text corpora and all other datasets we use within the thesis. Chapter 6 presents the results of an annotation study that we have conducted on the developed text corpora.

Part III addresses concrete approaches for subtasks of aspect-oriented review mining. We subdivide the part into four relatively independent chapters, each addressing a different subtask. Chapter 7 covers the subtask of identifying the relevant aspects of a given product or product type. Besides discussing various alternative approaches, we present and experiment with an approach that is based on terminology extraction techniques. Chapter 8 addresses the same subtask, but on a more coarse-grained level. We present different methods that allow to acquire and to detect the main aspect-related topics discussed in customer reviews. To initially acquire a set of relevant topics, we propose a semi-automatic process which is based on probabilistic topic modeling. To detect mentions of these topics in text, we compare dictionary-based and machine learning approaches. Chapter 9 covers the subtask of identifying sentiment expressions. We examine the utility of dictionary-based approaches for this problem setting and propose a method that allows to automatically create domain-specific, context-aware sentiment lexicons. Chapter 10 considers the subtask of detecting the sentiment polarity at the sentence level. We discuss and compare different supervised methods to classify individual sentences according to their polarity status.

Chapter 11 summarizes the thesis, discusses results, and points out some ideas for future work. Appendix A contains detailed annotation guidelines that serve as a basis for creating our text corpora. The guidelines also represent an ideal source to gain deeper insight into the shape of the corpora without being forced to examine their raw data.

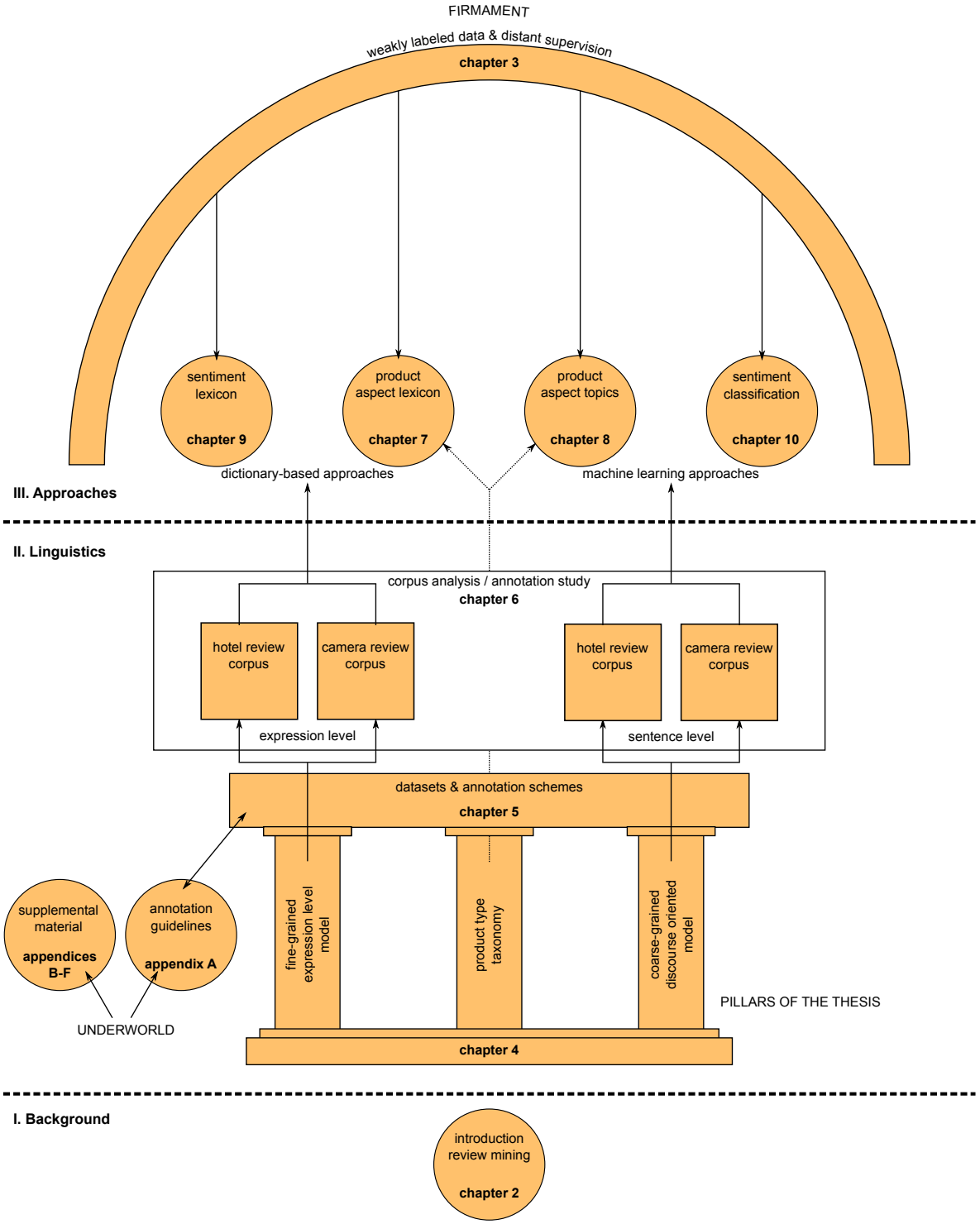


Figure 1.1.: The conception of the thesis.

Part I.

Background

Nothing exists except atoms and empty space; everything else is opinion.

Democritus

2. Sentiment Analysis and Customer Review Mining

The purpose of this chapter is to place the contributions of our thesis into a wider context. We provide a broad overview of the relevant research areas and fields of application. We discuss the most important problem settings, introduce the related terminology, and point out some basic approaches. Section 2.1 starts with an introduction to sentiment analysis and Section 2.2 provides an overview of customer review mining. Section 2.3 outlines the main research topic of this thesis, namely aspect-oriented review mining. Due to its utility for market researchers and social media analysts, sentiment analysis meanwhile caught much attention in the industry. Section 2.4 (in an exemplary manner) reviews some commercial sentiment analysis services. Subsequently, Section 2.5 summarizes the most relevant information provided in this chapter.

2.1. Sentiment Analysis

The term *sentiment analysis* describes a field of study in NLP which primary goal is to develop theories, models, and approaches to algorithmically treat the expression of opinions in natural language texts. Whereas early works in this direction date back to the 1970s/80s [63, 421, 431], sentiment analysis only recently received increasing attention of both, researchers and practitioners. Looking at the number of related publications, we find that the prominence of the research problem started to raise about a decade ago [96, 297, 390, 452, 455]. Since then, we can observe a steady increase in the number publications. For instance, a Google Scholar search for articles that mention "sentiment analysis" or "opinion mining" in the title, reveals that in 2012 alone more than 350 new articles were published. Loosely following Pang and Lee [294], the three main driving factors behind this growth of interest are presumably:

- Availability: With the rise of the "*Web 2.0*", people started to share their opinions online. Large amounts of "sentiment-laden", machine-readable data became available.
- Improved technology: Machine learning methods and tools for natural language processing improved and became accessible for a larger audience.
- Applications: With the proliferation of sentiment-laden data, researches and practitioners realized the opportunities with this kind of data (e.g., for market research or trend analysis).

2.1.1. Definitions, Tasks, and Terminology

Unfortunately, the relevant literature is quite inconsistent with regard to the use of terminology. Even key terms such as *sentiment analysis*, *opinion mining*, or *subjectivity analysis* are not consistently defined. By describing the most relevant subtasks in sentiment analysis, this section clarifies some important terms. The goal is to define a uniform terminology that we will use throughout the thesis.

Sentiment Analysis vs. Opinion Mining vs. Subjectivity Analysis

According to Pang and Lee [294, chap. 1.5] the inconsistent terminology is mainly caused by the diverse backgrounds of researchers. Whereas the term opinion mining primarily gained popularity within the communities associated with Web search and information retrieval, the terms sentiment

analysis and subjectivity analysis are more popular among NLP researchers and computational linguists. Besides these terms, we also find combinations, such as sentiment mining [453], opinion analysis [362], opinion recognition, or opinion detection [325]. We prefer the term **sentiment analysis** over all other terms to denote the research area. We believe that the term exhibits a broader coverage, better describes the different subtasks of the area, and meanwhile seems to exhibit a wider acceptance[1]. The following definition tries to capture the essence of the field of study. We heavily borrow from the description in the designated Wikipedia article [428]:

Definition 2.1 (Sentiment Analysis)**.** *Sentiment analysis is a field of study that addresses the application of NLP techniques to automatically identify and analyze* ***subjective information*** *in natural language texts. The goal is to determine the author's* ***opinion*** *about a specific* ***target****, or more abstract, about a specific* ***topic****. Subjective information may become manifest as a* ***judgment*** *or* ***evaluation****, the author's* ***affective state*** *when writing, or the affective state the author wants to evoke in the reader. The author may express his attitude on different levels of* ***granularity****, e.g., within individual text passages or as the general tone of the document. Besides pure identification, sentiment expressions are typically associated with different types of* ***semantic categories****, e.g., polarity, strength, or type of emotion.*

The length and vagueness of the definition already indicates that it is far from easy to summarize the main research problems related to sentiment analysis. In fact, sentiment analysis is a relatively heterogeneous field of study. Depending on the application scenario, the type of textual data, and the granularity of analysis, different subproblems emerge and diverse terminology is used. In the following, we present the most prominent tasks in sentiment analysis and introduce the related terminology:

Sentiment Polarity Classification

Probably the most well studied subproblem is *sentiment polarity classification*. Typically, the task is considered as a binary classification problem: Given a subjective text (e.g., a customer review or an editorial comment), the goal is to **determine whether the general tone of the text is predominantly positive or negative**. For instance, does the reviewer recommend the product or not? Does the editor support a specific viewpoint or does he[2] oppose it? We will see in Chapter 10 that many other variants of this basic task are suggested. For example, the task may be extended to an ordinal regression problem where the goal is to classify a text according to a rating scale (e.g., 1="worst" to 5="best"). We may further consider different levels of granularity — for example, document level classification vs. sentence level classification.

Obviously, a crucial point is how we define the two poles of sentiment. What is a positive opinion and what is a negative opinion? We cannot give a single answer here. A definition is much dependent on the concrete application scenario and differences may be subtle. For example, in the context of political debates, "positive" may refer to support and "negative" may refer to opposition [381]. When classifying customer reviews, the definition typically considers the evaluative nature of the text. Does the reviewer like or dislike the product? Providing a specific definition becomes even more important when computationally treating the sentiment polarity classification task. We will elaborate on this in Part II.

[1] A Google phrase search reveals that the term "sentiment analysis" produces around one million hits whereas the term "opinion mining" achieves only around 100,000 hits. A search for "subjectivity analysis" shows around 10,000 hits (queries last issued in March 2012).

[2] For reasons of simplicity and readability we will use the masculine third-person pronouns (he, him, his, himself) as generic pronouns throughout the thesis.

Subjectivity Classification

Also subjectivity classification is primarily considered as a binary classification task. The goal is to **separate subjective from objective information**. Again, the problem may be tackled at different levels of granularity. For instance, at the document level we may want to distinguish review-like documents from non-review documents [28, 396], or factual newspaper articles from editorial comments [359]. Subjectivity classification is also an important subtask in *sentiment retrieval* [161, 242, 288]. Pang and Lee [294] point out that, at the document level, the problem is quite closely related to the task of *genre classification* [104, 206]. On a more fine-grained level of analysis, the task is to identify individual text passages (e.g., paragraphs, sentences, or clauses) as being subjective or objective [295, 323, 434, 438, 455]. Fine-grained analysis may also involve the distinction between different grades of *sentiment strength* or *intensity* [436, 439]. For example, considering this dimension can be useful for automatically detecting offensive language in text [314, 397]. Very commonly, subjectivity classification is regarded as a prerequisite to sentiment polarity classification. First, subjective documents or text passages are separated from objective ones and then only the subjective documents/passages are further analyzed with regard to polarity [211, 295, 455].

As with the definition of "positive" and "negative", it is quite difficult to exactly distinguish the two classes "subjective information" and "objective information". Given a piece of text, even humans have problems in separating subjective from objective passages [386, 417]. For example, consider the following two excerpts from a hotel review: "The Wifi connection was slow." vs. "The Wifi connection was at most 16kb/s." The first statement is evaluative and we probably consider it as subjective information (i.e., another person could conceive the connection speed as satisfying or even fast). The second expression is at first sight a pure fact. However, we also know that a speed of 16kb/s indicates a weak or overloaded Wifi connection. The statement is thus intended to imply a negative evaluation. Shall we count this information as subjective? Again, an exact definition is much dependent on the concrete application scenario and type of text. We will discuss this issue in more detail in Part II.

Emotion Classification

The task of detecting the expression of emotion in natural language text can be considered as a refinement of the sentiment polarity classification task. The goal is to **classify a piece of text according to a predefined set of basic emotions**. Whereas sentiment polarity is commonly viewed as dichotomous ("positive" vs. "negative"), emotion classification tries to identify more fine-grained differences in the expression of sentiment. Most commonly, Ekman's [112, 113] six "basic" emotions, anger, disgust, fear, happiness, sadness, and surprise are used as class labels for this task [10, 11, 365]. Other theories, such as Plutchik's *wheel of emotions* with eight primary emotions [302], Scherer's *affect categories* [334], or Ekman's extended model [114] may also serve as a basis. Besides deriving a categorization from psychological theories of emotion, class labels may also be defined ad-hoc, based on concrete application needs [47, 155, 384, 446]. Applications for emotion classification are manifold, ranging from analysis of customer feedback [155] or observing trends in public mood [47] to analysis of clinical records [300]. In general, emotion classification is closely related to the research area of *affective computing* [301], which refers to the "study and development of systems and devices that can recognize, interpret, process, and simulate human affects" [427].

Sentiment Source Detection

The task of sentiment source detection aims at **identifying the person, the organization, or more general, the entity which is the source of subjective information**. For reasons of consistency, we will denote this entity as *sentiment source*, but the terms *opinion holder* or *opinion source* are also quite common in the literature [78, 79, 212, 363]. In many application scenarios (e.g., customer review mining) the sentiment source is simply the author of the text. However, the problem may be more

complex, involving nested sources of sentiment [417, 432]. For instance, newswire text often reflects different perspectives of distinct sentiment sources (including the author). Wiebe et al. [417] provide a good example:

> The *Foreign Ministry* said Thursday that it was "surprised, to put it mildly" by the *U.S. State Department's* criticism of Russia's human rights record and objected in particular to the "odious" section on Chechnya.

The sentence contains three sentiment sources (including the author). The first source is the (Russian) Foreign Ministry which is "surprised, to put it mildly" and which "objected in particular the odious section on Chechnya". The second source is the U.S. State Department which criticizes Russia's human rights record. Implicitly, the author is also a (potential) source of sentiment. Opposed to the earlier mentioned classification problems, determining sentiment sources is predominantly regarded as an *information extraction* task [36, 78, 211, 424]. It involves subproblems such as *named entity recognition* and *relationship extraction*. A typical application for sentiment source detection is for example a *multi-perspective question answering* system [363, 437, 455] that tries to answer questions of the form "What is X's viewpoint/opinion on topic Y?".

Sentiment Target Detection

As the name suggests, the goal of sentiment target detection is to **determine the subject of a sentiment expression**. Depending on the granularity of analysis, a sentiment target may refer to a concrete entity or to a more abstract topic. For instance, in aspect-oriented review mining we are interested in determining the reviewers' evaluations of very concrete aspects. Such targets typically become manifest at phrase or sentence level (e.g., "I really like the *picture quality*."). In this case, the task is primarily regarded as an information extraction task [182, 184, 207, 216, 353, 468] and it involves subproblems such as named entity recognition and relationship extraction. In contrast, *sentiment retrieval systems* are generally concerned with identifying opinions related to more abstract topics (e.g., "Which blogs report positively, which negatively on the topic of *Israeli settlement policy*?"). Such an analysis is normally conducted at the document level. At coarser-grained levels of analysis (e.g., document or sentence level), sentiment target detection is mostly viewed as an instance of a *text categorization* or, more general, as a problem in *information retrieval*. Sentences or documents are classified or ranked according to their relevance towards a given topic [40, 111, 259, 460].

2.2. Customer Review Mining

As people increasingly tend to share their views, opinions and experiences online, vast amounts of customer feedback data are easily available. For companies or market researchers such genuine customer feedback represents an extremely valuable source of information. However, the data is typically *unstructured*[3] and we thus need text mining approaches to identify and interpret the relevant information. Among the diverse sources and types of online customer feedback (e.g., in the form of blog entries, comments in social networks, posts to message boards), online customer reviews naturally represent a very valuable resource. Applying sentiment analysis techniques to analyze and summarize this specific type of data is denoted as *customer review mining*.

In this section we provide an overview of customer review mining as research area and field of application. We primarily set focus on two aspects. First, we elaborate on the type of data: What kind of information is typically available in customer reviews, how are reviews structured, and which sources for online review data exist? Second, we introduce and define the most relevant subtasks

[3] In contrast to *structured data*, unstructured data does not adhere to any predefined data model, such as an XML schema definition or a (relational) database schema. Most frequently, unstructured data comes in the form of natural language text.

related to customer review mining: Which application scenarios exist and which information needs are typically formulated?

2.2.1. Sources for Online Reviews

We regard an *online review* as a piece of text which is publicly available on the Web and which primary purpose is to evaluate a ratable entity (e.g., a product or service). Sources for online reviews are manifold. In the following we briefly describe the three most popular sources, namely *review sites*, *online shopping sites*, and *web logs*.

- **Review site**: A review site is a website which main purpose is to gather and publish reviews at a single place. The reviews may be authored by *professional critics* (e.g., http://reviews.cnet.com) or the reviews are based on *user-generated content* — that is, they are based on genuine customer experiences (e.g., http://www.buzzillions.com). We denote the former type as *expert review site* and the latter as *consumer* or *customer review site*. Review sites may be further distinguished by their *scope*. Some sites collect reviews for a single product type only (e.g., digital cameras: http://www.dpreview.com), some set focus on a specific topic (e.g., travel: http://www.tripadvisor.com), and others are more generic (e.g., http://www.epinions.com).

- **Online shopping site**: It is now common practice that e-commerce sites allow users to review the items they offer in their store. The primary motivation for this integration of online reviews is to improve the overall "shopping experience" and thus to attract more customers. Prominent examples are the online retailers Amazon.com and Ebay.com, which both provide millions of customer reviews on their websites. Online shopping sites typically publish consumer reviews (instead of expert reviews).

- **Web log**: The "blogosphere"[4] represents another source for publicly available reviews. Bloggers comment on their newly purchased products (e.g., a mobile phone), on a recent experience with some sort of service (e.g. a restaurant or hotel), or on a brand as a whole. Besides personal blogs that may occasionally contain review-like postings, specialized review blogs exist (e.g., http://www.photographyblog.com/reviews/ or http://dinersjournal.blogs.nytimes.com/). Whereas personal blogs typically represent genuine consumer opinions, specialized blogs are most often authored by professional critics.

Throughout the thesis, **we set focus on analyzing customer reviews** and do not consider the analysis of expert reviews. Whereas expert reviews have the advantage of providing a very detailed and profound analysis, consumer reviews represent genuine customer experiences. Knowing how customers perceive a product, service, or brand, constitutes an indispensable information for vendors (and also for potential customers). Customer reviews are typically shorter and less elaborated than expert reviews. Also the quality and helpfulness of individual consumer reviews may be questionable. But on the other hand, the amount of customer voices usually exceeds the number of publicly available expert reviews by far. With an automatized analysis (i.e., a customer review mining system), the sheer amount of the diverse customer voices can compensate for the reduced degree of detail or lower quality of individual reviews.

[4] "blogosphere: all of the blogs on the Internet as a collective whole" (Merriam-Webster.com. 2012. http://www.merriam-webster.com/dictionary/blogosphere)

2.2.2. Formats of Online Reviews

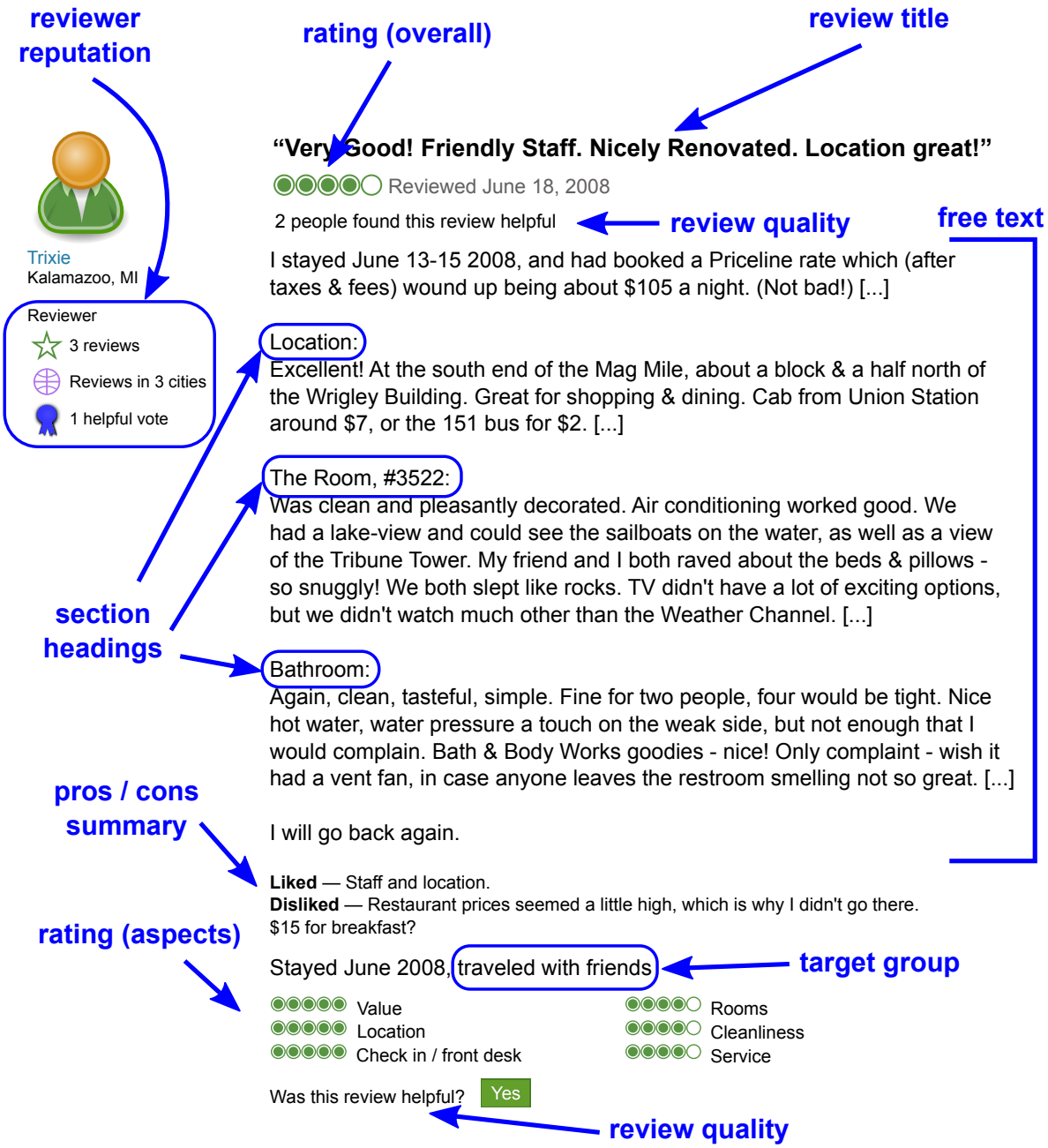


Figure 2.1.: The layout of a typical online customer review[5].

The format of online reviews typically differs between varying sources. Whereas blogs do not adhere to any predefined structure and are mostly plain text, review sites exhibit much more structure. Most review and online shopping sites force their users to use predefined web forms to publish their reviews. Users fill in fields such as "review title", "your overall rating", "your review", "pros/cons", etc. Fig. 2.1 illustrates the structure of a typical online customer review. We explicitly highlighted the most important parts of a review:

[5]Excerpt of an authentic customer review found on Tripadvisor.com (http://www.tripadvisor.com/ShowUserReviews-g35805-d87592-r17020381).

- **Review title**: Reviewers are encouraged to provide a descriptive heading of their review. As in our example, a review title often indicates the overall tone of a review and frequently serves as a brief summary of the most important impressions[6].

- **Overall rating**: The overall rating refers to a single score that represents the reviewer's overall impression. Commonly, a five-star, ordinal rating scale is used, where one star translates to a very negative assessment and five stars stand for a very positive rating (five-star scales are for instance used by Amazon.com, Ebay.com, or Epinions.com). Another common form is a ten-point rating scale, as for example used by IMDb.com or Reevoo.com. Three or four-point scales occur rarely.

- **Aspect rating**: Besides providing an overall rating, some customer review sites allow to rate individual aspects of a product. For instance, Tripadvisor.com encourages the user to explicitly rate aspects such as "location", "service", "cleanliness", etc. when reviewing a hotel. Since the relevant aspects depend on the reviewed product, individual aspect ratings are more common for review sites with a clearly defined scope. Other sites may allow to rate aspects of selected product types only. For example, Ebay.com defines ratable aspects for many electronic products (e.g., mp3 players, digital cameras, or tablet computers), but not for other products such as movies, shoes, or books.

- **Review text**: This is the part where the reviewer expresses his thoughts and opinions about the reviewed product. As this part is typically represented as a free text field in the corresponding web form, we will also refer to the review text as *free text* or *free text part*. Usually, review sites limit the length of the free text — for example, Amazon.com limits the maximum number of characters to 50,000 (Ebay.com restricts the length to only 3,500 characters). Some sites allow to use HTML markup within the review text. Just like other prose, authors often structure longer review texts with paragraphs and section headings. For instance, in our example, the paragraphs and headings correspond to different ratable aspects (location, room, and bathroom) of the reviewed hotel.

- **Pros and cons**: Reviewers summarize the main advantages and disadvantages of a product in this section. Typically, the pros and cons are structurally separated so that readers (but also machines, e.g., a *web scraping*[7] tool) can easily differentiate between positive and negative comments. The style and length of the summaries may differ between individual authors and review sites. Some reviewers simply enumerate the main positive and negative aspects (e.g., like the pros in our example), others go into more detail and formulate complete sentences. Some review sites (e.g., Reevoo.com) only provide the pros and cons as free text fields and do not offer a separate review text section. In this case, pros and cons are in tendency longer and contain complete sentences.

- **Review quality**: Most review sites encourage users to rate the quality of individual reviews. Such a procedure allows to rank the reviews by their helpfulness. For instance, Amazon.com uses this rating to display the most helpful favorable and most helpful critical review on the front page of a product review listing.

- **Reviewer reputation**: Statistics about individual reviewers such as the number of authored reviews or the amount of helpful votes establish the reviewer's reputation. We may use this data to reason about the trustworthiness of a particular review and exploit this knowledge as a further ranking measure.

[6] For instance, reviews.ebay.com demands that the review title should "summarize your experience with this product in one sentence" (description of the respective field in the web form).

[7] The term *web scraping* denotes the task of (automatic) information extraction from websites.

- **Target group**: Some review sites (e.g., Tripadvisor.com) allow to define a target group. It may refer to the intended audience of the review and/or to a self description of the author. In our example, the author classifies himself as traveling with friends (instead of traveling as a couple, alone, with children, etc.). Such a classification represents another criterion to estimate the helpfulness of individual reviews.

The most essential and informative part of a review is the part where the author can freely express his thoughts and opinions. We regard this this part as obligatory for an online customer review[8]. Other parts (e.g., the aspect ratings or the review helpfulness) can be considered as a form of metadata. Since the overall rating represents the most valuable type of metadata, it is typically considered as obligatory. Based on the author's options with regard to the free text fields, we can distinguish different types of reviews. In particular, most reviews can be attributed to one of the three formats depicted in Fig. 2.2:

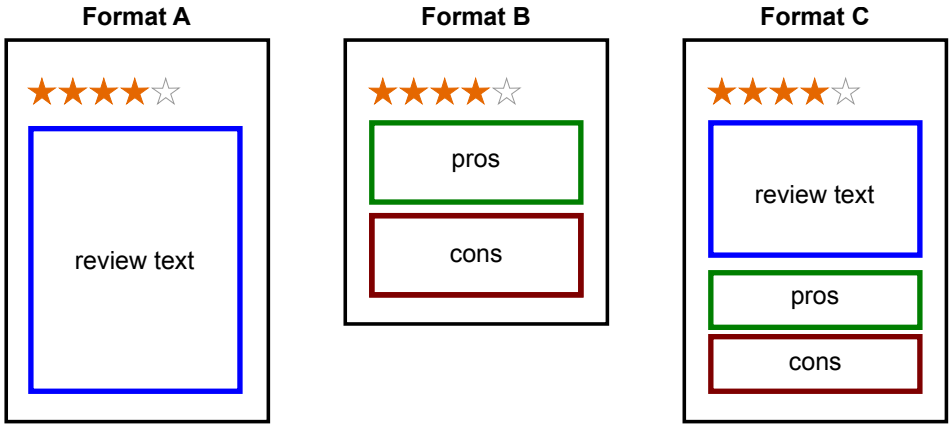


Figure 2.2.: The three most popular formats for customer reviews.

- **Format A (single review text)** presents a single free text field where authors can comment on the product. A prominent example for this format is Amazon.com.
- **Format B (separate pros and cons)** presents two separate free text fields for comments. Reviewers are forced to make an explicit distinction between positive and negative comments. This format is for instance used by Reevoo.com.
- **Format C (review text and pros/cons)** represents a combination of the two previous formats. In addition to the main review text, the format encourages a reviewer to summarize the main positive and negative points in separate sections. For example, reviews on Epinions.com or Priceline.com adhere to this scheme.

2.2.3. Specifics of the Application Domain

Compared to other fields of application for sentiment analysis, such as the analysis of newswire text, the domain of customer reviews exhibits some beneficial properties: The most obvious characteristic is that **a customer review is per se subjective**. By definition, its primary purpose is to evaluate the entity under consideration (e.g., a product or service). Subjectivity classification is thus primarily relevant for more fine-grained analysis (e.g., on the sentence or sub-sentence level), but not for document level analysis[9].

[8] Since we are interested in extracting customer sentiments from text, we do not consider "reviews" that solely consist of a single, numerical rating score.

[9] Nonetheless, subjectivity classification at the sentence level may improve the accuracy of document level polarity classification [295].

A further beneficial property is that reviews usually only reflect the perspective of the author. In contrast, newswire text often reflects different perspectives from distinct sentiment sources (cf., Section 2.1.1). Such a complexity, potentially comprising multiple nested sentiment sources, is typically absent in customer reviews. Even if other sources than the author appear in a review (e.g., "My sister complained about the short battery life of the camera."), it is **not relevant to distinguish different sentiment sources**. For instance, it is irrelevant whether the reviewer or his sister complains about the short battery life. What is relevant, is the mere fact that a complaint about the battery life exists at all. This is in sharp contrast to analyzing newswire documents, for instance in the context of multi-perspective question answering [37, 52, 412]. This task explicitly aims at determining the diverse viewpoints of different sentiment sources.

We already pointed out that customer reviews typically come with machine-readable metadata. Most importantly, **the primary subject of the review is known**. We know whether a review evaluates product X, product Y, or product Z. In contrast, when analyzing a collection of blogs, it is unclear which blog entry discusses a particular product. In addition, customer review sites enforce that a reviewer concentrates on evaluating a single product. Comparative reviews (which are more difficult to analyze) occur relatively seldom on customer review sites.

In general, when dealing with review documents of a specific product class (e.g., digital cameras) or type of service (e.g., hotels), the related **text corpus exhibits a higher homogeneity**. Evaluating a specific product type is much more concrete than discussing abstract topics, such as the latest development in the monetary policy of the U.S. Federal Reserve or the situation of human rights policy in Russia. Reviewers evaluate more concrete entities, namely the product and its various aspects. The set of ratable and relevant aspects is finite and often known a priori. We can assume that the *lexical diversity*[10] with regard to aspect mentions and with regard to the expression of sentiment is lower than in other domains (e.g., newswire text).

2.2.4. Subtasks in Customer Review Mining

- **Review classification**: This task directly corresponds to the sentiment polarity classification task. The goal of review classification is to determine the **general tone of a review**. In its most simple form the task's objective refers to binary classification with the two classes "recommended" versus "not recommended". Review classification operates on the document level and can thus be considered as an instance of *text categorization*[11]. Some of the earliest and most influential studies which examine review classification are by Pang et al. [297], Turney [390], and Dave et al. [96]. As with sentiment polarity classification, the task can be extended to distinguish more fine-grained classes. For instance, classifying a review according to a five-star rating scale represents a natural extension [21, 38, 148, 154, 284, 296, 345, 348]. Obviously, this *rating inference problem* is only relevant for corpora where explicit user ratings are absent (e.g., for product reviews extracted from web logs). Review classification is mostly irrelevant when analyzing customer reviews from online shopping or dedicated review sites. The required information is simply provided as meta data and can be extracted from the website or is even available via some application programming interface (API).

- **Aspect-oriented review mining and summarization**: For many application scenarios the coarse-grained, document level classification of reviews does not provide the required level of detail. Most reviewers express both, positive and negative sentiments in a single review. A generally positive review may also contain negative statements (and vice versa for a predominantly negative review). For instance, a hotel review may praise aspects such as the friendly staff, the

[10] Lexical diversity measures the number of distinct words used in a text (see also Section 6.2).
[11] The goal of text categorization is to classify a document into a predefined set of categories based on the document's contents or topic. We refer to Manning and Schütze [249] or Sebastiani [338] for further details.

rich breakfast buffet, and the nice room decor, but criticize that the bed was too soft and the air-conditioning quite noisy. The goal of aspect-oriented review mining is to **analyze the reviewers' sentiment with regard to individual product aspects**. Given a collection of customer reviews for some specific product or product type, the two main tasks are 1) to automatically identify all relevant aspects the reviewers have commented on and 2) to categorize the individual comments according to their sentiment polarity (commonly positive vs. negative vs. neutral). Based on this extracted information, structured summaries can be constructed, which enable quantitative and qualitative analysis of a review corpus. We provide more details of the aspect-oriented review mining task in Section 2.3.

- **Review identification**: The goal of review identification is to **determine whether a given document is a review or not**. Naturally, it is only relevant when working with sources where we do not know whether a document is a review. For example, when crawling documents from the blogosphere, only a small subset of texts are actually reviews. We could also think of a generic web search engine that, for a product-related query, restricts the results to reviews only. Review identification is for example considered by Barbosa et al. [28] and Ng et al. [279]. It typically involves subtasks such as subjectivity classification (reviews are per se subjective) and classification by site structure (reviews exhibit common formats). It may further involve the subtask of detecting the identity of the product or service which is reviewed in a document. Whereas dedicated review sites provide such information as metadata, on other sites the same information must be extracted from the text. Identifying mentions of concrete products in text can be considered as an instance of *named entity recognition*[12] (NER).

- **Review helpfulness prediction**: Whereas expert reviews normally provide very profound evaluations of a product, the quality and helpfulness of some consumer reviews may be questionable. We have seen that review sites incorporate voting systems that allow users to report helpful and unhelpful reviews. Based on such explicit user feedback it is easy to rank reviews by their utility. However, it may take time to gather enough user feedback and for non-review sites (e.g., blogs) a voting system simply would not make sense. In such cases, review helpfulness prediction tries to **rate a review text according to its helpfulness and utility for other readers**. The task is typically formulated as a regression problem where features are, for instance, the review length, the number of product aspect mentions, or the review structure (e.g., the number of section headings or paragraphs). Review helpfulness prediction is for example addressed by Ghose and Ipeirotis [141], Kim et al. [213], Moghaddam et al. [265], or Zhang and Varadarajan [463].

- **Review spam detection**: Most review sites do not verify the content of the published reviews. Typically, there is no restriction and anyone (not only the true customers) can register with the site and publish a review under some pseudonym. Vendors may write very positive reviews, praising their own products, while publishing extremely critical reviews of their competitors' products. In fact, a real business exists around hiring fake-review authors[13]. The goal of review spam detection is to **counteract the proliferation of fake-reviews**. Research in this direction is for example conducted by Jindal and Liu [192, 193] or by Lim et al. [232]. Apparently, review spam detection is already applied by some review sites such as Yelp.com[14] or Google+ Local[15].

[12] see for instance Jurafsky and Martin [198, chap. 22.1]

[13] http://www.nytimes.com/2012/08/26/business/book-reviewers-for-hire-meet-a-demand-for-online-raves.html

[14] http://officialblog.yelp.com/2010/08/dont-ask-for-reviews.html

[15] http://business.time.com/2012/08/28/why-you-shouldnt-trust-positive-online-reviews-or-negative-ones-for-that-matter/

2.3. Aspect-Oriented Customer Review Mining

For many application scenarios document level review classification is too coarse-grained and does not provide the desired information. Pure classification merely helps to gather information about how many customers are generally satisfied or unsatisfied. Based on these numbers we can discover trends in the customers' perception of a product, but we do not know the exact reasons for satisfaction or dissatisfaction. We do not know what the customers like and we do not know what they dislike. Aspect-oriented review mining[16] goes one step further and analyzes the customers' sentiment with regard to individual product aspects[17].

Whereas review classification considers only a single dimension (namely "sentiment polarity"), aspect-oriented review mining involves the joined analysis of two dimensions. On one dimension we want to discover all relevant product aspects and on a second dimension we want to identify related expressions of sentiment and determine their polarity. In contrast to review classification, the task is better characterized as a **problem in information extraction** than a problem in text categorization. We basically transform the unstructured information of a review text into a structured, aspect-oriented summary. Fig. 2.3 illustrates this process.

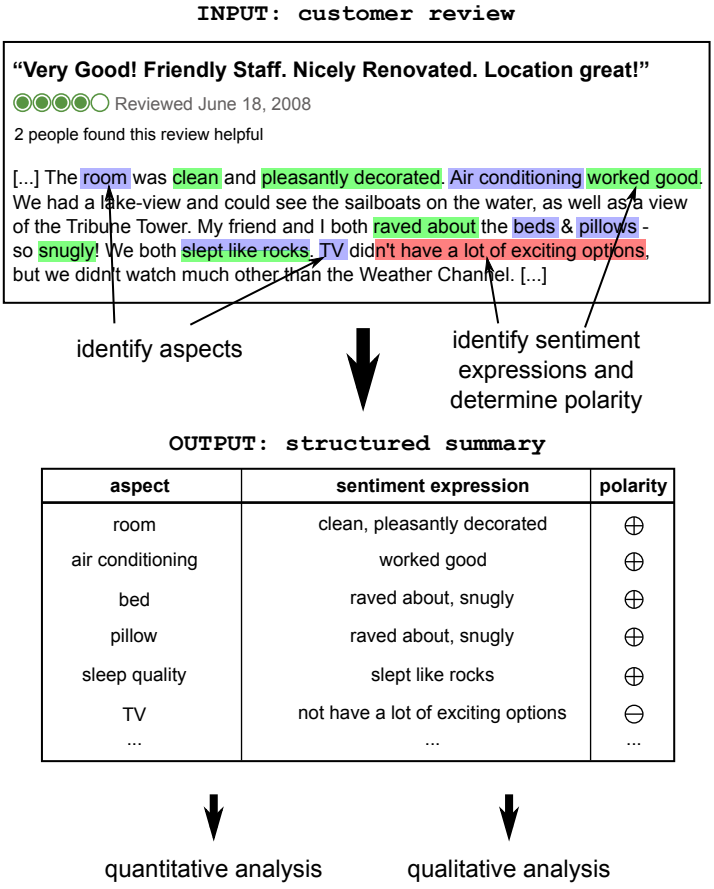


Figure 2.3.: Generating structured summaries by extracting product aspects and related sentiment expressions from unstructured customer review texts.

[16] In the literature the task is also known as *feature-based review mining* [102, 177, 304]. However, we stick to the term *aspect-oriented* throughout the thesis. Using the term *feature* may lead to confusion due to its different notion in the field of machine learning.

[17] For now, we use the term product aspect when referring to any ratable property of a product. This can be a real object (e.g., the camera lens) or a rather virtual property, such as the camera's ease of use. We will present a more elaborated model in Section 4.1.

Aspect-oriented review mining may operate on **different levels of granularity**. It can be conducted at the phrase level as for example indicated in Fig. 2.3. Alternatively, analysis may take place at the sentence or paragraph level. At the phrase level we are interested in extracting individual mentions of aspects and related expressions of sentiment. At the sentence or paragraph level we want to determine whether a whole sentence (or paragraph) expresses sentiment on a relevant product aspect. In the following we will give a brief overview of the two main subtasks of aspect-oriented review mining, namely *aspect extraction* and *sentiment polarity detection*. We will discuss the main challenges and survey basic approaches.

2.3.1. Product Aspect Extraction

The main problem in the context of aspect extraction is to identify those text passages which refer to mentions of product aspects. Given a dictionary of relevant product aspects, the task would be relatively easy. However, if the relevant product aspects are not known a priori, we need to derive them by examining the provided collection of review documents. We thus need to devise methods that automatically extract a set of the most relevant product aspects from a corpus of reviews. To do so, we have to define a notion of relevance and we must define a desired level of granularity. We may consider very fine-grained aspects (e.g., "color accuracy", "tone reproduction", "image noise", or "chromatic aberration") or we may consider more abstract concepts (e.g., "image quality", "ease of use", "battery", or "features"). Approaches to aspect extraction can be subdivided into three main classes. In particular, we differentiate between *unsupervised*, *supervised*, and *topic modeling* approaches:

- **Unsupervised approaches** are typically frequency-based and often involve the use of predefined linguistic or syntactic patterns to detect candidate phrases. Most commonly, the goal is to automatically construct a dictionary of product aspects from a given review corpus. Unsupervised approaches are for instance due to Hu and Liu [177], Popescu and Etzioni [304], Scaffidi et al. [332], Wu et al. [445], Yi et al. [452] or Su et al. [368]. Although such frequency-based methods are generally simple, they actually achieve quite good results. On the negative side, we observe that most approaches require manual tuning of parameters. Finding optimal parameter settings can be difficult as they typically dependent on the concrete dataset. We will study unsupervised approaches more closely in Chapter 7, where we will cast the task as a *terminology extraction* problem.

- With regard to **supervised approaches**, we can distinguish *sequence labeling* and *classification* methods. Sequence labeling techniques, such as *hidden Markov models* (HMM) [32, 310] or *conditional random fields* (CRF) [223], are most commonly applied if aspect extraction is considered at the phrase level. HMMs are for instance used by Jin et al. [190]. CRFs are considered by Jakob and Gurevych [183]. Most approaches consider syntactic relations and rely on the output of a *natural language parser* to derive appropriate machine learning features (e.g., Li et al. [230] or Jakob and Gurevych [183]). The main advantage of supervised methods is that they are generally more accurate than unsupervised approaches. On the other hand, they require labeled training data and are thus less portable to new application domains.

  At coarser grained levels of analysis (e.g., sentence or paragraph level) traditional classification techniques are most frequently applied. Given a small list of predefined aspects (e.g., 20 items), the goal is to classify a sentence or paragraph as referring to one or more of the aspects. Classification at the sentence level is for example proposed by Ganu et al. [138] or Blair-Goldensohn et al. [40]. The former consider the use of *support vector machines* (SVM) [88][250, chap. 15], while the latter learn *maximum entropy models* (MaxEnt) [39, 281]. With regard to coarse-grained analysis, we will examine supervised methods more closely in Chapter 8. For phrase level analysis, we will not further consider supervised methods in this thesis and instead refer, for instance, to Jakob [182].

- **Probabilistic topic modeling** represents a further approach to product aspect detection and extraction. Blei [42] describes *probabilistic topic models* as "algorithms for discovering the main themes that pervade a large and otherwise unstructured collection of documents". In the context of customer reviews these "themes" ideally cover the mentioned product aspects. However, traditional topic modeling approaches also cover the different shades of sentiment as independent topics. Researchers therefore adapted existing approaches and most commonly propose to model aspects and sentiment jointly [194, 233, 259, 383, 465]. Similar to traditional unsupervised methods, topic models exhibit the advantage of not relying on labeled training corpora. We will use a topic modeling approach to automatically acquire coarse-grained product aspects. Appendix D discusses this problem setting in more detail.

Challenges

Besides the main subproblem of identifying and extracting product aspects, we are further confronted with the following challenges:

- **Relations among aspects**: Product aspects are typically related among each other. For instance, we can observe *part-of* or *type-of relations* between different aspects (e.g., a lens cover is part of a camera lens and the landscape mode is a type of digital camera mode). Depending on the application scenario, we may want to make these hierarchical relations explicit and construct some sort of *product aspect taxonomy* (refer to Section 4.1). Another common relation is similarity. It is reasonable to group similar aspects and to detect synonyms. For instance, we may want to represent aspect references such as "image quality", "picture quality", or "quality of image" by the single canonical form "image quality". Automatically grouping entities (e.g., product aspects) and determining relations between them can be considered as a problem of *ontology learning* [60, 244].

- **Implicit aspect mentions**: Given a dictionary of relevant product aspects, it is relatively easy to identify explicit mentions of those aspects in a review text. It is much more difficult to discover implicit mentions. For instance, consider the excerpt in Fig. 2.3. The phrase "slept like rocks" implicitly refers to the aspect "sleep quality". Another example would be "the camera is too heavy". Without explicitly mentioning the term, the reviewer criticizes the camera's weight.

- **Comparative reviews**: By now, we tacitly assumed that a review only refers to a single product. This is however not always true. A reviewer may evaluate a product by comparing it to other, similar products. If this is the case, we need to determine the different product entities mentioned in the text. We carefully need to inspect which expressions of sentiment relate to which product entity. This specific challenge is for example considered by Ganapathibhotla and Liu [137] or Jindal and Liu [191]. Comparative evaluations are more common in expert reviews and occur relatively seldom in customer reviews.

2.3.2. Sentiment Analysis

Regarding the dimension of sentiment analysis, the main challenge is to identify those text passages where the reviewer evaluates the product or one of its aspects and then to determine the related sentiment polarity. Similar to the detection of aspects, we can distinguish unsupervised and supervised approaches:

- **Unsupervised approaches** most commonly involve the use of a *sentiment lexicon*. Such a dictionary lists terms and phrases that are typically related to the expression of sentiment. For instance, words such as "nice", "fantastic", "love", "hate", "ugly", or "bad" would be contained in a sentiment lexicon. Each entry is associated with a value that represents its *prior sentiment*

polarity (the polarity without considering any context). Sentiment lexicons may be manually created [360, 438] or constructed via an automatic process [22, 146, 201, 241, 312, 399]. We will reconsider the automatic construction of sentiment lexicons more thoroughly in Chapter 9. Not all unsupervised approaches rely on the existence of a sentiment lexicon. For instance, Popescu and Etzioni [304] propose a method based on *relaxation labeling* [178]. Also most of the previously mentioned topic models fall into the category of unsupervised approaches.

- **Supervised approaches** try to build models of sentiment expression by learning from annotated text corpora. As with supervised aspect extraction, we can mainly distinguish between sequence labeling and classification approaches. Sequence labeling is primarily applied for analysis at the phrase level [53, 469]. Traditional classification techniques are more common for sentence or paragraph level analysis [40, 136, 210, 422], but they may also serve for sentiment analysis at the sub-sentence level [76, 438]. We will consider supervised approaches for sentence level sentiment analysis more closely in Chapter 10.

Challenges

When trying to identify expressions of sentiment in natural language text and to determine the conveyed polarity, we are primarily confronted with the following challenges:

- **Implicit sentiment**: Besides explicit expressions of sentiment (e.g., "the check-in process went fast"), sentiment may also be expressed implicitly ("needed to wait two hours to check in"). Whereas the first example includes a subjective assessment ("fast"), the second example merely expresses a fact (two hour waiting time). To infer a negative evaluation of the check-in process in the second example, we need to apply the common sense knowledge that a two hour waiting time is normally inappropriate. In the literature this form of implicit sentiment is referred to as *objective polar utterance* [433], *evaluative fact* [282], or is denoted as *polar fact* [386].
- **Contextual polarity**: The sentiment polarity of a phrase may be context dependent. For instance, consider the sentence "the hotel staff was not very friendly". The negation "not" flips the otherwise positive polarity of the word "friendly". Words, phrases, or syntactic constructions that affect the sentiment polarity or sentiment strength are commonly denoted as *sentiment* or *valence shifters* [303]. A detailed study of contextual polarity is for example conducted by Wilson [434]. We will reconsider contextual polarity and sentiment shifters in Part II.
- **Target-specific polarity**: The sentiment polarity of words and phrases may depend on the modified target. For instance, consider the adjective "long". If it modifies the product aspect "battery life", it refers to a positive evaluation. However, in the context of the aspect "flash recycle time" it would be interpreted as negative. Most sentiment lexicons ignore this phenomenon and only consider the prior polarity of words. We will address the construction of domain and target-aware sentiment lexicons in Chapter 9.

2.4. Commercial Sentiment Analysis Services

Understanding customer opinions is of great importance for any business. Due to this strong information need and the vast amount of sentiment-laden data published in the social media, sentiment analysis caught much attention in the industry. Meanwhile, many start-up companies, but also some of the established text analytics vendors offer social media analysis services that involve sentiment analysis components. Also web search engines, such as Google Product Search[18], integrate sentiment analysis into their systems (e.g., review classification/summarization).

[18] http://www.google.com/shopping/

Table 2.1 presents a comprehensive overview of some exemplary systems. Take note that the list is not intended to be exhaustive; much more companies offer similar services. The table's content is mostly based on studying the vendors' product descriptions, trying demo applications, reading white papers, or analyzing available screenshots. Only a few companies provide more details about their sentiment analysis components (e.g., in form of academic publications or patents). We thus cannot go into much details here and our observations have to be interpreted with care. The screenshot in Fig. 2.4 exemplarily shows some results obtained with a commercial sentiment analysis service.

Most commercial sentiment analysis approaches seem to rely on simple lexicon-based approaches. They simply aggregate phrase level sentiment scores and ignore more complex linguistic constructs (e.g., sentiment shifters). Some other systems apply natural language parsing and provide a rule engine to model more complex constructs. Machine learning techniques for fine-grained sentiment analysis (e.g., sequence labeling methods such as CRF or HMM) seem to be rather uncommon. We presume that one reason is the cost involved with creating appropriate training datasets.

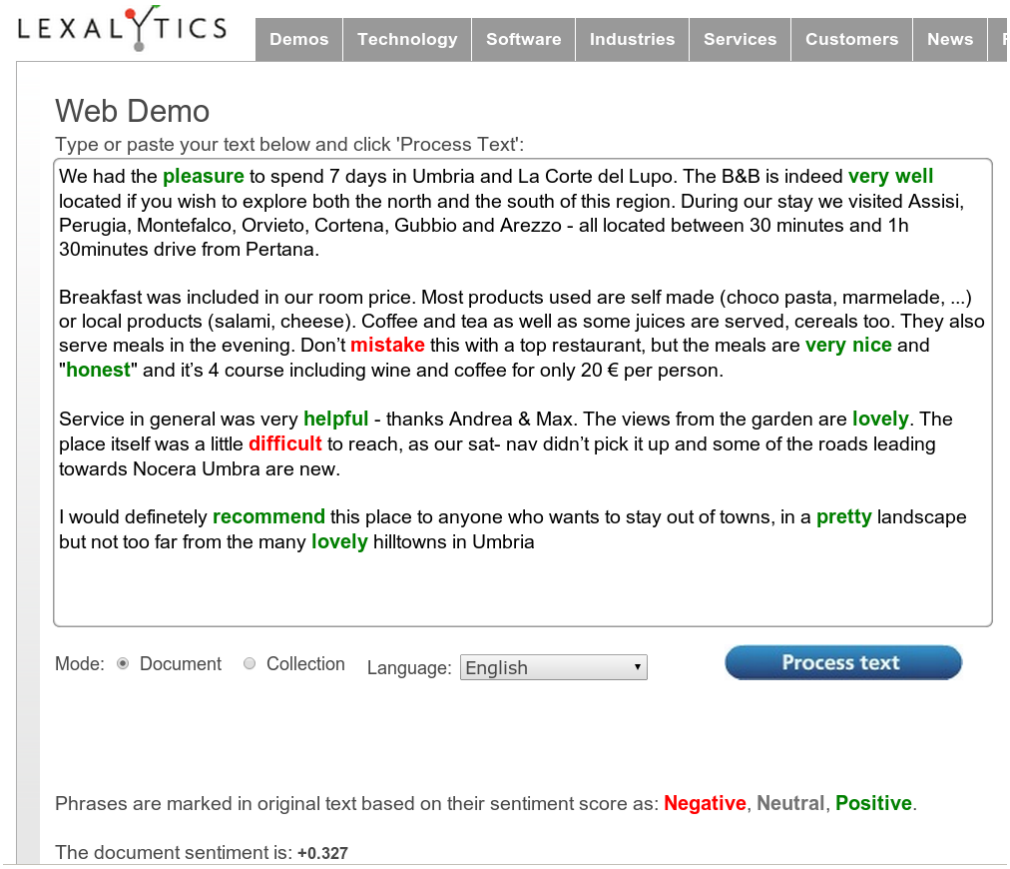


Figure 2.4.: Screenshot showing exemplary results with a commercial sentiment analysis service (Lexalytics demonstrator at http://www.lexalytics.com/web-demo). As input we provided the review excerpt from Chapter 1. The green (positive) and red (negative) highlighted phrases refer to the sentiment expressions the service could detect. The service calculated an overall score, which is depicted at the very bottom.

type of service	product name	type of analysis	approach	references
social media analysis	Attensity Analyze	phrase level polarity, entity recognition	sentiment lexicon, rule engine (?), sentiment shifter detection	[19]
″	Clarabridge	aggregated phrase level polarity scores, emotion classification	sentiment lexicon, rule engine	[80]
″	IBM Cognos Consumer Insight	phrase level brand sentiment	sentiment lexicon	[179]
″	Lexalytics Sentiment Analysis	aggregated phrase level polarity scores	sentiment lexicon	[229]
″	Lithium Social Web Analytics	phrase level brand sentiment, emotion classification	unknown	[234]
″	Salesforce Marketing Cloud	polarity classification	unknown	[330]
″	Sysomos Social Media Monitoring Dashboard	polarity classification	classification model	[372]
review mining	Google Product Search	aspect-oriented sentiment summary	sentiment lexicon, fine & coarse-grained aspect dictionary, machine learning	[40, 278, 318, 383]
toolbox / API	Alchemy API Sentiment Analysis	phrase level polarity scores, entity recognition	sentiment lexicon (?)	[8]
″	Rapid-I RapidSentilyzer	polarity classification	classification model	[313]
″	Repustate Sentiment Analysis API	clause level polarity scores	sentiment lexicon (?)	[321]
″	SAS Sentiment Analysis	aspect-oriented sentiment summary, polarity classification	sentiment lexicon, aspect taxonomy, rule engine	[331]
twitter analysis	Sentiment 140	polarity classification	classification model	[145, 342]
″	SocialMention.com	polarity classification	sentiment lexicon (?)	[350]
″	Twitrratr.com	polarity classification	sentiment lexicon (?)	[392]

Table 2.1.: Listing of exemplary commercial sentiment analysis services (it is not intended to be exhaustive).

2.5. Summary

In this chapter we placed the research topic of our thesis into a wider context. In particular, we provided a broad overview of sentiment analysis, customer review mining, and aspect-oriented review mining. In Section 2.1 we defined **sentiment analysis** as field of study that addresses the application of NLP techniques to automatically identify and analyze subjective information in natural language texts. As the main subtasks in sentiment analysis, we introduced *sentiment polarity classification* (positive vs. negative sentiment), *subjectivity classification* (opinion vs. fact), and *emotion classification* (distinguish basic emotions expressed in a piece of text).

Section 2.2 discussed **customer review mining** as an important field of application for sentiment analysis tasks. We distinguished varying sources for online reviews (e.g., review sites vs. blogs), considered the typical elements of a customer review (e.g., overall rating, review text, pros/cons summaries), and described the most common review formats. We further pointed out some beneficial properties of the customer review domain, such as the availability of machine-readable meta data, the reduced lexical diversity of related text corpora, and the irrelevance of distinguishing between different sentiment sources. Section 2.2.4 presented the most important tasks in the context of review mining, namely *review classification* (recommended vs. not recommended), *aspect-oriented sentiment summarization* (extract and summarize sentiment polarity w.r.t. individual product aspects), *review identification* (review-like vs. non-review), *review helpfulness prediction* (rate the utility of a review), and *review spam detection* (fake review or not).

In Section 2.3 we introduced the task of **aspect-oriented review mining**. We elaborated on the two primary problem settings in this context, namely automatic detection of relevant product aspects and fine to coarse-grained sentiment polarity analysis. We pointed out the major challenges related two both subtasks and sketched basic approaches. In particular, we differentiated between *unsupervised*, *supervised*, and *topic modeling approaches*. We concluded the chapter in Section 2.4 with a brief overview of existing commercial sentiment analysis services.

The only true wisdom is in knowing you know nothing.

Socrates

3. Reducing the Costs of Human Supervision in Text Analytics Tasks

As we learned in the previous chapter, and similar to other text analytics tasks, *supervised machine learning* approaches play a major role for customer review mining (e.g., for review classification or aspect-oriented sentiment summarization). Also *special-purpose dictionaries* (e.g., sentiment lexicons) typically constitute an integral part of text analytics tasks. Unfortunately, creating such dictionaries or training machine learning models usually involves a substantial amount of manual effort. For instance, supervised machine learning requires annotated training corpora and providing such annotations is typically a manual (and therefore costly) process. Corpora or dictionaries often need to be created from scratch and tailored for a specific application domain. For example, different sentiment lexicons need to be compiled for the analysis of hotel or digital camera reviews (see also Section 9.3). Obviously, it would be of great advantage if the total costs for acquiring labeled training data or creating application specific dictionaries could be reduced to a minimum. Considering the acquisition of training corpora, the following two factors are important:

- The amount of the training data necessary to achieve a desired accuracy.
- The average effort or cost required to obtain a single annotation — that is, the *per-annotation costs*.

Tackling the first factor is predominantly a research topic in the machine learning community; keywords are *semi-supervised learning* [70] and *active learning* [285, 343]. To address the second factor, researchers in the NLP community currently study the utility of *crowdsourcing*[1] approaches; they examine whether a workforce of non-experts can adequately substitute a group of expert annotators [175, 347]. As an overarching research question of this thesis, we also address the second factor — that is, we try to reduce the per-annotation costs. But instead of examining a crowdsourcing approach, we consider another scheme for acquiring labeled/annotated data: We propose to automatically extract training data from the Web and build machine learning models based on large amounts of this *weakly labeled data* [35, 90, 264, 393]. Our hypothesis is that many websites, especially user-generated content (e.g., customer reviews), contain valuable (semantically rich) information that we can gather to construct training data for many different tasks.

We examine the utility of weakly labeled data as a recurring topic throughout the thesis. Whereas the paradigm is more generally applicable (see the next section), we particularly focus on the utility of weakly labeled data for subtasks in aspect-oriented customer review mining (in Part III, we will introduce and experiment with concrete approaches for product aspect detection and sentiment analysis in customer reviews).

The purpose of this chapter is to briefly survey the topic of "weakly labeled data" and to place the concept into a wider context, namely reducing the costs of human supervision in text analytics tasks. The remainder of the chapter is organized as follows: Section 3.1 comprises the main part of this chapter. We discuss how weakly labeled data can be used to reduce labeling costs and we introduce the closely related concept of *distant supervision* (which we will also denote as *indirect crowdsourcing*).

[1] The term crowdsourcing, which is combination of the words crowd and outsourcing, was introduced by Jeff Howe in a 2006 Wired Magazine article [173]. Howe [172] defines crowdsourcing as "the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call". The Wikipedia defines it as "a process that involves outsourcing tasks to a distributed group of people", which "can occur both online and off-line" [426].

As mentioned earlier, researchers mainly consider two other paradigms for reducing labeling costs. Section 3.2 briefly discusses crowdsourcing approaches and Section 3.3 briefly reviews the most important concepts developed in the machine learning community.

3.1. Weakly Labeled Data, Distant Supervision, and Indirect Crowdsourcing

To grasp the basic idea of exploiting weakly labeled data, consider the following example: One of the largest sources for movie reviews is probably the Internet Movie Database (IMDb)[2]. Apart from professional critics, also users of the site can write movie reviews. As these review texts are typically accompanied by numerical rating, it is straightforward to use them as training data for a *sentiment classification* task: Each review text represents a sample which is implicitly labeled by the user-provided rating. We can easily devise a heuristic that aligns review texts (samples) and ratings (labels), so that we can extract millions of these tuples to construct a huge training corpus. In the most basic case, we would train a classifier that separates movie review texts according to the general tone of the document (i.e., positive vs. negative). Exemplary studies, which build sentiment classifiers with weakly labeled training from IMDb.com, report high classification accuracies of over 86% [295, 297].

Abstracting from this concrete example, the basic modus operandi of the described scheme is to employ high-precision (possibly low-recall) heuristics to automatically derive labeled training data from large semi- or unstructured datasets. Due to the lack of human supervision, such heuristically derived training data may be noisy and we therefore denote it as being **weakly labeled** or weakly annotated. Following recent publications [145, 264, 322, 370], we denote the modus operandi of automatically extracting training data as **distant supervision**. Equivalently, we also introduce the term **indirect crowdsourcing**: Similar to traditional crowdsourcing, distant supervision approaches leverage data that is collected by a huge, undefined, and distributed group of people. But in contrast to crowdsourcing, people are not directly/explicitly stimulated to generate a certain kind of data — by heuristically aligning and mapping different data sources, distant supervision *indirectly* exploits the massive workforce of the crowd. In the following, we elaborate more closely on distant supervision by studying some more examples that apply this paradigm.

Distant Supervision for Information Extraction

A very valuable and attractive source for such approaches is Wikipedia[3]. As a concrete example, the "Wikipedia-based Open Extractor" system by Wu and Weld [443] utilizes the English Wikipedia corpus for the task of *open information extraction* [6, 7, 124]. To construct a training corpus, they heuristically map the structured information from Wikipedia's *infoboxes*[4] to related text passages of the associated Wikipedia article (see Fig. 3.1). For instance, the infobox of the Stanford University article contains an attribute-value-pair $\langle established, 1891 \rangle$. A sentence of the article that contains both, the subject of the article (or a coreference to it) and the attribute value (here: 1891) (e.g., "...founded the university in 1891 ..."), indicates a natural language expression of the $\langle established\_in \rangle$ relation type and can be used as a training sample. Aggregating samples for various relation types over the whole Wikipedia yielded a training set of about 260,000 annotated sentences. In a similar fashion, Wikipedia's infoboxes are used to train *ontology learning* systems [369, 441, 442]. Mintz et al. [264] substitute the infobox information with structured knowledge extracted from the Freebase[5] semantic

[2] http://www.imdb.com/
[3] Characteristics such as Wikipedia's high coverage and diversity, its high quality [143], its factual language, and its internal structure promote this special usage.
[4] In Wikipedia, *infoboxes* are fixed-format tables that summarize unifying aspects of interrelated articles.
[5] http://www.freebase.com/

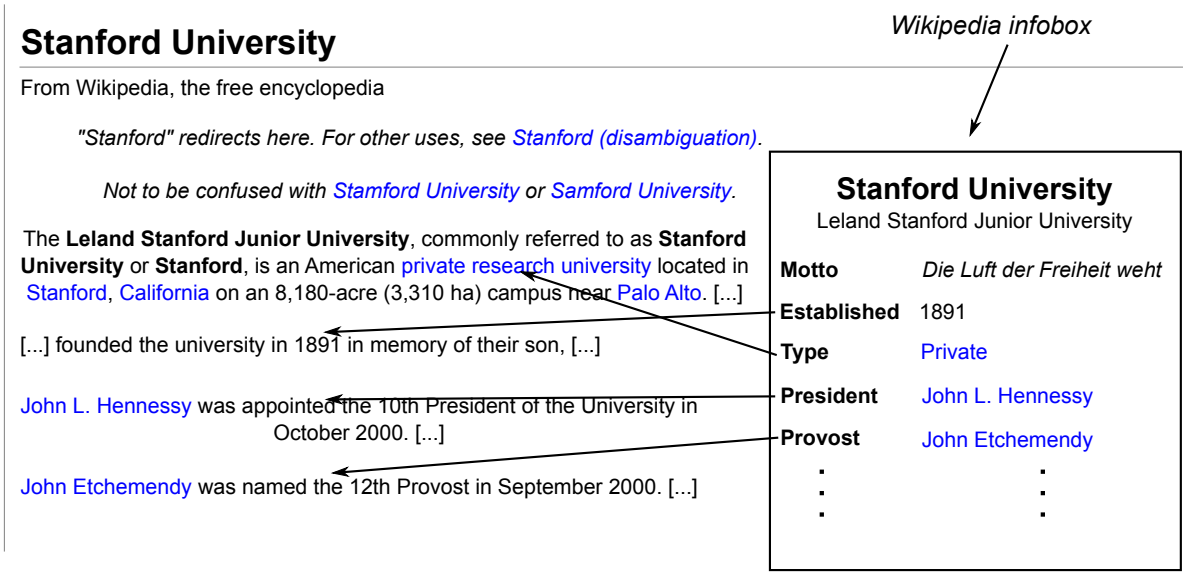


Figure 3.1.: Distant supervision by mapping the structured information of Wikipedia's infoboxes to textual representations in the related article. For example, we can extract a sample that indicates how the semantic relation "isPresidentOf" may be expressed in natural language ("... was appointed the ... President of ...").

database. They map typed relations between two entities (as encoded in Freebase) to unlabeled sentences in a large text corpus. Based on the assumption that "if two entities participate in a relation, any sentence that contain[s] those two entities might express that relation", they construct weakly labeled training copora for a *relationship extraction* task. Hoffmann et al. [167] and Nguyen and Moschitti [280] refine the distant supervision heuristics initially proposed by Mintz et al. [264].

Further works consider distant supervision approaches in the context of biomedicine. For instance, Craven and Kumlien [90] train an information extraction system that searches for relationships between entities such as genes, proteins, diseases, or cell types in biomedical publications. They construct a weakly labeled training corpus by mapping structured information from a protein database to textual representations in *PubMed*[6] articles. Similarly, Morgan et al. [266] consider information extraction from biomedical publications. Using a weakly labeled corpus, they train a hidden Markov model to detect mentions of gene names in PubMed abstracts.

Distant Supervision for Sentiment Analysis

The most cited text corpus for *sentiment classification*, namely the "Cornell Movie Review Dataset" [296, 297], is based on an indirect crowdsourcing approach. As described in the introductory example, IMDb.com movie reviews along with the user-provided ratings constitute the weakly labeled training samples. Pang and Lee [295] construct a similar corpus for sentence level *subjectivity classification*. They extract text snippets from RottenTomatoes.com[7] as samples for the subjective class and plot summaries from IMDb.com as samples for the objective class. Due to the simplicity and convenience of the approach, many more corpora were constructed by exploiting the mapping between numerical ratings and textual representation [44, 232, 265, 403]. Some review sites also provide numerical ratings for individual aspects (see Section 2.2.2), which can be used to create more fine-grained annotations [164, 348, 402].

[6] PubMed is a database that covers abstracts of publications on life sciences and biomedical topics. http://www.ncbi.nlm.nih.gov/pubmed/

[7] http://www.rottentomatoes.com

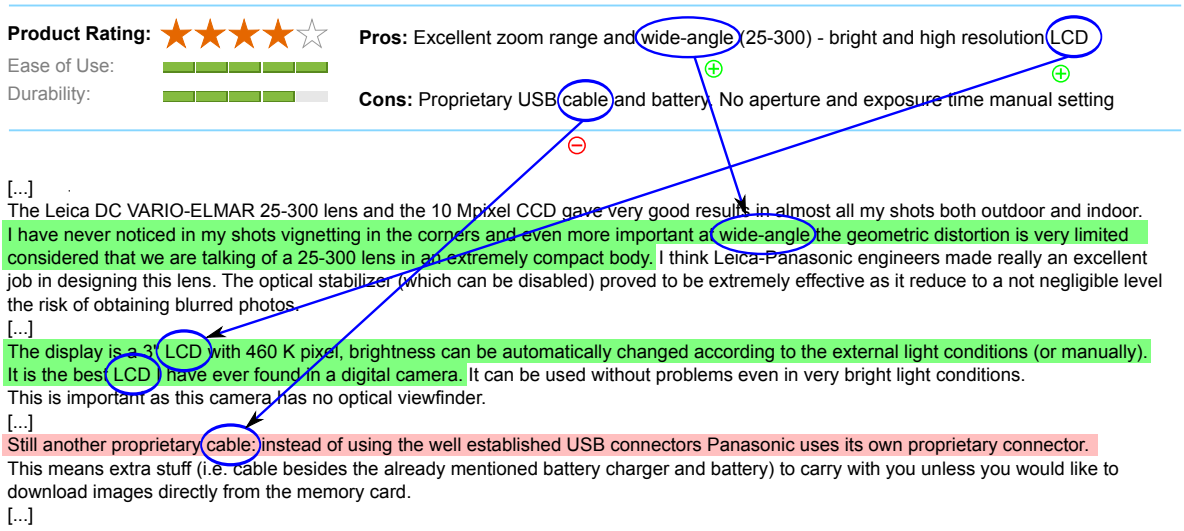


Figure 3.2.: Distant supervision by mapping aspect keywords from pros/cons summaries to sentences in the review text.

Instead of exploiting review ratings to determine the sentiment polarity of a document, Read [315] proposes to consider occurrences of specific *emoticons*[8]. If a paragraph contains a "smile emoticon" (:-)), it is labeled as positive; if it contains a "frown emoticon" (:-(), it is annotated as negative. The same procedure is used by Go et al. [145] and Pak and Paroubek [291], who construct weakly labeled corpora of microblogging posts. Purver and Battersby [305] extend the basic idea and build a weakly labeled dataset for *emotion classification*. In particular, they map emoticons and hashtags[9] to the six basic emotions defined by Ekman [114] (see also Section 2.1.1).

Similar to some approaches we will present in Part III, some researches examine the utility of pros/-cons summaries for sentiment analysis tasks. For instance, [51] make the distant supervision assumption that aspect keywords that are mentioned in the pros/cons are likely to be discussed in the review text (see Fig. 3.2). They train a generative model that allows to predict which key aspects a given review document discusses. The basic idea of exploiting pros/cons information to label sentences in the review text originates from Kim and Hovy [210]. But instead of learning a model that predicts key aspects, they propose a method that automatically extracts reasons for pros and cons from a review document.

Further works that consider distant supervision approaches in the context of sentiment analysis are for instance by Thomas et al. [381], who gather weakly labeled data from congressional floor debates, and Barbosa et al. [28], who train a *review identification* system on heuristically constructed training corpora. Naturally, the concept of indirect crowdsourcing is not constrained to information extraction or sentiment analysis tasks. For example, Mihalcea [261] describe a heuristic that exploits Wikipedia's internal structure to create training samples for a supervised *word sense disambiguation* system. Spitkovsky et al. [358] examine a "lightly-supervised" approach that leverages HTML markup to guide the training of a *natural language parser*.

[8] The term emoticon refers to "a group of keyboard characters (as :-)) that typically represents a facial expression or suggests an attitude or emotion and that is used especially in computerized communications (as e-mail)" (definition in MerriamWebster.com http://www.merriam-webster.com/dictionary/emoticon, retrieved 12/2012).

[9] A hashtag is a word that is prefixed with the symbol # (e.g., #happy or #sad). It is a form of metadata and typically groups semantically similar or topic-related messages in microblogging services (definition adapted from Wikipedia: http://en.wikipedia.org/w/index.php?title=Hashtag&oldid=535475973).

General Remarks

Considering the common ground of the cited case studies, we define distant supervision as follows:

Definition 3.1 (Distant Supervision)**.** *The paradigm of distant supervision describes the approach of heuristically aligning two data sources, where one source is interpreted as the label set and the other as the sample set. The sample set is typically extracted from the Web, the label set may originate from structured sources (e.g, a knowledge base) or may also be extracted from unstructured sources such as the Web. The* ***heuristic labeling function*** *is based on a* ***distant supervision assumption****, for example, that sentences occurring in the pros of a customer review express positive sentiment. The purpose of distant supervision is to automatically gather* ***weakly labeled data*** *that is used to train a machine learning algorithm.*

An immanent presumption for distant supervision is that the processes of extracting data and aligning instances with labels are simple and accurate. We believe that the advent of user generated content (UGC) is an enabling factor in that spirit. To support this believe, consider the following beneficial characteristics:

- **Homogeneity and Structure:** Websites that rely on user generated content typically use structured interfaces (e.g., HTML web forms) to let users create content on their site. This structure is reflected by the HTML-templates that render the content. In effect, user generated content is more likely to exhibit some structure and thus promotes *easy and accurate content extraction*. Furthermore, contributors often agree on style and structure conventions (e.g., the Wikipedia "Manual of Style"[10]), which obviously enhances homogeneity and therefore also promotes the use of web data extraction heuristics.
- **Metadata:** Also, in order to structure the content, UGC-enabled sites encourage their users to provide metadata about their entries. For instance, metadata may be given in form of tags, mappings to an existing topic (e.g., a forum entry), or a rating (e.g., for a product review). Generally, such metadata have a clear semantic and lend itself to be used as *labels* for training instances.
- **Accessibility:** Web 2.0 platforms typically store user generated content in databases. Rendering this content as a web site is only one way to "export" the data. Due to the fact that data is stored in databases, exporting it in a machine-readable format such as XML is very easy. Thus, many platforms provide *structured access* to their content, e.g., by means of web-services or RSS-feeds. In such cases web data extraction is a trivial task.
- **Amount of Data:** Due to their openness and simple interfaces, UGC-enabled websites provide the potential of massive online collaboration. The huge amount of information available through popular sites promotes the use of *high precision/low recall heuristics* for the extraction of labeled data.

3.2. Crowdsourcing Approaches

Annotated training corpora are traditionally created by a group of domain experts or explicitly briefed persons (e.g., students with certain background knowledge). Selecting such experts as "laborers" to do the tedious, time-consuming, and often redundant work of hand-labeling data induces high per-annotation costs. Besides distant supervision techniques, researchers currently study the potential of using *crowdsourcing* approaches.

[10] http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style

Crowdsourcing basically means to outsource a problem or a task to an undefined (typically distributed) workforce of people. As a prominent example, consider the "Millennium Prize"[11]. Mathematicians (but not limited to) around the globe are attracted by an one million dollar award (and the fame!) for correctly solving one of the seven most crucial mathematical problems. Another example is Google's "Android Developer Challenge"[12]. To promote their mobile operating system and to attract developers to their platform, Google offered prizes for the most innovative applications.

However, crowdsourcing is more than a new form of holding contests and awarding prizes. Companies and institutions have adopted it as an alternative business model [171, 371]. Amazon's "Mechanical Turk"[13] (AMT) may be regarded as a step towards a professionalization of this business model. AMT provides the infrastructure for an online labor marketplace — basically, it connects requesters announcing tasks to the system with workers who are monetarily rewarded for accomplishing them. Workers are typically paid by a piece-rate and wages depend on the complexity of the tasks. Common tasks are for example labeling images, verifying hyperlinks on websites, transcribing audio, verifying addresses and phone numbers, classifying products into categories, or providing spelling corrections. Tasks on AMT are mostly of simple and repetitive nature, but require human intelligence and cannot be solved by computers.

Researchers currently study the impact of using such an undefined workforce of non-experts for a variety of annotation projects in natural language processing. For instance, Snow et al. [347] and Hsueh et al. [175] compare the quality and usability of non-expert annotations (obtained via AMT) with expert annotations. Their basic findings are that crowdsourcing can be considered as a cheap, fast, and yet reliable method to collect annotations. The plenitude of data and the acquisition of parallel, independent annotations allows to compensate the increased noise that is introduced by relying on non-expert annotators. Munro et al. [270] report similar results in the context of language studies. Lofi et al. [240] consider crowdsourcing approaches for information extraction tasks.

Crowdsourcing approaches can be characterized by the way crowds are attracted [103]. Apart from explicit pecuniary incentives, such as the wages paid on AMT, other, more implicit reasons for collaboration exist. One important motivation is amusement. Some researchers exploit this fact and design *games with a purpose*. For instance, von Ahn [401] creates a game where players collaboratively annotate images or von Ahn et al. [400] propose a web-based game where users locate objects in images. Similar crowdsourcing approaches exist for the purpose of creating annotated datasets for music/sound retrieval [224] or preference ranking [34].

3.3. Machine Learning Approaches

Whereas distant supervision and crowdsourcing approaches serve to reduce the per-annotation costs, certain machine learning techniques promise to reduce labeling costs by minimizing the required amount of labeled training data. Probably the most prominent technique in this context is *semi-supervised learning* (SSL) [70]. Halfway between supervised and unsupervised learning, it addresses the problem of learning in the presence of both, labeled and unlabeled data. The underlying assumption is that the required number of labeled samples can be reduced by taking advantage of large amounts of unlabeled data (which is typically available at no costs). One of the earliest ideas to SSL is *self-learning* (also denoted as *bootstrapping*) [196]. It is successfully applied in a variety of natural language processing (NLP) tasks [3, 26, 85, 324, 406]. Another prominent scheme is *active learning* (AL) [285, 343]. Active learning systems *actively* select the samples that a user should annotate. By choosing the most "informative" samples, the hope is to increase the learning rate of a system and thus to minimize the number of required samples.

[11] http://www.claymath.org/millennium/
[12] http://code.google.com/android/adc/
[13] https://www.mturk.com/

Closely related to the SSL and AL is *pre-labeling* [25, 93], which however aims at reducing the per-annotation costs. The idea is to assist the annotator by using the learned model to pre-label previously unlabeled instances. The underlying assumption is that (1) only a fraction of pre-labeled instances needs revision and (2) that it is easier to revise incorrect annotations than to produce new annotations from scratch. Thus, the goal is to reduce the amount of actions required to label a single instance.

3.4. Summary

In this chapter, our goal was to provide an overview the distant supervision paradigm. We embedded the topic as part of a more general question, namely how to reduce the manual effort that is typically entailed with creating lexical resources such as labeled training copora. We introduced distant supervision as a technique to automatically create weakly labeled training data for machine learning algorithms. We presented a definition of the paradigm and described several successful applications in information extraction, sentiment analysis, and natural language processing. We further pointed out some beneficial properties of user-generated content and argued that the advent of this kind of data represents an enabling factor for distant supervision approaches. In the last two sections, we completed the picture by briefly discussing two other approaches for reducing labeling costs, namely crowdsourcing and semi-supervised machine learning.

Part II.

Models, Datasets, and Problem Analysis

> Whenever a theory appears to you as the only possible one, take this as a sign that you have neither understood the theory nor the problem which it was intended to solve.
>
> *Karl Popper*

4. Modeling the Expression of Sentiment in Customer Reviews

This chapter introduces our perspective on how to model the expression of sentiment in customer reviews. Providing an analytic definition (in a mathematical sense) is rather impractical. Nevertheless, a thorough understanding and an accurate model of how sentiment is expressed form the indispensable basis for developing annotation schemes, compiling evaluation corpora, and computationally treating opinions. Needless to say, the presented model is influenced by previous work on modeling sentiment in natural language text, such as Bloom et al. [45], Ding et al. [102], Kessler et al. [209], Polanyi and Zaenen [303], Popescu and Etzioni [304], Toprak et al. [386] and Wiebe et al. [417]. We indicate similarities and differences to these other models where appropriate and go into more detail in a separate section on related work.

Although dealing with natural language and its use, we clearly state that our perspective on modeling the expression of opinion is the perspective of an engineer rather than that of a linguist, psychologist, or sociologist. It is not our ambition to develop a universal model that would cover also the smallest linguistic nuance, nor is it our ambition to present or resemble a general cognitive theory of emotion and opinion. Nonetheless, we do not neglect the models, theories, and concepts developed in these fields of study. In summary, our fundamental requirements for the model are the following:

- **Operationalization**: An opinion is an abstract concept. Our model shall define its (abstract) constituents in such a way that they can be measured. In that sense, operationalization corresponds to reducing any existing ambiguities. In particular, we require that our model can be implemented by a well-defined annotation scheme. Given annotation instructions, a human must be capable of annotating a customer review in accordance to the model. Annotation instructions must be directly derivable from model assumptions and must be clearly expressible.

- **Constructiveness**: We refer to our model as being constructive if it can be implemented in software, that is, a computational understanding of the model can be achieved. We restrict our model in such a way that, with employing state of the art or conceivable technology[1], a computer program can be developed that is capable of automatically annotating a customer review in accordance to the model.

- **Aspect Orientation**: Most customer reviews contain mixed opinions. Reviewers comment on different aspects of a product — some aspects are praised, others are criticized. A "five out of five star review" may also contain negative judgments and a "one out of five star review" may nevertheless express positive sentiment on some aspects. A practicable model must be capable of capturing opinion expressions at the level of aspects[2].

- **Focusing**: One size does not fit all. Different application domains exhibit different characteristics and therefore influence the model which is to be developed. For example, opinions expressed in newswire text are frequently conveyed by reported or direct speech events (e.g., "During the meeting, Berlusconi criticized the left sowing 'hatred and envy' and 'politicized judges'."). In contrast, reported and direct speech is virtually never used in customer reviews

[1] Here, we address base technology that is employed in the context of NLP and machine learning, such as part of speech taggers, natural language parsers, general sequence taggers, or classification and clustering algorithms.

[2] The ability to recognize opinions at the level of aspects subsumes the capability of classifying the general tone of a product review — for example, by aggregating the aspect level sentiment.

and, as a consequence, does not need to be considered as part of a model. We explicitly restrict the model's scope to the domain of customer reviews. We do not claim that the model is applicable with the same level of fit in other domains, such as political debates or financial news.

We model the expression of opinion in customer reviews at two different levels of granularity. In particular, we consider a fine-grained *expression level model* and a more coarse-grained *discourse oriented model*:

- **Expression Level Model**: The expression level model allows to capture individual utterances of opinion at the *mention level*. For example, in the sentence "The images were soft and grainy, the focus was inconsistent and the color were often oversaturated.", we can observe three aspect mentions ("image", "focus", and "color") and each of them is associated with a negative sentiment. Modeling at the expression level allows to describe the individual linguistic constituents and relations that make up the sentiment targets and expressions.
- **Discourse Oriented Model**: Aspects and sentiments may be modeled on a more coarse-grained level. Reconsider the preceding example sentence. We may also classify the whole sentence as expressing negative sentiment on the picture quality of the camera. Instead of considering the individual linguistic constituents of an opinion expression, we are more interested in the abstract function and properties of a specific text passage as part of the discourse structure of a review document.

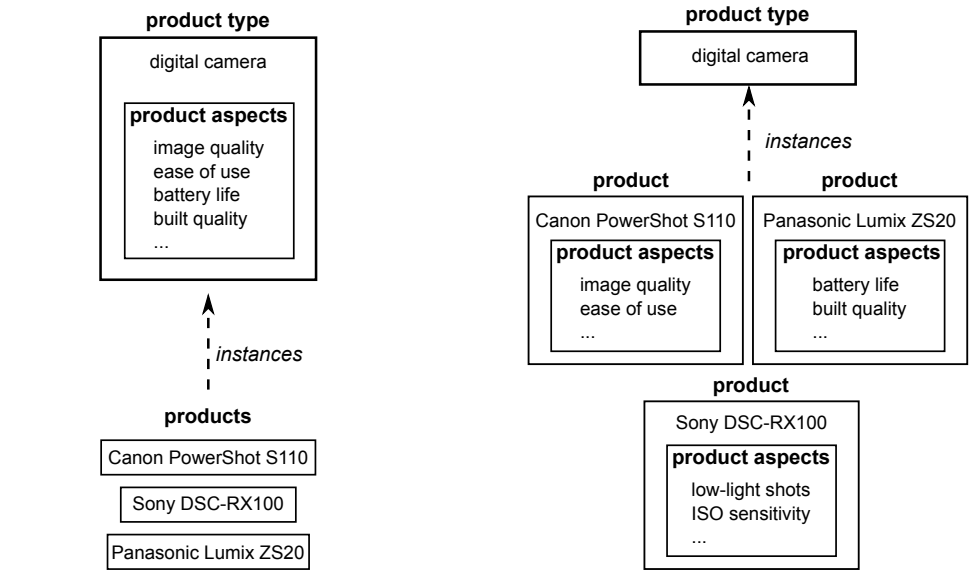
The rest of the chapter is organized as follows: Preliminary to discussing the actual models, Section 4.1 defines our understanding of product aspects by introducing the concept of a *product type taxonomy*. Next, we describe our perspective on modeling opinion expressions in detail. We describe the *discourse oriented model* in Section 4.2 and introduce the *expression level model* in Section 4.3. We discuss some limitations of both models in Sections 4.2.4 and 4.3.4. Section 4.4 reviews related work on modeling opinions and identifies relevant differences. Section 4.5 summarizes the chapter.

4.1. Modeling Product Types, Products, and Aspects

As the section title implies, we distinguish between the terms *product type*, *product*, and *product aspect*. We refer to a product type when addressing a whole class of products. For example, the classes of digital cameras, mp3 players, hotels, or restaurants all constitute a separate product type. We use the term product to denote individual instances of a product type, e.g., "Canon EOS 600D", "SanDisk Sansa Clip+", "Hotel Adlon Kempinski", or "Curry 36". As illustrated in Fig. 4.1, product aspects may be defined with respect to a concrete product (e.g., "Canon EOS 600D") or in terms of a whole product type (e.g., digital cameras). How to model the aspects depends on the actual application scenario and the requirements for a review mining system. Due to the shape of our review corpora, we decided to model product aspects with regard to the product type. (Other researchers, e.g., Hu and Liu [177], define aspects with regard to a concrete product.) Independent of this choice, aspects may either refer to physical entities (e.g., the lens of a digital camera), or to abstract objects, such as "picture quality" or "ease of use".

With regard to product aspects, we further distinguish between *coarse-grained* and *fine-grained* aspects. We speak of coarse-grained aspects when we mean the broader concepts that are predominantly commented on in reviews of a specific product type. In the following, we interchangeably denote this level of granularity as *concept level* or *topic level*. We cannot provide an accurate definition for what constitutes a concept and restrict ourselves to examples[3]. In fact, our notion of coarse-grained

[3] Also see Buitelaar and Magnini [60, sec. 4.3] who discuss the problem of defining concepts w.r.t. ontology engineering.



(a) aspects as attributes of the product type

(b) aspects as attributes of a product

Figure 4.1.: The relation between the concepts *product type*, *product*, and *product aspect*. Product aspects may be modeled as attributes of the product type (a) or of a concrete product (b).

aspects is data driven. Appendix D describes how we use *probabilistic topic models* to derive a set of relevant topics from a large collection of review documents.

We speak of fine-grained aspects when we refer to concrete instances of an entity at the concept level. For example, the aspects "battery life", "battery power", or "battery charger" are all instances of the broader concept "battery and charging". We postulate that a fine-grained aspect is associated with exactly one entity at the concept level. We denote this fine-grained level of analysis as *mention level*; conceptual entities become manifest as concrete mentions of fine-grained aspects within a text. A mention is a normalized, canonical representative of a *textual surface form* (the actual string) of an aspect. For example, the fine-grained aspect "battery life" may also occur as "life of the battery" or "battry life" (misspelling) at the textual surface.

Similar to Ding et al. [102], Popescu and Etzioni [304], or Kessler and Nicolov [207], we postulate that the attributes of a product type can be hierarchically decomposed along *semantic relations* such as "part-of" (meronymy), "type-of" (hyponymy), "feature-of", or "synonym-of"[4]. We introduce this kind of hierarchy for both the concept level and the mention level of aspects. We propose to first decompose a product type into a set of broader concepts. These concepts are then hierarchically structured along semantic relations, resulting in a tree representation where the root node is the product type and inner nodes are concepts. For example the concept *user interface* is a feature of the concept *ease of use* which itself is a feature of the product type *digital camera* (root node). On the mention level, we associate each fine-grained aspect to exactly one entity on the concept level (inner node of the tree), so that each concept is related to a set of fine-grained aspects. Sets of fine-grained aspects may again be hierarchically decomposed along semantic relations. We summarize our description more formally in Definition 4.1:

[4] See for instance Cruse [91] or Murphy [271] who study semantic relations in detail.

Definition 4.1 (Product Type Taxonomy). *Let* $\mathbf{P}$ *be a specific type of product or service (e.g., digital camera or hotel). Then a product type taxonomy* $\mathbf{T}$ *is a tree with* $\mathbf{P}$ *as the root and subordinate concepts* $\mathbf{C_i}$ *as inner nodes. A concept* $\mathbf{C_i}$ *is related to its parent (the root or an inner node) through a semantic relation. Each node* $\mathbf{C_i}$*, including the root, is associated with a set* $\mathbf{F_{C_i}}$ *of normalized textual surface forms of products aspects, denoted as fine-grained aspects. A set* $\mathbf{F_{C_i}}$ *of fine-grained aspects may again be hierarchically structured along semantic relations.*

Figure 4.2 illustrates the constituents of a product type taxonomy (hierarchies at the mention level are not explicitly depicted).

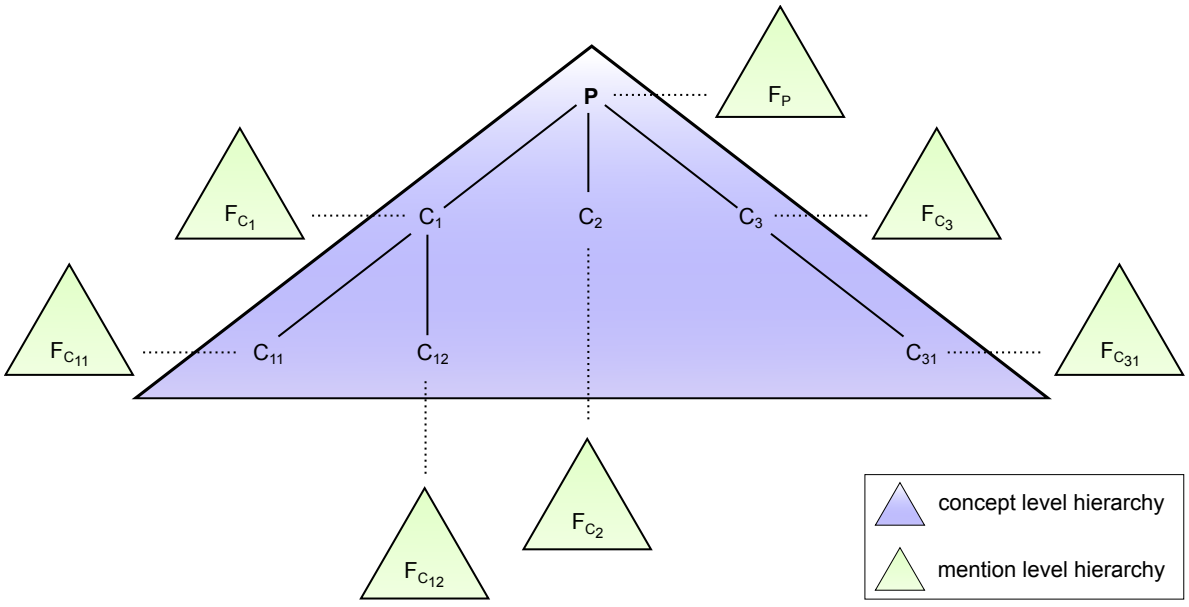


Figure 4.2.: Building blocks of a product type taxonomy. The illustration highlights the hierarchical decomposition at the concept and mention level.

The motivation for modeling product aspects in a hierarchy (opposed to a flat, unrelated list) follows two observations: First, decomposing objects (here products) along semantic relations is quite natural. For instance, ontologies [152] represent domain knowledge in such a way. Another example is the WordNet lexical database [263], which organizes *synsets* of nouns along meronymy and hyponymy relations. Second, we can observe that sentiments propagate along such relations. That is, sentiments expressed towards a part, type, synonym, or feature inherently address also the related node. We can aggregate sentiments on each level of a hierarchy (concept and mention level), allowing for multi-level fine-grained analysis. For example, when expressing negative sentiments on the coverage of the flash (e.g., uneven coverage), a negative sentiment is inherently conveyed towards the fine-grained aspect flash (and the equally named concept) and thus also to the product itself. The hierarchical model therefore helps to structure and summarize the detected sentiment in a comprehensive form. A customer review mining tool may visualize the hierarchy and help the user in exploring the extracted information.

Figure 4.3 exemplifies a product type taxonomy for digital cameras and illustrates how sentiments propagate in this model. Concepts are depicted as blue rectangles, which are connected via *part-of* and *feature-of links*. Each concept is associated with a set of fine-grained aspects (the green rectangular boxes). Again, for reasons of clarity and comprehensibility we do not illustrate semantic relations between fine-grained aspects (e.g., that coverage and reach are features of a flash). Sentiment expres-

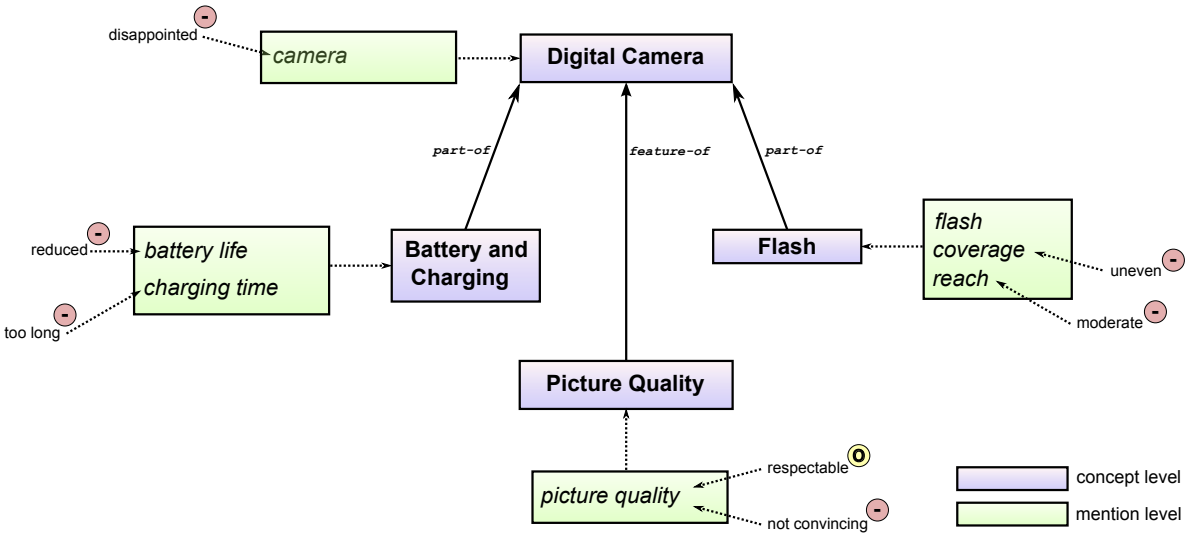


Figure 4.3.: Propagation of sentiment within an exemplary product type taxonomy.

sions are linked to concepts via concrete mentions. Polarities attached to a sentiment expression are depicted as circles, enclosing either a plus sign (positive), a minus sign (negative), or a zero (neutral).

We want to point out that it is not our intention to describe a complete *domain ontology* of a product type. We rather regard our definition of a product type taxonomy as a simple tool that allows us to express the important hierarchical relations between product aspects. We believe that it is flexible enough to describe the hierarchical relations of product aspects for very diverse product types (e.g., digital cameras, mobile phones, hotels, restaurants, movies), and that it is of great value for structuring and summarizing the information obtainable from customer reviews.

4.2. Discourse Oriented Model of Customer Reviews

Like the majority of text documents with informative character, a customer review is written in a linear, coherent way. Consider an author reviewing his newly bought camera. He may start his review with a brief introduction, referring to his personal context (e.g., "I'm an enthusiastic photographer since more than a decade now..."), then formulate his thoughts on different aspects of the product, first about the picture quality then about battery life, etc., he may give some advice on the usage of the product and finally he may conclude by pointing out his overall impression. That is, we can observe a flow of different *topics* (picture quality, battery life) and *functions* (personal context, evaluation, advice, conclusion). Such shifts of topic or function represent a document's *discourse structure*. We therefore briefly review the concept of *discourse* as understood in NLP [5]:

4.2.1. Discourse in Natural Language Processing

Very broadly speaking, discourse addresses language phenomena beyond the sentence level. We cite Jurafsky and Martin [198, chap. 21] who point out that "language does not normally consist of isolated, unrelated sentences, but instead of collocated, structured coherent groups of sentences". The literature typically distinguishes between a fine-grained modeling of discourse structure and a coarse-grained perspective. In fine-grained modeling one is interested in typifying the relations between sentences of a document. For example, a sentence S_1 may be related to a succeeding sentence S_2 as

[5] Take note that in other fields of study, such as philosophy or sociology, the notion of discourse is understood more broadly.

being the explanation for an incident described in S_2. Other typical *coherence relations* are for example "Cause", "Result", "Elaboration", or "Occasion" [198]. Fine-grained discourse structure is generally represented in a hierarchy. For instance, Asher et al. [17, 18] or Somasundaran [351] study the fine-grained discourse structure in the context of sentiment analysis.

We are interested in the coarse-grained discourse structure of a document — that is, we only consider the coherency of text passages, but without addressing explicit coherence relations. Such a shallow model is commonly denoted as *discourse segmentation*. Modeling the flow of topics and functions can be regarded as an instance of discourse segmentation. In this specific case, coherence is defined with respect to the two orthogonal dimensions *topic* and *function*.

4.2.2. Motivation

Our motivation to consider a discourse oriented model comprises mainly three aspects: First, we believe that besides considering very fine-grained product aspects (namely individual mentions), there exists also an *information need* to subsume concrete aspect mentions to coarser-grained aspects. The discourse oriented model follows this path.

A second motivating factor is based on the observation that aspects and opinions are not necessarily expressed explicitly. In such a case we cannot identify a specific phrase on the mention level that conveys the information. The addressed aspect or sentiment may be recognizable only when considering context, i.e., a longer text passage, such as a whole sentence or paragraph. In the following example the aspect is implicitly mentioned: "I really like that the camera fits into my pocket.". The author addresses the size of the camera, but never explicitly mentions this aspect. The general tone of this passage is positive, as the author states that he likes this characteristic of the camera. Also opinions may be communicated implicitly, i.e., they cannot directly be attributed to a concrete sentiment bearing text span. This is for instance the case if specific facts imply an evaluation or judgment. In the literature the phenomenon is referred to as *objective polar utterance* [433], *evaluative fact* [282], or is denoted as *polar fact* [386]. As an example for such a polar fact, consider the following sentence: "The flash recycle time can reach up to 20 seconds.". Although the sentence contains only facts, it implies an evaluation of the flash recycle time. A flash recycle time of 20 seconds is clearly not desired and thus the sentence implies negative sentiment.

A third motivating factor is that we are interested in distinguishing the different functions which text passages typically fulfill in the discourse structure of a review document. Although a customer review is qua definition a highly focused and evaluative document, not every part of it addresses a relevant topic or is of evaluative nature. For example, the introductory sentence we cited at the beginning of this section is irrelevant with respect to evaluating the product, but it can be relevant for estimating the trustworthiness or expertise of the author (e.g., a professional photographer could be considered more proficient than a person who is completely new to photography). Differentiating between *discourse functions* allows us to model and analyze customer reviews with respect to a further dimension. For example, it is capable of distinguishing between explicitly evaluating, concluding, problem describing, or advice giving text passages. We believe that there exists an information need with regard to distinguishing these discourse functions. It is therefore our goal to derive the most typical functions used in the discourse structure of customer reviews.

4.2.3. Model Description

Having motivated the discourse oriented model, we now describe its basic constituents and their relations in more detail. Figure 4.4 provides a graphical overview of the model. The basic constituents are represented by the three orthogonal (i.e., independent) dimensions, *discourse function*, *topic*, and *sentiment polarity*. Each segment is at least attributed with a discourse function, but values on the topic and polarity dimensions may be empty. An empty value on the topic dimension indicates an

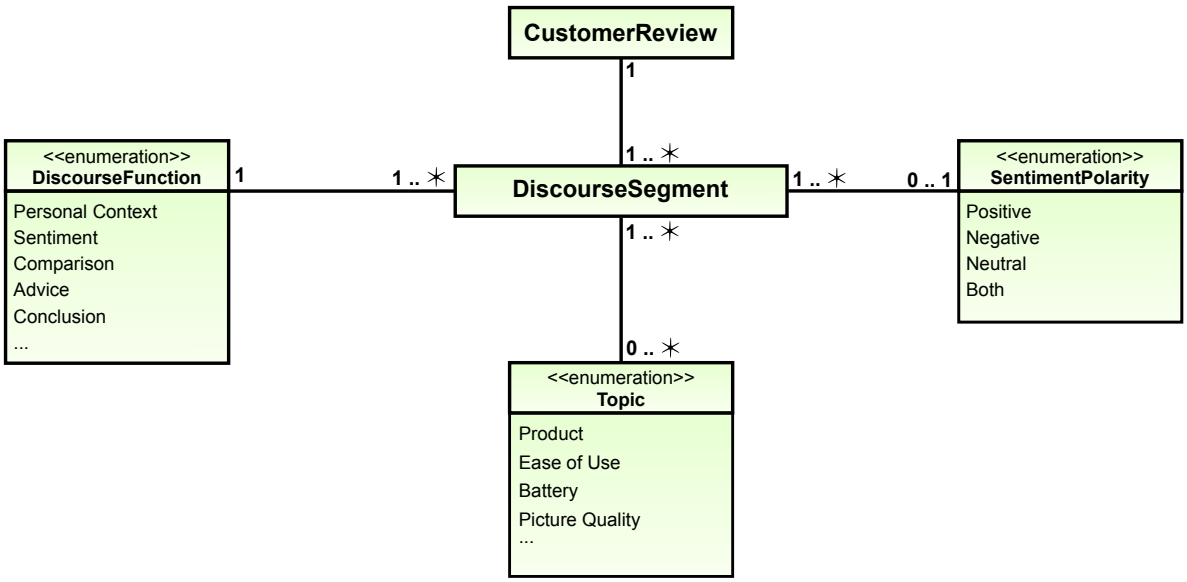


Figure 4.4.: The discourse oriented model of sentiment in UML notation.

off-topic segment — that is, none of the predefined topics can be attributed with the text passage. An empty value on the polarity dimension indicates that the segment contains only objective statements. Segment boundaries are defined by the coherency with regard to all three dimensions. That is, each shift in either the function, topic, or polarity dimension defines a new discourse segment.

Definition 4.2 (Discourse Oriented Model). *A customer review is subdivided into a set of consecutive, non-overlapping discourse segments. A discourse segment consists of a text passage that exhibits coherency with respect to its* ***discourse function***, ***topic***, *and* ***sentiment polarity***. *Each segment is associated with exactly one discourse function. Furthermore, segments are either* ***off or on-topic***. *On-topic segments are associated with one or more predefined, domain dependent topics. Discourse segments may contain explicit or implicit utterances of opinion. In that case, a segment is attributed with a* ***sentiment polarity***.

Discourse Function

A discourse function refers to the function of a segment with respect to the discourse structure of a review. We postulate that a predefined set of discourse functions is universally applicable for all customer reviews, independent of the addressed type of product. A complete list of these predefined discourse functions is specified as part of the annotation scheme we present in Section 5.2. The annotation guidelines in Appendix A.2.7 discuss the individual discourse functions in detail and provide many examples. Section 6.1.3 analyzes the distribution of discourse functions in our hand-annotated corpora.

Topic

A topic refers to the definition of coarse-grained product aspects as presented in Section 4.1 — that is, a topic subsumes fine-grained aspects that are semantically or conceptually similar. For example, in the digital camera domain, we can imagine topics such as "picture quality", "ease of use", "video recording", "lens", or "battery life".

Opposed to the set of discourse functions, the predefined set of topics is dependent on the specific product type under consideration. We postulate that such a set of topics is inherent to each particular domain and therefore, a list of topics can be compiled[6]. We present such a list of predefined topics for the two product types "digital camera" and "hotel" in Section 5.2.

As the minimal unit for a discourse segment is a sentence, it is possible that multiple, different topics are addressed in a single segment. We therefore model the association between topics and discourse segments as a many-to-many relation. One discourse segment may be associated with multiple topics (or none at all). This assumption contrasts other similar models [55, 136, 194] which postulate that a sentence is associated with no or exactly one topic.

Sentiment Polarity

Sentiment polarity is modeled independently from the discourse function and topic dimensions. Also off-topic segments may contain subjective information. For example, a reviewer may highlight what a professional photographer he is and praise his skills. Each segment which expresses sentiment is attributed with a sentiment polarity type of either "positive", "negative", "neutral" or "both". For the same reason as we allowed for multiple topics, we introduce the special sentiment polarity type "both". A segment is attributed with this type in case it contains positive and negative sentiment expressions. Objective segments are simply not associated with any sentiment polarity type. Furthermore, we distinguish between polar fact segments and other polar segments. The sentiment polarity of a polar fact segment is restricted to either "positive" or "negative". Take note, that we do not cover sentiment strength/intensity in the discourse oriented model.

Summary

With the discourse oriented model we present a coarse-grained, topic-oriented perspective on modeling customer reviews. We assume that a review can be partitioned into non-overlapping coherent segments, where segment boundaries are aligned along sentence boundaries and are defined with respect to a shift in one of the three dimensions. We further postulate that a predefined set of discourse functions exists and is applicable to every customer review, independent of a specific product type. Topics refer to coarse-grained aspects of a product. A predefined list of topics can be compiled for each type of product. Sentiment (possibly in the form of polar facts) is recognized independently of the other dimensions and without considering the sentiment strength. The minimal unit of text that may constitute a discourse segment is a single sentence.

4.2.4. Limitations of the Discourse Oriented Model

Predefined Set of Topics

The model assumes that a fixed set of topics can be compiled for each domain of interest. But there is no assumption on the cardinality of such a set. The set of topics could possibly consist of a single element (effectively just indicating on or off-topic segments) or hundreds (effectively representing each separate product aspect as a topic). That is, the cardinality of the set depends on the desired granularity with regard to the topics. It is the task of the modeler to decide on the granularity (typically based on application needs). However, it is difficult to decide whether a compiled set of topics is complete. A simple solution would be to add an artificial topic "OTHER" that subsumes all relevant topics not included in predefined set of topics. We overcome the problem of determining the set of topics by taking a data-driven approach. We basically use *probabilistic topic modeling* techniques that

[6] Take note that a subset of coarse-grained product aspects is often incorporated as a collection of explicitly ratable aspects on review sites. For instance, the website Epinions.com asks the reviewer to explicitly rate the aspects "ease of use", "durability", "battery life", "photo quality", and "shutter lag" when evaluating a digital camera.

help to explore and structure text corpora in an unsupervised manner. More details are provided in Chapter 8.

Coarse-Grained Perspective

Due to its coarse-grained perspective, the model is not capable of capturing the basic constituents that convey sentiment. Instead of considering the individual linguistic features that make up an opinion expression, the model only captures the effects. For example, in this model we are not aware that a negation "not" shifts the prior polarity of the expression "like" — we are only aware that a text passage is attributed with negative sentiment. However, the level of granularity suffices to answer questions such as "Which reviews contain passages that express positive/negative opinion on the location of the hotel?". It also allows for tasks such as "Show all snippets that conclude a review and highlight the service of the hotel".

Enumeration of Product Aspects

Discourse segments are aligned at sentence boundaries, thus the smallest unit of a segment is a single sentence. Although we allow multiple topics to be associated with a single segment, the model does not fit well when reviewers simply enumerate their likes and dislikes (e.g., "Friendly and attentive reception staff, impeccably clean rooms every day, comfortable bed (if a bit short for my 6'3"), ipod dock with radio, modern design without being overly pretentious, big flat-screen tv, and free Wifi!!!"). The model is intended to capture the general topic prevalent in a text passage instead of capturing individual mentions at the sub-sentence level. In the previously cited example sentence nearly each noun phrase is associated with a separate topic. We may attribute the topics "service", "cleanliness", "sleep quality", "room amenities", "decoration", and "internet".

4.3. Expression Level Model of Customer Reviews

We now describe a more fine-grained perspective on modeling customer reviews which we denote as *expression level model*. We are interested in identifying fine-grained (linguistic) constituents of sentiment targets and expressions and want to determine how they relate to each other. Opposed to the discourse oriented model, which addresses the general topic and tone of a text passage, individual mentions of targets and expressions along with their constituents are captured. Fig. 4.5 provides an overview of the expression level model.

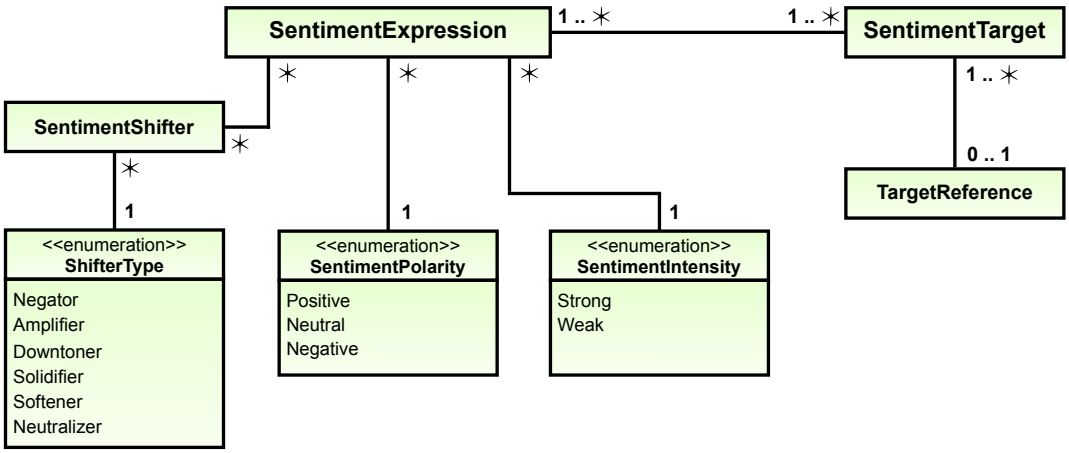


Figure 4.5.: The expression level model of sentiment in UML notation.

To point out the different linguistic constituents and their relations in example sentences, we introduce a graphical notation (see Fig. 4.6). We outline sentiment expressions as a thick, solid rectangle with rounded edges. The example sentence contains the two sentiment expressions "soft" and "cozy". The polarity of a sentiment expression is depicted by thumbs up and thumbs down symbols: 👍 (positive), 👎 (negative) and ☞ (neutral). We indicate the intensity by the following two arrows: → (average) and ↗ (strong). If the polarity of a sentiment expression is *target-specific* (to be discussed), the outlining rectangle is grayed out, such as for the adjective "soft" in the example. Sentiment targets are highlighted with a thick, solid, chamfered rectangle. In this case we have the single sentiment target "king-size bed". All sentiment shifter annotations are illustrated with a thick, dashed rectangle with rounded edges. Here, we have the word "really" as a shifter that modifies both sentiment expressions. If a relation between any of the constituents is of importance, we put emphasis on it by adding named pointers to the graphic. The pointers in the example describe that both sentiment expressions target the same product aspect and that their intensity is shifted by the word "really". Take note that we choose the name of a pointer to reflect its precise semantic, e.g., we write "amplifiedBy" instead of "shiftedBy". For reasons of clarity, subsequent examples will only illustrate the constituents that are relevant in the given context.

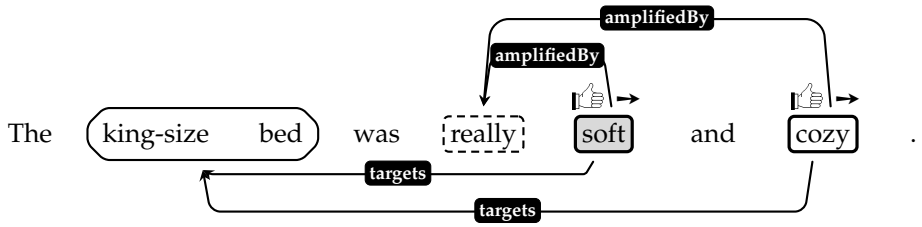


Figure 4.6.: Graphical notation used to explain the constituents of the expression level model.

The following discussions are based on Example 4.1. It shows an artificial review that is intentionally kept simplistic, but suffices to exemplify the discussed ideas.

Example 4.1 (Invented Digital Camera Review)

(4.1) I am utterly disappointed with this camera, I really do not like it.

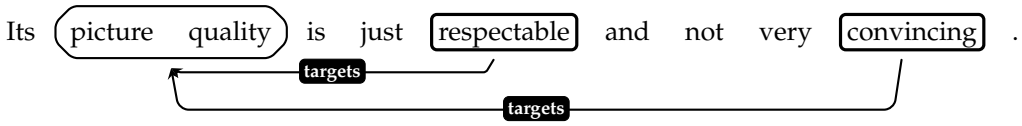
(4.2) Its picture quality is just respectable and not very convincing.

(4.3) And unfortunately the flash has only moderate reach and an uneven coverage.

(4.4) Also the battery life appears to have been reduced and charging time is somewhat long.

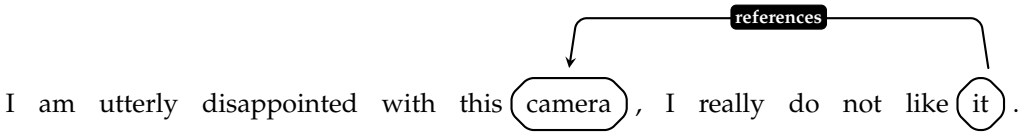
4.3.1. Sentiment Targets

A sentiment target refers to the textual surface form of a fine-grained product aspect. It is associated with one or more sentiment expression and we postulate that there is an existential dependency between sentiment target and sentiment expression. In other words, a sentiment target does not exist (or is not recognized) if no sentiment is expressed on it. We further model the association between sentiment targets and sentiment expressions as a many-to-many relation. A single target may be addressed by multiple sentiment expressions and a single sentiment expression may relate to multiple sentiment targets.



Sentence (4.2) exemplifies that a sentiment target may be associated with multiple sentiment expressions. The sentiment target "picture quality" is targeted by the two sentiment expressions "respectable" and "convincing". In a sentence such as "Especially color accuracy and picture sharpness are amazing.", the single sentiment expression "amazing" addresses the two different targets "color accuracy" and "picture sharpness".

Concerning sentiment targets, we distinguish *nominal*, *named*, and *pronominal* mentions[7]. With regard to these types, we basically follow the nomenclature[8] used in the context of the "Entity Detection and Tracking" task of the "Automatic Context Extraction" (ACE) evaluation campaign [87]. We speak of a *nominal mention* (typically a noun or noun phrase) if the sentiment target refers to an explicitly named product aspect, such as "picture quality" in sentence (4.2) or "battery life" in sentence (4.4). Also named entities, e.g., "Canon EOS 600D", may represent the target of a sentiment expression (e.g., "I fell in love with my new EOS 600D."). We refer to this type of target as a *named mention*. In addition we consider pronominal targets. Sentence (4.1) exemplifies this type:



The pronoun "it" refers to the antecedent "camera". The expression level model covers pronominal relations by means of the "TargetReference" type. In this example the pronoun "it" takes the role of the sentiment target (it is modified by the expression "like") and the aspect "camera" is considered as the target reference (take note that here the camera is also a sentiment target by itself). We only consider pronoun relations if the pronoun is target of a sentiment expression. For example, in sentence (4.2) we disregard the possessive pronoun "its". We further restrict that a pronoun cannot take the role of a target reference — that is, we do not model chained references. Observe that a target reference may refer to the antecedent of multiple pronominal mentions.

4.3.2. Sentiment Expressions

We will now discuss the different constituents of a sentiment expression and how they relate to each other. Naturally, the central part is the concrete utterance that conveys the sentiment, e.g., "disappointed" in sentence 4.1. Its primary attribute is the sentiment polarity:

Sentiment Polarity

Similar to Polanyi and Zaenen [303], Wilson et al. [438], or Toprak et al. [386] we distinguish between *prior polarity* and *contextual polarity* of a sentiment expression — that is, we distinguish between the polarity of an expression when considered in isolation versus polarity when considered in the context of potential sentiment shifters, such as negators or neutralizers. However, our definition of prior polarity is different in the sense that we incorporate its potential dependency on a specific sentiment target. To distinguish our notion of prior polarity from definitions in the literature, we denote it as *target-specific prior polarity* (see also Fahrni and Klenner [125] or Klenner et al. [214]).

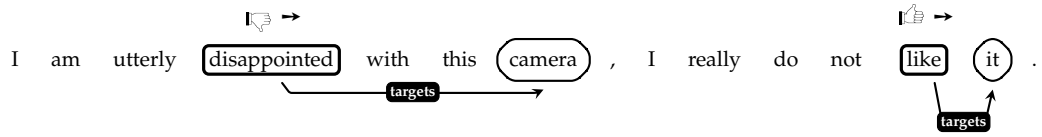
[7] Take note that, primarily for the purpose of corpus exploration, our annotation scheme for the expression level model covers a fourth target type, namely "implicit mention".

[8] refer to the annotation guidelines at http://www.ldc.upenn.edu/Catalog/docs/LDC2005T09/guidelines/EnglishEDTV4-2-6.PDF

Definition 4.3 (Target-Specific Prior Polarity)**.** *Let* $\mathbf{e}$ *be a sentiment bearing expression and* $\mathbf{t}$ *be a sentiment target. Then the target-specific prior polarity* $\mathbf{p}$ *is given by a function* $\mathbf{p}(\mathbf{e}, \mathbf{t}) \rightarrow \mathbf{p} \in \{\mathbf{positive}, \mathbf{negative}, \mathbf{neutral}\}$.

To see that the polarity of a sentiment expression is potentially dependent on its target, consider for example sentence (4.4). When modifying the target "charging time", the adjective "long" has a negative connotation. But when referring for instance to "battery life", it obviously communicates positive appraisal. Observe that the prior polarity of a phrase may, but does not need to be dependent on the target. For instance phrases such as "amazing", "love", or "excellent" are connoted with a positive polarity independent of the concrete target.

Sentiment polarity, as defined in the expression level model, always refers to (target-specific) prior polarity. It ignores any associated sentiment shifter. For example, consider the second clause of sentence (4.1). The sentiment expression is "like" and it targets the pronoun "it" (referring to camera). We assign the sentiment polarity "positive", ignoring the negation "not". The contextual polarity of the sentiment expression would be "negative". However, we do not model contextual polarity explicitly, as it can be derived from the target-specific prior polarity and the associated sentiment shifters.



Sentiment Intensity

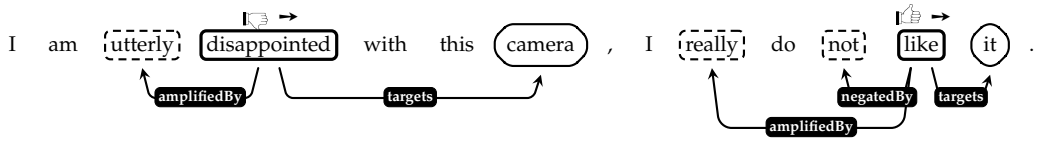
In addition to the prior polarity, a sentiment expression is attributed with a *prior intensity*. We distinguish between two degrees of intensity. A sentiment expression either exhibits an *average* or a *strong* intensity. In contrast to prior polarity, we regard the intensity as immanent to the sentiment bearing phrase, that is we model it as being independent of a sentiment target. As an example, consider again sentence 4.1. Both sentiment expressions "disappointed" and "like" show an average intensity. The reviewer could have used a more intense language. The statement may for instance read "I'm utterly disappointed with the camera, I really hate it.". We would then attribute the sentiment expression "hate" with a strong intensity. Our annotation guidelines in Appendix A.3 provide more examples that show how we distinguish between average and strong intensity.

4.3.3. Sentiment Shifters

A sentiment expression may be modified by zero or more sentiment shifters. Sentiment shifters may influence the *polarity*, the *intensity*, or the *degree of certainty* of a sentiment expression. Founded on the concept of *valence shifters*, as proposed by Polanyi and Zaenen [303], and similar to Kessler et al. [209], we distinguish six major types of sentiment shifters: *negators*, *amplifiers*, *downtoners*, *solidifiers*, *softeners*, and *neutralizers*.

Negation

This is the most important sentiment shifter. It predominantly has influence on the sentiment polarity. Most commonly, it is assumed that it flips the (target-specific) prior polarity of a sentiment expression [209, 303] — for example, as in sentence (4.1).



However, negation may also influence the sentiment intensity and not necessarily flips polarity. For instance, consider the following invented sentence from a hotel review:

(4.5) The breakfast was not fantastic, but it was adequate for the price.

The prior polarity of the sentiment expression "fantastic" is obviously "positive" and its intensity is "strong". But negating "fantastic" does not flip polarity to "negative", it rather shifts polarity towards "neutral". Similar to Liu and Seneff [238] or Remus and Hänig [319], we argue that negation has an "asymmetric" influence on polarity/intensity. Let $>_{polarity}$ be an order defined on a sentiment polarity/intensity scale. For example, consider the order $StronglyPositive >_{polarity} Positive >_{polarity} Neutral >_{polarity} Negative >_{polarity} StronglyNegative$. With regard to negation, we believe that the following relations are true:

$$\left[\begin{array}{lll} \text{Positive (}\textit{good}\text{)} & >_{polarity} & \neg\text{Negative (}\textit{not bad}\text{)} \\ \neg\text{Positive (}\textit{not good}\text{)} & >_{polarity} & \text{Negative (}\textit{bad}\text{)} \\ [\neg]\,\text{StronglyPositive (}\textit{[not] fantastic}\text{)} & >_{polarity} & [\neg]\,\text{Negative (}\textit{[not] horrible}\text{)} \\ \neg\text{Neutral (}\textit{not adequate}\text{)} & =_{polarity} & \text{Negative (}\textit{bad}\text{)} \end{array}\right].$$

Intensity Shifters

A second type of shifters are *intensifiers*[9]. They either strengthen or weaken the prior intensity of a sentiment expression. We distinguish between *amplifiers* (strengthen) and *downtoners* (weaken). Although most intensifiers are adverbs (e.g., "very", "really", or "hardly"), they are not restricted to this part of speech (see also Section 6.2.3). Sentence (4.1) illustrates this concept: It contains two intensifiers. The adverb "utterly" amplifies the intensity of the sentiment expression "disappointed". Being "utterly disappointed" conveys a stronger negative sentiment than just being "disappointed". In the same manner, the intensifier "really" is used in the second clause of the sentence to amplify the (negated) sentiment expression "like". Sentence (4.4) contains an example for a downtoner. The adverb "somewhat" weakens the (prior) intensity of the sentiment expression "long".

Certainty Shifters

The third type of shifters affects the sentiment expression's *degree of certainty*, i.e., the confidence, belief, or commitment the author attributes towards his sentiment expression. We distinguish between *solidifiers* and *softeners*[10]. A solidifier reflects an increase in the author's certainty with respect to the sentiment expression. Consider the following sentence:

(4.6) Without question, this hotel is a jewel.

The phrase "without question" acts as a solidifier for the sentiment expression "jewel". Typical words or phrases that function as solidifiers are *comment adverbs* [163, p.156], such as "truly", "definitely", "surely", or other phrases, such as "convinced", "confident", or "without doubt". A softener decreases the degree of certainty. Sentence (4.4) contains an example. The word "appears" lowers the author's confidence in his negative opinion (namely, that the battery life is reduced). Other words and phrases that are often used as softeners are for instance "seems", "might", "may", "probably", or "seemingly".

[9] see also Carter and McCarthy [67, pg. 908] or Polanyi and Zaenen [303]

[10] Observe that, opposed to Rubin [327], who classify an utterance into five levels of certainty, we only consider a ternary decision ("solidified", "softened", or "no modification").

Neutralizer

This shifter type refers to words and phrases that set the sentiment expression into *irrealis mood*, effectively "neutralizing" it:

(4.7) This hotel really would be a jewel.

In this example sentence the modal operator "would" cancels out the positive appraisal of the hotel. Due to the conditional mood, the hotel is only hypothetically a "jewel". Other modal operators, such as "could", "should", or "ought to", may function as neutralizers too. Also subjunctive forms, typically indicated by the word "if", may have this effect (e.g., "If the hotel is great, I will stay longer."). Further indicators for the irrealis mood are words and phrases that express wishes, hopes, and expectations of the author.

4.3.4. Limitations of the Expression Level Model

Sentiment Sources

We do not model sentiment sources (opinion holders) as it is our belief that knowing the source of a sentiment expression is irrelevant when analyzing product reviews. First, the phenomenon that other persons' opinion is expressed is relatively seldom. In a preliminary study we analyzed thirty randomly chosen reviews[11] in the digital camera domain (294 sentences) and thirty randomly chosen reviews[12] on hotels (378 sentences). This study revealed that three sentences in the digital camera set and only a single sentence in the hotel reviews contain a different source than the author[13]. The second aspect that supports our belief is the observation that for the task of summarizing product reviews knowing the concrete sentiment source is of little value. No information need exists in this direction.

Comparisons

Besides discussing the advantages and disadvantages of a product in isolation, a reviewer may also compare it to previously experienced products or to the product's competitors[14]:

(4.8) I've had a Nikon D80 for over 2 years and take great pictures with it in all settings.

(4.9) The Nikon Coolpix s360 is a very bad camera as well, slightly better than this Canon.

The first sentence is an example for a comparison with a product experienced in the past. Here, the positive sentiment ("great pictures") cannot be attributed to the entity which is reviewed. The author introduces another product entity (Nikon D80) apart from the reviewed entity and comments on one of its aspects. Also no comparative expression (e.g., "greater picture quality than...") is explicitly used. Sentence 4.9 on the other hand, contains an explicit comparison using a comparative expression ("better than"). The underlying problem with comparisons is that mentioned sentiment targets can no longer be reliably attributed to the reviewed entity; they may in fact refer to other product entities. We decided not to model comparisons. In effect, we assume that a review always deals with a single entity alone[15].

[11] downloaded from http://www.epinions.com
[12] downloaded from http://www.tripadvisor.com
[13] We did not count sentences where the source is in the first person plural, e.g., "To tell the truth, we were pretty disappointed with the hotel."
[14] both sentences are taken from a review of Amazon user *Lillian36* (grammatical mistakes not corrected), see http://www.amazon.com/review/RN0F47SL1XWHE
[15] Take note that this is closely related to our decision not to model coreference structures.

Coreference

Mentions corefer if they refer to the same entity. A *coreference chain* is the set of expressions that all refer to the same entity. In the presented expression level model we are not interested in capturing complete coreference chains. We only model pronominal relations between two expressions and that only if the pronoun is a sentiment target. We do not model coreference relations between noun phrases or names:

(4.10) I recently bought the new Canon XYZ.

(4.11) I really like the camera, it's just great.

In these example sentences all underlined mentions corefer to the conceptual entity "Canon XYZ". In sentence 4.11 the mentions "the camera" and "it" are sentiment targets and *anaphoric*. The mention "the camera" is an example of a (definite) noun phrase that references another entity. We therefore disregard the relation between "camera" and "Canon XYZ" and only consider the pronominal anaphoric relation between "it" and "camera". Our motivation to only capture pronominal relations is to reduce the complexity of the model.

Semantic Relations between Sentiment Targets

In Section 4.1 we pointed out that product aspects can be are arranged in a hierarchy and defined a product type taxonomy that associates different aspects by means of semantic relations. We can observe that these relations also become manifest on the textual surface of a customer review. For example, in sentence (4.3) the verb "has" expresses the "feature-of" relation between the mention "flash" and the two mentions "reach" and "coverage". In contrast to Kessler et al. [209], we do not model semantic relations on the textual surface level. It is not our goal to derive such relations from textual evidence, but assume that they can be externally modeled as part of a knowledge base such as our product type taxonomy.

4.4. Related Work

Throughout the chapter we introduced related work when indicated, focusing on the specific aspects discussed. We now take a broader perspective and set the modeling of opinion expressions in reviews into a wider context. At the most fundamental level, one can distinguish between *psycholinguistic models* [83, 286, 287, 329] and *computational models*. Since we take the perspective of an engineer, we are primarily interested in computational models. We roughly categorize related work in terms of their primary perspective, distinguishing approaches in linguistics, statistics, and information extraction.

4.4.1. Linguistics

Private State

In her PhD thesis [418] and in a subsequent article [419], Wiebe presents a computational model of recognizing subjective sentences in narrative texts. Her notion of subjectivity and opinion stems from the concept of *private state*. This concept is central to a whole line of work following these initial publications (cf., Bruce and Wiebe [57], Wiebe et al. [416], Wiebe [420], Wilson and Wiebe [432, 435], Wilson et al. [436, 438]) and forms the basis for the construction of the widely used "multiple perspective question answering" (MPQA) corpus [413, 417] (we cover the characteristics of this corpus more closely in Section 5.4.1).

A private state, as introduced by Quirk et al. [309], is a very general term defined as a "state that is not open to objective observation or verification". Due to this broad definition, it covers concepts

such as opinions, beliefs, thoughts, emotions, evaluations, or judgments. Wilson and Wiebe [435] view private states with respect to their functional components, i.e., as states of "experiencers" (sentiment sources in our notion) holding "attitudes" (sentiment expressions) toward "targets" (sentiment targets). Besides this decomposition in functional components, they distinguish between three types of private state expressions in text, namely *explicit mentions*, *speech events*, and *expressive subjective elements* (see also Banfield [27]). Each type of private state expression is represented by a span of text and is associated with one or more sources (the persons that express or experience the private state), a single (optional) target (what the private state is about) and the expressed attitude. Attitudes are attributed with *type* (e.g., positive/negative attitude, positive/negative arguing, positive/negative intentions, or speculation), *intensity* (low, medium, high, or extreme), and *markers* such as whether the attitude is conveyed through sarcasm, repetition, or contrast.

This model and the resulting annotation scheme for the MPQA corpus of opinion annotations has been extended in a series of subsequent work. For example, Wilson et al. [438] extend the model to recognize contextual polarity, i.e., the polarity of a private state in the context of sentiment shifters. Somasundaran et al. [355] additionally consider discourse-level elements by examining associations between sentiment targets. They distinguish between "same target" relations, which reflect that targets refer to the same evocation of an entity (e.g., anaphora, part-whole, or instance-class relations) and "alternative target" relations, which represent contrastive targets.

Our expression level model adopts some central aspects of the previously cited line of work, namely the functional decomposition of attitude expressions into sentiment expressions and related sentiment targets (although we allow multiple targets to be related to a single sentiment expression). Opposed to them and for the named reasons, we do not cover the experiencer (sentiment source) of a private state expression. We also do not consider speech events as they are practically not existent in review documents. Also in contrast to them, we do not explicitly model the different aspects covered by the term private state (e.g., emotion vs. judgment). In that sense, our notion of private state is more shallow. A further difference is that we capture the constituents of sentiment intensity or degree of certainty in more detail by modeling sentiment shifter types, such as amplifiers, downtoners, solidifiers, or softeners. Similar to them, we model discourse level associations between targets: in our case, anaphoric relationships.

Appraisal Theory

The *appraisal system*, as proposed by White and Martin [409] and developed within the tradition of *Systemic Functional Linguistics* [158], is concerned with linguistic realization of opinions and attitudes within text. Attention is paid to "the means by which writers/speakers positively or negatively evaluate the entities, happenings and states-of-affairs with which their texts are concerned" [409, p.2]. At the most basic level, the appraisal framework distinguishes between *attitude*, *engagement*, and *graduation*. These three subsystems are modeled as orthogonal dimensions that operate in parallel. The attitude subsystem deals with the expression of "private states" and is subdivided into the three categories *affect* (emotions: expressing positive and negative feelings), *judgment* (ethics: evaluation of a person's behavior) and *appreciation* (aesthetics: evaluation of "things", including man-made and natural phenomena). Attitudes are attributed with positive or negative orientation. The engagement dimension is concerned with linguistic constructions, such as modality, polarity[16], or comment adverbs, that are used by speakers/writers to position themselves with respect to the expressed attitude. The graduation subsystem considers the resources authors use to grade the strength of their evaluations. With respect to attitudes, graduation refers to adjusting the degree of positive or negative orientation (sentiment intensity). On the other hand, with regard to engagement, graduation addresses the author's degree of confidence, commitment, or belief (degree of certainty).

[16] In this context *polarity* refers to the grammatical notion, i.e., marking of affirmative or negative contrasts in a clause.

The most evident differences to the scheme developed by Wiebe et al. [417] are the following: No explicit decomposition into functional components such as source, expression, or target is conducted. Also no distinction is made with regard to different types of expression (explicit mentions, speech events, or direct subjective elements). On the other hand, the appraisal framework goes into more detail on the attitude dimension (describing private states) by separating affect, judgment, and appreciation. In general, both models describe similar phenomena (e.g., sentiment orientation, sentiment intensity, or degree of certainty), but the appraisal framework provides a more fine-grained structure by separating along the three axes attitude, engagement, and graduation.

Based on the notion of appraisal theory, computational models and annotation schemes have been developed in the context of sentiment analysis: For example, Taboada and Grieve [373] present a model for appraisal-based review classification. They enrich a document level model, capturing the semantic orientation of reviews, by incorporating the three different types of attitudes (affect, judgment, and appreciation). They hypothesize that adjectives can be attributed with an "evaluative potential" with respect to each of the three types of attitude (e.g., "afraid" has an affect potential of 0.6, a judgment potential of 0.3, and an appreciation potential of 0.1). In compliance with Whitelaw et al. [411], Bloom et al. [45] define an *appraisal expression* as "a textual unit expressing an evaluative stance towards some target". They regard an appraisal expression as a frame that is filled with attributes such as the source, the target, the type of attitude, and the orientation. Their model also covers graduation and polarity (in terms of affirmative/negative). In accord with our expression level model, they postulate the existence of domain dependent target taxonomies, however they do not model them along generic semantic relations such as "part-of" or "type-of". A further work is by Read et al. [316], who conduct an annotation study in which book reviews are annotated according to the appraisal framework. They annotate "appraisal-bearing" terms with the type of attitude, engagement, and graduation.

4.4.2. Information Extraction

Models considered in this section have in common that sentiment analysis is considered as an information extraction task. The presented models typically eschew a general theory of opinion or emotion, but mostly rely on the functional decomposition that is formulated by the private state theory.

Relatively early and influencing work is due to Hu and Liu [177]. They model sentiment analysis as extracting triples that consist of slots for an *opinion target*, an *opinion passage*, and the *semantic orientation* of the opinion passage. In their model each unique opinion target (a concept or entity) is represented by a finite set of words and phrases that are considered to be synonyms. They distinguish explicit and implicit targets, as well as explicit and implicit opinion passages. Although they recognize the hierarchical representation of product types, their actual model is flat. They do not consider sentiment intensity or general sentiment shifters, but recognize negation using a simple polarity-flip model. The annotation scheme developed for evaluation purposes in Hu and Liu [177] is even simpler than the actual model. Their corpus contains only annotations for targets and polarity. Annotations are at sentence level and do not mark explicit text spans. The triple-based model is also adapted by Popescu and Etzioni [304].

The models most closely related to our expression level model are due to Toprak et al. [386] and Eckert et al. [109]. Toprak et al. present an annotation study of user-generated discourse, namely customer reviews of online universities and online services. Similar to us, they model reviews on two levels of granularity. On a coarse-grained level they differentiate sentences with respect to their relevancy towards the overall topic of the review and whether it contains an opinion on the topic (opinions may be explicit or polar facts). At this level they do not mark the concrete topic of the sentence, nor the polarity of explicit opinions (polarity of polar facts is marked as positive, negative, or both). Discourse functions (refer to Section 4.2) are also not considered. Their fine-grained model

operates on the expression level. The basic slots that need to be filled are *target*, *holder*, *opinion expression*, *polarity*, and *strength*. They further consider sentiment shifters in terms of negators, amplifiers, and downtoners. Targets may be related via an anaphoric link, but other semantic relationships are not considered.

Eckert et al. [109] and Kessler et al. [209] also present an annotated corpus of customer reviews (automobile reviews). Their annotation scheme is the most complex one which we can find in the literature. Central to their model are entities and their relations. Their model captures entities regardless of whether they are part of a sentiment expression or not. For each identified entity the complete co-reference chain is considered. Entities are typed by means of a subset of the ACE[17] mention types (extended by domain specific types) and may be interrelated via *refers-to* (co-reference), *part-of*, *feature-of*, *instance-of*, *member-of*, or *accessory-of* links. Entities may be the target of a sentiment expression, which are considered single or multi-word phrases. Each sentiment expression is attributed with a prior-polarity (similar to us defined as target-specific) and may be modified by shifters that alter the sentiment intensity (strengthen or weaken), the polarity (negators) or the degree of certainty (increase, decrease, or neutralize). In addition to our expression level model, they consider an optional opinion holder and model comparisons explicitly.

4.4.3. Probabilistic Topic Modeling

The line of work we cover in this section stems from research in topic modeling. In general, a *probabilistic topic model* is a statistical model that is designed to recognize hidden (latent) topics underlying a collection of text documents. Writing a document is modeled as a *probabilistic generative process* in which an imaginary author first chooses the topics he is going to write about (a distribution of topics is drawn) and then samples the individual words of the document from different *language models* that are associated with each topic. Topic modeling approaches are generally unsupervised and therefore scale well to large datasets. In the following we describe the statistical models on a rather high level — more detailed information on some approaches relevant to our work is provided in Chapter 8.

When regarding product aspects as coarse-grained, abstract topics (like in our discourse oriented model), topic modeling approaches represent a natural fit. It is therefore not surprising that in recent years a multitude of approaches have been proposed to adapt topic modeling algorithms to incorporate the sentiment analysis dimension. Studies mainly differ in the way they incorporate opinion expressions and how the basic model for topic inference is adapted. One of the earliest works in this direction is due to Mei et al. [259]. They extend the basic mixture model of topics by representing sentiment as two additional language models (positive/negative sentiment) that are orthogonal to the topic-related language models[18]. In the probabilistic generative process, a word is either sampled from one of the topic-related language models, the positive sentiment language model, or the negative sentiment language model. The major disadvantage of this specific model is that due to the strict separation of topic and sentiment language models, the interdependency between topic and sentiment is not recognized. The model is not capable of capturing target/topic specific polarity.

Lin and He [233] and Jo and Oh [194] propose a joint model, integrating topic and sentiment in a single language model. Whereas in standard models a single distribution of topics is associated with one document, in their model a document is associated with multiple such distributions, each corresponding to a sentiment label (e.g., positive or negative). Under this model a word is sampled from a distribution over words that is conditional on topic and on the sentiment label — that is, topic words and opinion words are drawn from the same distributions. Zhao et al. [465] suggest to explicitly separate opinion words from topic words by introducing additional language models, namely a general opinion model and aspect-specific opinion models. In the generative process, indicator vari-

[17] The Automatic Content Extraction (ACE) program [87]
[18] In fact, they also introduce a background language model for common English terms.

ables determine whether a word is "generated" by a topic-related[19] language model or by one of the opinion models.

Compared to models centered around (psycho-) linguistic theories as discussed in the previous sections, statistics-driven approaches naturally have a very shallow understanding of opinion. The majority of models regard opinion simply as a polarity label (positive vs. negative) associated with a word; sentiment intensity, for example, is generally not considered. Further, due to the bag-of-words assumption (i.e., word ordering is ignored) that underlies each of the presented models, sentiment shifters, such as negators or neutralizers, cannot be captured. For example, such models will probably predict positive sentiment for a phrase "not good camera" and predict the sentiment for "not bad camera" as being negative. Models differ with regard to how they represent target-specific polarity. Whereas Mei et al. [259] do not consider this interplay at all, Lin and He [233] or Jo and Oh [194] address this issue. A further distinction is the way product aspects (topics) are "generated". Whereas Jo and Oh [194] assume that all words within a single sentence are produced by the same underlying topic, other models allow multiple topics to be associated with a sentence. Titov and McDonald [382] further differentiate between local and global topics and propose a *multi-grain topic model*.

In general, considering only the expressiveness and disregarding the concrete implementation as statistical models, the presented approaches can be best compared to our perspective taken with the discourse oriented model: We also assume the existence of a set of topics (representing coarse-grained product aspects), that a review is based on a mixture of these topics, and have a shallow understanding of opinion (represented as positive, negative, both or none).

4.5. Summary

In this chapter we presented our perspective on modeling the expression of sentiment in customer reviews. We first introduced our notion of a product aspect. In particular, we distinguished the three terms product type, product, and product aspect and differentiated between coarse-grained (*concept level*) and fine-grained (*mention level*) aspects. We proposed to hierarchically structure aspects in a **product type taxonomy**, which we primarily motivated by the observation that such a model helps to structure and summarize detected sentiment in a comprehensive form.

As with regard to modeling product aspects, we also distinguished a coarse-grained and fine-grained perspective concerning the actual expression of sentiment. The **discourse oriented model** operates on the coarse-grained level, disregarding individual linguistic constituents of opinion. It basically assumes that a review can be decomposed into segments that are coherent with respect to the three dimensions *discourse function*, *topic*, and *sentiment polarity*. We primarily motivated this type of model by pointing out an information need with regard to extracting sentiments on the concept level, but also by the observation that certain features (namely *polar facts* and *implicit mentions*) are difficult to capture on a more fine-grained level.

With the **expression level model** we introduced a perspective that allows to capture the individual constituents of sentiment expressions. It mainly resembles the functional components of opinion expressions as introduced in the theory of *private states*. The model basically distinguishes the constituents *sentiment target* and *sentiment expression* and postulates a many-to-many relation between them. Sentiment targets either directly refer to a mention of a product aspect or indirectly via a pronominal relationship. Sentiment expressions are attributed with a *polarity* and *intensity* value and may be modified by multiple *sentiment shifters*. Also the expression level model is primarily motivated by an information need. We believe that being able to extract and summarize aspects and related sentiments on the fine-grained mention level additionally provides valuable information to the users of a customer review mining system.

[19] Similar to Mei et al. [259], they include a background language model. Additionally, they distinguish between a global aspect model and T local aspect models (with T being the predefined number of aspects).

Language is a part of our organism and no less complicated than it.

Ludwig Wittgenstein

5. Datasets and Annotation Schemes

Annotated corpora form a crucial basis for nearly all natural language processing (NLP) tasks. The process of creation as well as the analysis of such datasets helps to understand and to quantify the characteristics of a specific task. Furthermore, annotated corpora are fundamental to evaluation and mistake analysis of concrete approaches. In case of supervised machine learning approaches, they also constitute an obligatory requirement for the implementation (namely the training phase).

Whereas for other NLP related tasks such as information extraction widely accepted evaluation corpora are available (e.g., the MUC[1] and ACE[2] datasets), no comparable corpora exist for the task of aspect-oriented sentiment analysis. Regular and prominent academic competitions, such as the ones held as part of the MUC or ACE program, have not yet been established within the research field of sentiment analysis (with the exception of the TREC 2006-2008 Blog Tracks [243, 288] and tracks at the NTCIR-6/7/8 workshops [339, 340] where *sentiment retrieval systems* were evaluated). This lack is mainly due to the relative novelty of sentiment analysis as an area of research, its diversity of objectives, and the heterogeneity of the researchers' backgrounds and scientific communities. With regard to the task of aspect-oriented sentiment analysis of user-generated content, the current situation is that only few hand-annotated corpora exist and among these, the granularity as well as the schemes of annotation differ widely. In view of these facts and for the purpose of attaining evaluation datasets that exactly fit our needs, we chose to compile our own hand-annotated corpora from scratch.

The following Section 5.1 characterizes the nature of our datasets. We outline our motivation to examine two different genres of reviews and provide basic descriptive statistics about the datasets we have compiled. Subsequently, Sections 5.2 and 5.3 describe the two annotation schemes we have derived from the discourse oriented and the expression level model of opinion expression. The chapter concludes with a review of related corpora in Section 5.4.

5.1. Dataset Characteristics

All the experiments we present in the following chapters are conducted on two different datasets we sampled from a web crawl of prominent customer review sites. We crawled a document collection of 417,170 hotel reviews from the travel website Tripadvisor.com and 180,911 digital camera reviews from the online retailer Amazon.com as well as the review aggregation websites Epinions.com and Buzzillions.com. The reviews represent typical user-generated content; they are all written by customers/users instead of being authored by any professional editor. Texts exhibit a rather informal style — they often lack correct English grammar, exhibit the use of slang words, and contain an above-average amount of misspellings. Our web crawls are monolingual, all crawled documents are written in English.

Our primary motivation to conduct experiments on two distinct datasets is to increase the significance and reliability of our evaluation results. In particular, being able to compare results obtained from two different domains allows us to analyze the generalizability of derived assertions more soundly. It prevents us from drawing conclusions based on phenomena that might be inherent

[1] Created to compare competing information extraction systems at the Message Understanding Conferences (MUC) [151, 275], which were held during the 1990s.

[2] The Automatic Content Extraction (ACE) program [87] is a successor of the MUC conferences and has also established several corpora for the evaluation of information extraction systems.

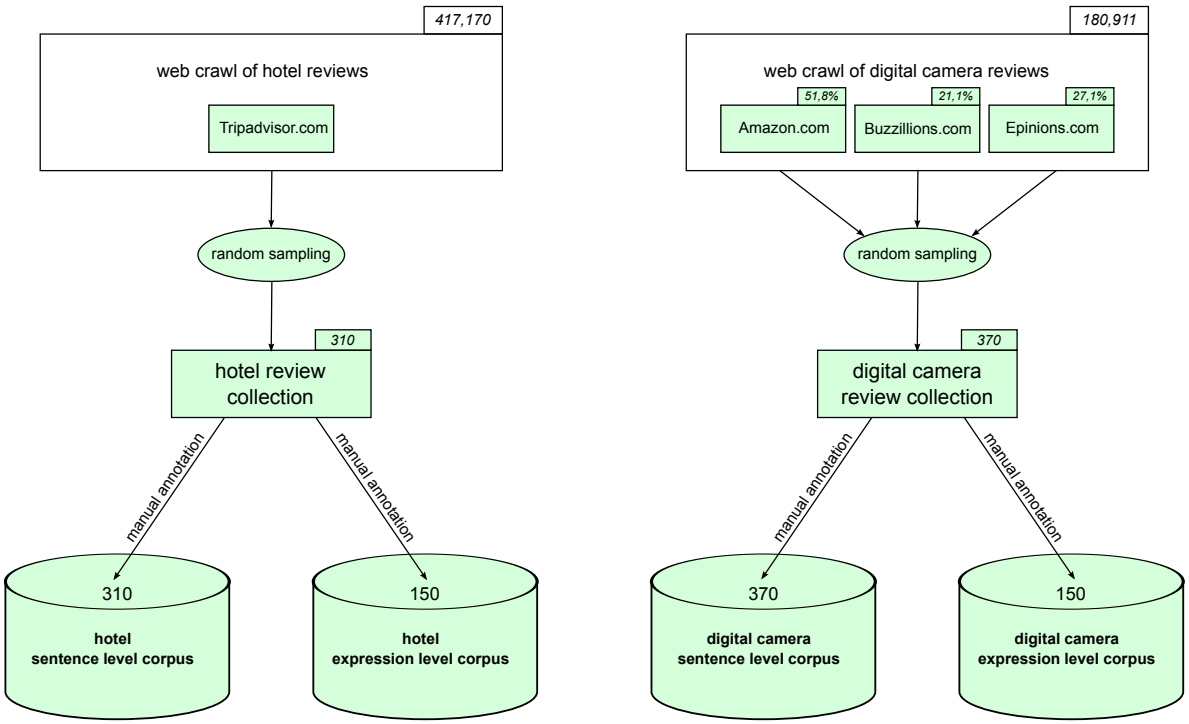


Figure 5.1.: Basic information about the customer review datasets used within the thesis.

only to one specific domain. Furthermore, we are able to examine the cross domain applicability of the models developed in the preceding chapter.

We select the specific domains of hotel and digital camera reviews with the following intentions in mind: First, they represent two quite distinct genres of customer reviews, namely reviews of products (e.g., cars, mp3 players, refrigerators) and reviews of services (e.g., restaurants, hairdressers, health clubs). Among both genres, digital cameras and hotels are very popular targets of customer reviews. Second, both domains have been considered in other, related studies, which makes our results more comparable. And third, as a very practical consideration, due to the relative popularity of the selected domains, it is easier to crawl huge collections of review documents.

From the acquired web crawls we randomly sampled a set of 310 hotel reviews as well as a set of 370 digital camera reviews. Based on these two collections, we have created four different, hand-annotated text corpora: Each of the two collections is manually annotated according to the two different annotation schemes that we present in the next sections. Whereas on the sentence level we annotate the complete sample sets of 310 and 370 reviews, we only annotate subsets of 150 documents each on the expression level. The decision to restrict the corpus size is simply due to the massively higher effort coupled with annotating on the expression level. Figure 5.1 illustrates this setup.

The collections exhibit the statistics described in Tables 5.1 and 5.2. Both datasets are composed of roughly 3,500 sentences and 60,000 tokens. The average length of a review is very similar in both domains, while hotel reviews tend to be slightly longer (11.2 versus 9.4 sentences per review). Also the average length of a sentence is similar with 17.4 and 17.5 tokens[3] per sentence. The shortest reviews in both collections are composed of a single sentence only and the shortest sentences consist of a single token. An important observation that was also pointed out in several other studies [92, 96, 136] is that the distribution of reviews is significantly skewed towards positive reviews. Table 5.2 lists the share of reviews in the hotel and camera collections broken down into user ratings (with 5 being the best rating). Counting reviews with ratings 1-2 as negative and reviews with ratings 4-

[3]tokenized by StanfordCoreNLP tokenizer (http://nlp.stanford.edu/software/corenlp.shtml)

statistic	hotel	digital camera
Reviews	310	370
Sentences	3476	3493
Avg. sentence/review	11.21	9.44
Std. sentence/review	8.08	13.17
Min. sentence/review	1.00	1.00
Max. sentence/review	62.00	128.00
Tokens	60456	61076
Avg. tokens/sentence	17.39	17.49
Std. tokens/sentence	10.51	10.01
Min. tokens/sentence	1.00	1.00
Max. tokens/sentence	92.00	73.00

Table 5.1.: Descriptive statistics of the annotated review corpora.

rating	hotel	digital camera
1	12.6%	7.5%
2	9.3%	4.2%
3	15.3%	10.6%
4	29.7%	31.5%
5	33.2%	46.2%

Table 5.2.: The distribution of reviews with ratings from one to five stars.

5 as positive, we can observe that in the hotel collection nearly three times more positive reviews exist than negative reviews. In the digital camera collection the skew is even more pronounced. We observe around 80% positive reviews versus around only 10% negative reviews. Take note that this imbalance is less pronounced in both corpora when examining the distribution at the sentence or expression level. We will discuss these numbers and implications in more detail in Chapter 6.

In the following two sections we give an overview of the annotation schemes we derived from the proposed discourse oriented and expression level models and briefly outline the basic annotation process. The annotation schemes slightly adapt and extend the original models, mainly with additional features of exploratory nature. For a detailed discussion of the annotation process we refer to the annotation guidelines presented in Appendix A. They cover many exemplary annotations and give a deeper insight into the content of the created corpora.

5.2. Sentence Level Annotation Scheme

We implement the discourse oriented model at the sentence level — that is the unit of annotation is a single sentence. At first sight this decision might seem contradictory to the notion of discourse we introduced earlier ("language phenomena beyond the sentence level"). However, we believe that this approach is adequate for the following two reasons: First, the process of hand-annotation becomes much easier and is more efficient. Segment boundaries (i.e., sentence boundaries) can be automatically detected and pre-annotated with very high accuracy. Thus, in the majority of cases the annotator is relieved from setting boundaries manually — only in case of error he needs to revise the span of an annotation. Second, by annotating on the sentence level, we do not lose any expressiveness of the discourse oriented model. Discourse segments spanning multiple sentences can be automatically reconstructed by combining consecutive, identically annotated sentences.

Type «DiscourseSegment »

attribute	usage	valid values
*DiscourseFunction**	Mandatory for each sentence, defines the discourse function.	exactly one of the functions defined in Table 5.4
SentimentPolarity	Defines the polarity value if the sentence expresses an opinion or states a polar fact.	`positive`, `negative`, `neutral`, `both`, or `EMPTY`
Topic	Comma-separated list of the topics covered by the sentence.	domain dependent, one or more of the topics defined in Tables 5.5a and 5.5b, or `EMPTY`
PolarFact	A flag indicating whether the sentence expresses a polar fact.	`true` or `false`
NonFocusEntity	A flag indicating whether the sentence targets an entity which is not in the primary focus of the review.	`true` or `false`
Irrealis	A flag indicating whether the sentence refers to an irrealis event.	`true` or `false`
Confidence	A flag indicating whether the annotator is uncertain about correct annotation.	`low` or `EMPTY`

Table 5.3.: Attributes of the discourse segment annotation type.

5.2.1. Definition of Discourse Functions and Topics

The discourse oriented model postulates that, independent of the addressed product type, a fixed set of discourse functions is universally applicable for capturing the "functional" dimension of customer reviews. We obtained an initial set of discourse functions as part of a preliminary study. We analyzed and preliminarily annotated 30 reviews in the hotel, digital camera, and mp3 player domain each (i.e., 90 reviews in total). The initially devised set consisted of 20 different functions, but was further refined during the final annotation process. Functions such as "Wish" were removed as they either occurred very rarely or correlated too closely with other functions and thus were very hard to distinguish. As we will learn below, the final set consists of 16 distinct functions.

The predefined sets of topics are dependent on the product type. We utilized unsupervised, probabilistic topic modeling to find the relevant topics. For both product domains, the topic inference process was applied on collections of 50,000 documents. Manual analysis and revision of the automatically derived topics provided the final set. Appendix D describes this process in more detail.

5.2.2. Annotation Scheme

The proposed scheme resembles the discourse oriented model and supplements the model's basic constituents by a set of additional attributes that we believe are helpful from a practical point of view. The scheme is quite simple, consisting of a single type of annotation which we meaningfully denote as «*DiscourseSegment*». Table 5.3 summarizes the attributes related to a discourse segment. Attributes which are marked with a star symbol are mandatory.

Discourse Function

Each sentence is annotated with exactly one annotation of the *DiscourseSegment* type. The type's main properties are the *DiscourseFunction*, the *SentimentPolarity* and the *Topic* attribute. Defining the "discourse function attribute" is mandatory for each sentence. The annotator is limited to choosing a function from a set of predefined discourse functions. The set covers $16 + 1$ named discourse

function	description
Advice	The reviewer gives an advice on using the product or how to circumvent a particular problem.
Comparison	The reviewer compares the product with another, similar product.
Conclusion	The reviewer concludes and summarizes his comments and potentially expresses a recommendation.
Expectation	The reviewer describes what he expects from the product.
Fact	The reviewer provides factual information about the product or its aspects (may be a polar fact).
General Remark	The reviewer provides some general remarks that apply to the whole class of products which is reviewed.
Lack	The reviewer mentions the absence of a feature or part of the product.
Other Review	The reviewer comments on other reviews of the product.
Personal Context	The reviewer provides personal information, e.g., that he is a professional photographer.
Problem	The reviewer describes a problem that is encountered with the product or one of its aspects.
Purchase	The reviewer mentions his personal procedure in purchasing the product, e.g., where and when he bought the product.
Requirement	The reviewer describes his requirements with regard to the product or its aspects.
Section Heading	The reviewer uses a heading to structure the review.
Sentiment	The reviewer explicitly expresses his opinion or feelings.
Summary	The reviewer summarizes the advantages or disadvantages of the product.
Usage	The reviewer describes a concrete situation when using the product, potentially commenting on his experience.
OTHER	Any other discourse function that is not listed above.

Table 5.4.: List of predefined discourse functions with short descriptions. The functions are listed in lexicographic order.

functions which are tightly related to the domain of customer reviews, but are independent of any concrete genre. For the case that a sentence does not match any of the predefined functions, the annotator chooses the special value named `OTHER`. The set of predefined discourse functions, each complemented by a short description, is presented in Table 5.4. For detailed examples we refer to the annotation guidelines provided in Appendix A.2.

Sentiment Polarity

For each sentence the annotator must decide whether the reviewer expresses an opinion or whether the sentence contains rather factual information. Opinion expressing sentences or sentences stating a polar fact are annotated by setting the value of the "sentiment polarity attribute". The annotator marks the sentence as either containing predominantly positive, negative, or neutral opinions. Sentiment intensity is not graduated. In case a sentence contains mixed polarity, i.e., positive and negative comments, the polarity attribute is set to `both`. For sentences not expressing any opinion the attribute is left empty.

Topic

If the reviewer refers to the product or one of its aspects, such a sentence is denoted as *on-topic*. In that case, the annotator fills the "topic attribute" by selecting one of the predefined topics compiled for the hotel review and digital camera review domains. If the sentence covers multiple different topics, all associated topics are enumerated by means of a comma-separated list. In case the sentence is *off-topic*, that is, it cannot be associated with one of the predefined topics, the topic attribute remains empty. As discussed in Section 4.1, topics can be hierarchically structured. We model a two-level hierarchy of topics, effectively distinguishing main topics and subtopics. The sets of predefined topics and subtopics for the hotel and digital camera domains are presented in Tables 5.5a and 5.5b. For the domain of hotel reviews we distinguish 10 main topics and in addition 12 subtopics. The number of topics in the camera domain is higher with 15 main topics and 13 subtopics. Appendix D explains how we derived the set of topics. Observe that in comparison to other studies, which for example only consider four or five distinct topics [40, 138], the coverage of our topic sets is much higher.

topic	**subtopics**
cleanliness	
decoration	
dining	breakfast
facility	elevator, recreation
internet	
location	security, parking
service	check-in/out
sleep quality	noise, bed
room	air conditioning, bathroom, room amenities, view
price	

(a) hotel domain

topic	**subtopics**
accessory	
battery	
connectivity	software
ease of use	user interface, user manual
features	face detection, image stabilization, underwater capability
flash	
memory	
optics	focusing, zoom
appearance	built quality, dimensions
picture quality	low-light performance
price	
screen/viewfinder	
settings	macro mode, scene modes
speed	
video recording	

(b) digital camera domain

Table 5.5.: List of predefined product aspects for the domains of hotel and digital camera reviews. The aspects are presented in lexicographic order.

Exploratory Attributes

In addition to expressing opinions explicitly, reviewers may evoke positive or negative impressions by characterizing certain facts regarding a product. Typically, these facts are concerned with desirable or undesirable properties of the product. The reader infers a positive or negative attitude by applying commonsense knowledge — for example, by knowing that "fitting into a pocket" is a desirable property for a digital camera. The annotator marks such sentences by setting the "polar fact attribute" to `true`. A polar fact sentence is recognized as expressing opinion and thus must be accompanied with a non-empty sentiment polarity attribute.

A typical customer review is explicitly related to a single product and describes its advantages and disadvantages. However, reviewers may also cite and evaluate other, similar products, e.g., for the purpose of comparison. Our goal is to manually identify such sentences in which the reviewer refers to products other than the one being in the primary focus. We provide the "non-focus-entity attribute" for this situation.

The presence of modal operators or other linguistic constructs may have a great influence on the perception of an opinion. If reviewers evaluate a property of a product in an unreal context, the polarity of the expression can be shifted or even neutralized. This is for example a typical case when they express expectations, wishes, requirements or make their evaluation conditional on something. The annotation scheme distinguishes sentences that refer to an irrealis event from sentences that refer to real events by means of the "irrealis attribute".

Annotating natural language text is typically a difficult task. Annotators will likely encounter cases where even provided guidelines cannot help to eliminate uncertainty about correct annotation. The scheme allows the annotator to express his uncertainty by setting the "confidence attribute" to `low`.

5.3. Expression Level Annotation

In this section we present the annotation scheme for the expression level model. It is concerned with the specific linguistic constituents of sentiment expressions and therefore consists of multiple annotation types. Each annotation marks a specific span of text and is automatically attributed with a (document-wide) unique ID. Spans of texts may cover multiple tokens, but are not allowed to cross sentence boundaries. The different annotation types are interrelated to each other and the scheme captures this aspect. Annotators use the annotations' unique IDs to relate one annotation to another.

5.3.1. Annotation Scheme

In accordance to the expression level model, we decompose an opinion into three major constituents, namely *sentiment expressions*, *sentiment targets*, and *sentiment shifters*. Each of these constituents is reflected by a separate annotation type. A sentiment expression annotation may refer to one or more sentiment targets and may be modified by one ore more sentiment shifters. The sentiment expression and the sentiment target annotations are existentially dependent on each other — the former cannot exist without the latter and vice versa. In other words, it is invalid to produce a sentiment expression annotation that has no target or to create a sentiment target annotation that does not target any sentiment expression[4]. A sentiment shifter annotation is existentially dependent on a sentiment expression, but a sentiment expression can occur without a referring shifter annotation.

To capture pronominal relations, the scheme offers a *sentiment target reference* annotation type. In addition, we introduce an annotation type *product aspect mention*, which is used to cover mentions of product aspects that are not part of a sentiment expression (i.e., that are not marked as a sentiment

[4]Our annotation tool chain enforces this requirement.

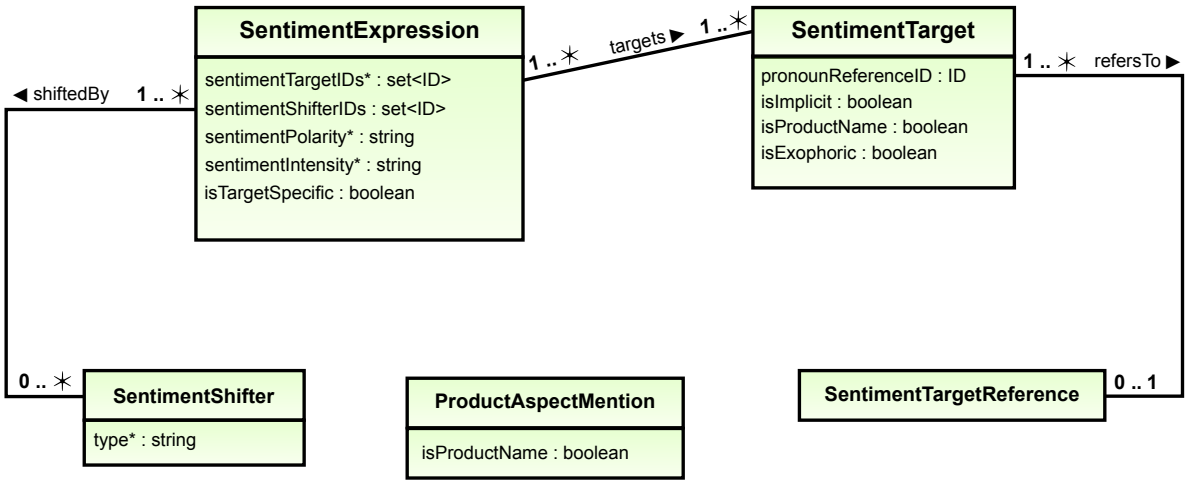


Figure 5.2.: The annotation scheme for the expression level model of sentiment.

target). Figure 5.2 gives an overview of the different annotation types and their relations. In the following, we discuss each type and its corresponding attributes in more detail.

Sentiment Expression

A sentiment expression is the central part of the expression level annotation process. It refers to the expression that explicitly evokes an opinion and sets up the prior polarity and intensity. This can be a single word or a whole phrase. All major parts of speech (adjectives, verbs, nouns, or adverbs) may act as a sentiment expression. In Table 5.6 we summarize the attributes which are defined by the sentiment expression annotation type. Again, attributes that are marked with a star symbol are mandatory.

Type «SentimentExpression »

attribute	**usage**	**valid values**
*sentimentTargetIDs**	Associates the sentiment expression with corresponding sentiment targets.	valid `annotation ids` of a sentiment target annotations
sentimentShifterIDs	Associates the sentiment expression with corresponding sentiment shifters.	valid `annotation ids` of sentiment shifter annotations
*sentimentPolarity**	Defines the polarity of the expression.	`positive`, `negative`, or `neutral`
*sentimentIntensity**	Defines the intensity of the expression.	`strong` or `average`
isTargetSpecific	Flag that indicates whether the expression is target-specific.	`true` or `false`

Table 5.6.: Attributes of the sentiment expression annotation type.

The attribute denoted as "sentiment target IDs" is used to link the sentiment expression to one or more sentiment targets. The annotator fills in the annotation ID(s) of the relevant sentiment target(s). In case multiple targets are involved, a comma-separated list of IDs is given. By allowing for such a list of IDs, the many-to-many relation between sentiment expressions and targets can be easily

modeled. Take note that the attribute is mandatory — a sentiment expression is defined to be related to at least one sentiment target.

In case a sentiment expression is modified by one or more sentiment shifters, the annotator interlinks the types by means of the "sentiment shifter IDs" attribute. Again, a list of IDs allows for easy implementation of the many-to-many relationship defined between sentiment expressions and shifters. The attribute is optional — a sentiment expression does not need to be modified by any shifter.

The "sentiment polarity" attribute refers to the *target-specific prior polarity* of the word or phrase that is annotated. The annotator decides whether the sentiment expression conveys a predominantly `positive`, `negative`, or `neutral` sentiment to its target. When making this decision, the sentiment expression is always considered in isolation from any sentiment shifter.

A sentiment bearing phrase exhibits an immanent sentiment intensity which is captured by the attribute "sentiment intensity". The annotator classifies the intensity of an expression into the two levels `average` and `strong`. An expression is only annotated as `strong` if it clearly stands out with regard to other expressions (refer to the annotation guidelines in Appendix A.3 for a more detailed distinction).

In Section 4.3 we discussed the phenomenon of target-specific prior polarity. Words or phrases may convey sentiment only in conjunction with a specific target or their polarity may flip with different targets. To measure this phenomenon we incorporate the exploratory attribute "isTargetSpecific" to the annotation scheme. The flag is set to `true` if the sentiment expression is target-specific, otherwise we leave the attribute empty.

Sentiment Target

The sentiment target annotation refers to a product aspect that is targeted by a sentiment expression. A target may become manifest as a single word, a compound noun, or a more complex phrase. The part of speech is variable, but most commonly, nouns function as sentiment targets. We summarize relevant attributes for this annotation type in Table 5.7. All attributes are optional.

Type «SentimentTarget »

attribute	usage	valid values
pronounReferenceID	If the target is a pronoun, associates it with the corresponding product aspect.	valid `annotation id` of a reference annotation
isImplicit	Flag that indicates whether the target is implicit.	`true` or `false`
isProductName	Flag that indicates whether the target is a product name.	`true` or `false`
isExophoric	Flag that indicates whether pronoun reference is exophoric.	`true` or `false`

Table 5.7.: Attributes of the sentiment target annotation type.

In case the sentiment is expressed towards a pronoun, the annotator marks the pronoun as sentiment target and lets the attribute "pronoun reference ID" point to the "sentiment target reference" annotation that identifies the referred product aspect. The annotation of pronouns is restricted to cases where the referred aspect is relevant for the examined product domain. Pronominal relations that take not part in a sentiment expression are also not considered by the annotation scheme.

In the special case when the pronoun refers to a target that is not mentioned within the document, a reference is called *exophoric*. Such references effectively "operate between the text and the external

world" [67]. The annotation scheme offers the "isExophoric" attribute of the sentiment target type to capture such a situation.

We distinguish between explicit and implicit mentions of product aspects. Whenever a sentiment target addresses a text span that implies, but does not explicitly represent a nominal mention of a product aspect, the annotator sets the "isImplicit" attribute of the sentiment target type to `true`. This is generally the case when the reviewer paraphrases a product aspect. Take note that the "isImplicit-flag" is not set if the sentiment target is a pronoun (we do not regard pronouns as implicit targets).

Despite nominal product aspects or implicit mentions, a *named entity* can constitute a valid sentiment target. We want to differentiate between these classes of sentiment targets and therefore introduce the flag "isProductName". Whenever a sentiment expression targets a product name, the attribute must be set to `true`. A product name may refer to the name of a producer (e.g., "Canon", "Samsung", or "Sony"), to a specific model (e.g., "EOS 550D", "ES80", or "DSC-W570B") or to a combination of both (e.g., "Canon EOS 550D").

Sentiment Shifter

The sentiment shifter annotation type refers to an expression that shifts the polarity, intensity, or certainty degree of a sentiment expression. A shifter may be composed of a single word or a longer phrase. All word classes may act as a sentiment shifter, but most commonly adverbs function as such. The sentiment shifter annotation takes a single mandatory attribute that defines the type of the shifter. The annotation type is outlined in Table 5.8.

Type «SentimentShifter »

attribute	usage	valid values
*type**	Defines the type of the sentiment shifter.	`Negator`, `Amplifier`, `Downtoner`, `Solidifier`, `Softener`, or `Neutralizer`

Table 5.8.: Attributes of the sentiment shifter annotation type.

The valid sentiment shifter types `Negator`, `Amplifier`, `Downtoner`, `Solidifier`, `Softener`, and `Neutralizer` represent exactly the types defined in Section 4.3 and are thus not discussed any further.

Sentiment Target Reference

Besides the three major constituents described previously, the expression level model considers simple co-reference relations. Whenever a sentiment target is a pronoun, the annotator identifies the corresponding referent, i.e., the entity the pronoun refers to and relates it to the target. For this purpose the annotation scheme provides the "sentiment target reference" annotation type. It is a simple "marker" annotation that has no further attributes. The relation between a pronominal sentiment target and its referent is annotated via the "pronoun reference id" attribute of the sentiment target annotation type. We restrict a sentiment target reference to exclusively refer to *nominal mentions* of product aspects.

Product Aspect Mention

The four presented annotation types *sentiment expression*, *sentiment target*, *sentiment shifter*, and *sentiment target reference*, together with their attributes and relations, implement the expression level model. Not part of the expression level model, but introduced as further markable, is the annotation

type *product aspect mention*: It refers to nominal or named product aspect mentions that are not directly targeted by any sentiment expression. Annotating product aspect mentions can be regarded as a secondary goal. The main purpose of introducing this additional annotation type is to enable a detailed evaluation and mistake analysis for the task of fine-grained product aspect extraction. The annotation stands on its own and is not related to any other annotation type of the expression level annotation scheme. Table 5.9 describes the product aspect mention annotation type. It provides the single, optional attribute "isProductName" that differentiates nominal from named aspect mentions.

Type «ProductAspectMention »		
attribute	**usage**	**valid values**
isProductName	Flag that indicates whether the mentioned aspect is a product name.	`true` or `false`

Table 5.9.: Attributes of the product aspect mention annotation type.

5.4. Other Available Datasets

In this section we provide an overview of other, publicly available datasets that explicitly target or are closely related to the task of aspect-oriented sentiment analysis. We are particularly interested in gold standard corpora which may serve as reliable evaluation datasets. We therefore only consider hand-annotated corpora more closely. Automatically extracted or boot-strapped datasets are not covered. We refer to Pang and Lee [294, chap. 7] who provide a broader review of available datasets for sentiment analysis.

Table 5.10 summarizes the attributes of the set of manually annotated and publicly available datasets that are most relevant in our context (we restrict our consideration to English-language corpora). The basic criteria we use to distinguish the presented corpora are the *task* the particular corpus has been designed for, the *granularity* and *expressiveness* of the applied annotation scheme, as well as the *domain* the annotated documents stem from.

5.4.1. MPQA Opinion Corpus

The *MPQA Opinion Corpus* is one of the most prominent and earliest annotated corpora in the field of sentiment analysis. It was initially devised in 2002 [412] for the task of multi-perspective question answering (MPQA) and was continuously extended to address several other tasks. As we discussed in Section 4.4.1, the underlying model is centered on the concept of *private state* [309] and the derived annotation scheme views these private states "in terms of their functional components — as states of *experiencers* holding *attitudes*, optionally toward *targets*." [417]. The annotation scheme reflects these functional components on the expression level. The original scheme [432] distinguishes between two different ways to express private states (*direct subjective* annotation frame and *expressive subjective* frame) and covers the source of such expressions by means of an *agent* frame. Private states are attributed with polarity and intensity information. The original corpus is augmented by Wilson et al. [438] to also contain information about the *contextual polarity* of direct subjective expressions. Wilson and Wiebe [435] further extend the scheme to cover *targets* of private states, as well as *attitude types* (distinguishing sentiment, agreement, intention, arguing, and speculation). Furthermore, Stoyanov and Cardie [361] add an annotation frame that covers the *topic* of sentiment targets. Co-referring topic frames are resolved and resulting clusters are labeled with a name.

source/name	task	granularity	annotation scheme	domain	language
NTCIR-6/7/8 [339, 340, 341]	(multilingual) sentiment retrieval	sentence	subjectivity, polarity, topic relevance, sentiment source (as string), sentiment target (as string)	news	EN,CN,JP
TREC 2006/07/08 Blog Track [243, 288, 289]	sentiment retrieval	document	subjectivity, polarity (ranking), topic relevance	weblog	EN
MPQA 2.0 [413, 417, 434]	multi perspective question answering	expression	subjectivity, polarity, intensity, sentiment source, sentiment target, attitude type	news	EN
Ding et al. [102], Hu and Liu [177]	aspect-oriented sentiment analysis	sentence	sentiment target/expression tuples (as strings) attributed with polarity and intensity	reviews (electronics)	EN
Ganu et al. [138]	aspect-oriented sentiment analysis	sentence	topic, sentiment polarity	reviews (restaurants)	EN
Zhuang et al. [468]	aspect-oriented sentiment analysis	sentence	sentiment target/expression tuples (as strings) attributed with polarity	reviews (movies)	EN
Toprak et al. [386]	aspect-oriented sentiment analysis	sentence & expression	polar fact, topic relevance, sentiment target (with anaphora resolution), sentiment expression (with polarity, intensity), sentiment shifter, sentiment source	reviews (education, services)	EN
JDPA Sentiment Corpus [209]	aspect-oriented sentiment analysis	expression	sentiment target (with co-reference resolution), sentiment expression (with polarity), sentiment shifter, entity relations, sentiment source	weblog (cars)	EN

Table 5.10.: Other manually annotated corpora for sentiment analysis.

Without doubt, the MPQA Opinion Corpus and the numerous studies which evolved around its development are most influencing within the field of sentiment analysis. A major share of subsequently devised corpora adapts the concepts which emerged in this line of work. With regard to the expression level model, we already pointed out in Section 4.4.1 that also the basic constituents of our model and annotation scheme can be attributed to this line of research. Concerning the discourse oriented model and scheme, we are aware that, for example, the recognition of polar facts is also addressed by Wilson [433] where the phenomenon is denoted as *objective polar utterance*. The annotation types covering neutralizers (expression level) or sentences with irrealis events (sentence level) are closely related to the "insubstantial feature" introduced in [417].

The most recent version 2.0 of the MPQA Opinion Corpus comprises 692 documents, containing in total 15802 sentences. The corpus mostly covers news text: It is composed of different subsets. The major share (539 documents) stems from the original corpus, which contains selected foreign and U.S. news stories from the years 2001 and 2002. Other subsets are based on articles from the Wall Street Journal (85 documents), the American National Corpus[5], and the ULA-OANC corpus[6]. We cannot directly make use of the MPQA Opinion Corpus for training or evaluation purposes as the target domain (news articles) and our application domain (customer reviews) are too different. However, to create our own corpus, we adapted some of the underlying concepts to match the customer review domain.

5.4.2. NTCIR and TREC Evaluation Campaigns

We know of two major evaluation campaigns that are also concerned with sentiment analysis. Both, the NII Test Collection for IR Systems (NTCIR) workshop series, as well as the Text Retrieval Conference (TREC) workshop series, are rooted in the information retrieval community.

The sixth NTCIR workshop introduced a multi-lingual opinion analysis task (MOAT), which was continued during NTCIR-7 and 8. The focus of the task is on *cross-lingual sentiment retrieval* and

[5] http://americannationalcorpus.org/
[6] http://nlp.cs.nyu.edu/wiki/corpuswg/ULA-OANC-1

opinion aware question answering. The provided evaluation corpora consist of news articles from Chinese, Japanese, and English-language news sources. The English-language subcorpus of the NTCIR-8 test collection comprises 150 documents and 6564 sentences in total. The corpus is annotated at the sentence level and provides information about the sentiment sources, sentiment targets, sentiment polarity, as well as the relevance of the sentence with regard to a set of predefined, very broad topics (e.g., "Cosovo civil war" or "nuclear weapons tests"). Sentiment sources and targets are provided as simple string-valued attributes of the sentence annotation and are not explicitly resolved by means of a separate annotation frame.

From 2006 to 2008 the TREC Blog track covered an opinion retrieval task. It follows the observation that blog related queries often express an information need based on "opinion, or perspective-finding nature, rather than fact-finding" [289]. The *Blog06 Test Collection*[7] which was used throughout the workshops is composed of about 3.2 million blog posts crawled over a period of three months and covers multiple preselected topics. The opinion retrieval task required participants to identify blog posts that express an opinion with regard to a specified target (named entity, event, or concept). Relevance assessments and subjectivity information (including polarity) are provided at the document level for pooled subsets of the submitted runs of the participants.

Both corpora, NTCIR and Blog06, can be regarded as aspect-oriented in the sense that they contain assessments about target relevance. However, the task (sentiment retrieval) is of different nature than ours: targets/aspects are defined broadly and span very diverse domains. Due to this and the fact that annotations are rather coarse-grained, both datasets are not sufficient to evaluate aspect-oriented customer review mining systems.

5.4.3. Customer Review Datasets

Hu and Liu Corpus of Consumer Electronic Reviews

Hu and Liu [177] have been the first to annotate a corpus of user-generated content for the purpose of aspect-oriented sentiment analysis. It consists of 113 customer reviews, spanning 4555 sentences and 81,855 tokens. The reviews (crawled from Amazon.com and Cnet.com) address five different consumer electronics products[8] (two digital cameras, a DVD player, a mp3 player, and a cellular phone). Granularity of annotation is at the sentence level. Each sentence is annotated with regard to the contained "features" (product aspects) and the corresponding contextual sentiment polarity. In their terminology, features refer to the fine-grained aspects of a concrete product. In other words, relevant feature sets also differ between products of the same domain — for example, the two digital cameras exhibit different sets of features. In average, they annotate 69 features for a product. Listing 5.1 shows several exemplary sentence level annotations. Implicit mentions of features are marked with the *[u]* attribute (e.g., as in the sixth sentence).

Listing 5.1: Examples of sentence level annotations within the Hu & Liu Corpus.

```
canon powershot g3[+3]##i recently purchased the canon powershot g3 and am extremely satisfied with the
    purchase .
product[+3]##it is a very amazing product .
picture[+2], auto mode[+2]##but at the same time , it takes wonderful pictures very easily in " auto "
    mode , so that even an average joe like me can use it !
photo quality[+2], auto mode[+2]##i began taking pics as soon as i got this camera and am amazed at the
    quality of photos i have took simply by using the auto mode .
photo[+3]##i was able to take great photos of the 4th of july fire works , and got some amazing shots of
    the kids playing with sparklers .
picture[+2],feel[+1][u]##it takes great pictures , operates quickly , and feels solid .
four megapixel[+1]##four megapixels is great .
```

[7] http://ir.dcs.gla.ac.uk/test_collections/access_to_data.html

[8] An extension of the original corpus with nine products is available and has been used in Ding et al. [102]

As can be seen from the excerpt, the style of annotation is rather shallow. Although focused on fine-grained aspects, the specific mentions (i.e., their text spans) are not annotated in the text. Also, the contextual polarity information is not resolved to sentiment expressions and sentiment shifters. The scheme can be compared most closely to our sentence level annotation scheme, but differs in the following important points: With regard to product aspects, we annotate sentences on the concept level instead of the mention level (e.g., they distinguish "picture", "photo", "picture quality", and "photo quality", which we subsume to the single concept "picture quality"). We also provide information about the discourse function of each sentence, which is not available in their corpus. Further, we annotate additional information about polar facts and irrealis contexts. It has also been noted that the Hu/Liu corpus partly lacks a consistent annotation. Feiguina and Lapalme [127] present a list of these issues.

Ganu et al. Corpus of Restaurant Reviews

Ganu et al. [138] present a corpus of restaurant reviews that is annotated on the sentence level. The corpus consists of approximately 3,400 sentences. Each sentence is categorized into one or more of six predefined topics ("food", "service", "price", "ambience", "anecdotes", and "miscellaneous") and is manually labeled with its sentiment polarity status ("positive", "negative", "neutral", or "conflict"). A subset of 450 sentences is used to measure inter annotator agreement among three human annotators. Relatively high *kappa coefficients*[9] of above 0.8 are reported for three of the topic categories and for the positive sentiment class. Except for the "anecdotes" topic, all other classes exhibit good agreement with kappa values of above 0.6. Whereas being comparable in size and annotation scheme to our sentence level corpora, the granularity with regard to the predefined set of topics is much lower. In effect only four domain specific topics are annotated, the topics "anecdotes" and "miscellaneous" can be subsumed as a class similar to our (implicit) off-topic category. Our datasets distinguish 22 (hotel) and 31 (topics).

Zhuang et al. Corpus of Movie Reviews

Zhuang et al. [468] developed a hand-annotated corpus of movie reviews extracted from the Internet Movie Database (IMDB). The dataset comprises roughly 16,000 sentences and 260,000 tokens in 1,100 reviews. Opinion is annotated with respect to the expression level, but the unit of annotation (i.e., the text span or frame) is a sentence. Similar to the Hu/Liu corpus, expression level information is provided as attributes of a sentence annotation. With regard to sentiment targets they distinguish "feature words" and "feature types". Feature words resemble the mention level of aspects, whereas feature types correspond to concept level aspects (topics). The list of predefined feature types comprises 20 categories. For sentiment expressions the contextual polarity is captured (positive or negative), but no information about intensity or potential shifters (e.g., negation) is given. Only sentences containing opinion word / feature word pairs are annotated. With regard to feature types the annotation scheme is comparable to our sentence level dataset (we annotate topics), but we capture topic and sentiment independent of each other and do not restrict the annotation to topic/opinion pairs. As with the Hu & Liu dataset, no information about discourse functions, polar facts, or irrealis contexts is available. With respect to the expression level, the annotations are less complex than ours. First, sentiment targets and expressions are not identified as annotation frames (they are string-valued attributes of a sentence annotation) and second, important constituents, such as negation or amplification, are not modeled.

[9] Cohen's [84] or Fleiss' kappa [131] are statistical measures commonly used to assess the agreement among annotators.

Toprak et al. Corpus of Services Reviews

Similar to us, Toprak et al. [386] present a corpus that is annotated on the sentence and on the expression level. The corpus is composed of customer reviews (extracted from Rateitall.com and Epinions.com) that target two different domains: online universities and online services. The dataset covers 240 university reviews (2786 sentences) and 234 service reviews (6091 sentences). As we noted in Section 4.4.2, their sentence level annotation scheme has three main attributes describing the sentence's relevancy with regard to the overall topic of the review, whether it contains an explicit opinion, or represents a polar fact (in that case the polar fact polarity is given). They consider the sentence level annotations as a precursor to expression level annotations and argue that it increases the reliability of determining correct text spans. Expression level annotations include sentiment target, expression, shifter, and holder identification. Sentiment expressions are attributed with a polarity (positive, negative, and neutral) and an intensity value (weak, average, and strong). With regard to sentiment shifters, they capture amplifiers, downtoners, and negation. Also anaphoric relations are annotated by means of a reference marker. Similar to us they do not annotate complete co-reference chains. The basic constituents covered by their two annotation schemes tightly overlap with our schemes, since both models are derived from the functional components underlying the MPQA annotation scheme. Despite the fact that the corpus addresses relatively exotic domains[10], we primarily miss topic and discourse function annotations for our evaluation and exploration purposes.

2010 JDPA Sentiment Corpus

The most recent and most complex corpus in the context of aspect-oriented sentiment analysis is the *2010 JDPA Sentiment Corpus for the Automotive Domain* by Kessler et al. [209]. It contains 335 blog posts on automobiles, spanning 13,126 sentences and 223,001 tokens. The annotation scheme operates on the expression level and the basic constituents again represent the functional components of the MPQA model. The central difference to our and the other annotation schemes is the way they handle sentiment targets. Sentiment targets are annotated as mentions of entities, which they define as "discourse representations of concrete objects (e.g., car, door) and non-concrete objects (e.g., handling, power)" [209]. Despite domain specific entities, they also capture a subset of the entity types defined by the ACE program [87] (e.g., "Person", "Organization", "Location", or "Time"). Named, nominal and pronominal mentions are linked to the associated entities via co-reference chains. Additionally, relations between mentions such as *part-of*, *instance-of*, or *feature-of* are marked within the discourse. Sentiment expressions are related to concrete mentions and are attributed with prior polarity information; however, prior intensity is not provided. For entities that have been judged to be important, they separately provide an entity level sentiment polarity value that summarizes the sentiment towards this entity and its parts. Their set of sentiment shifters follows the ideas presented by Polanyi and Zaenen [303]. Similar to us, they cover negators, intensifiers (amplification and downtoning), committers (solidifier, softener) and neutralizers. Implied mentions of product aspects, as well as implicit sentiment expressions (polar facts) are not covered by their annotation scheme. They also do not differentiate discourse functions and neither provide information on the concept level.

5.5. Summary

In this chapter we described the instantiation of the discourse oriented and expression level model as annotation schemes. We pointed out that **the discourse oriented model is implemented at the sentence level** and motivated this decision by an easier annotation process (without loss of expressiveness). In comparison to the original models, we extended both annotation schemes to cover additional attributes that are mostly of exploratory nature and of value for a more detailed evaluation.

[10] Toprak et al. note that this is due to specific project requirements.

To increase the significance and reliability of evaluation results, we chose to compile corpora for two different domains, namely hotel and digital camera reviews.

We further discussed the nature of the corpora, highlighted the acquisition process, and presented the datasets' basic characteristics. Having two distinct annotation schemes and two distinct domains, **we created four manually labeled corpora**. As annotating on the expression levels incurs a much higher effort, the related corpora only consist of a subset of the sentence level corpora. Sections 5.2 and 5.3 covered detailed descriptions of both annotation schemes. With regard to the sentence level annotation scheme we presented the predefined sets of discourse functions and topics and pointed out how the sets were derived.

We motivated the basic decision to develop our own evaluation corpora from scratch by the **lack of adequate, widely accepted corpora that fit our purposes**. Section 5.4 looked more closely on other available datasets. By evaluating their characteristics we could further stress the need for compiling our own evaluation corpora.

In theory, there is no difference between theory and practice. But, in practice, there is.

Jan L. A. van de Snepscheut

6. Corpus Analysis

The primary goal of our analysis is to gain insight into the nature of the different phenomena that we discussed in the previous chapters. Apart from quantifying these aspects and providing descriptive statistics of the corpora, our analysis is guided by the question of how to operationalize our models in a real system. In particular, we aim at answering the following three questions:

- What is the relevance of the different aspects of our models with regard to aspect-based sentiment analysis systems? For example, which share of sentiment expressions exhibit a target-dependent polarity and in consequence, does a system need to cope with this phenomenon?

- Are there any significant differences with regard to the domains of service (hotel) and product (camera) reviews and if so, of which nature are they and how do they effect the design of a customer review mining system?

- Which tasks and problems in customer review mining can be addressed by using the presented corpora?

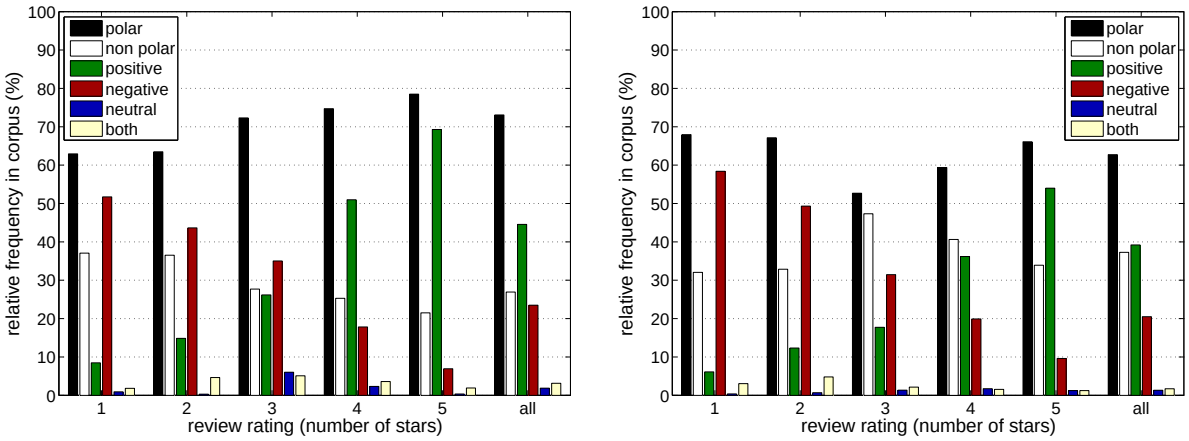
The remainder of this chapter is organized as follows: In Section 6.1, we analyze the corpora with respect to the sentence level annotations and Section 6.2 covers the expression level annotations. In each of the two sections, we analyze the different functional components of the particular annotation scheme and discuss relevant corpus statistics with respect to the earlier mentioned questions. The chapter is intended to provide the basic insights that can be drawn from an exploratory analysis of the corpora. We leave more detailed and problem specific analyses for discussions in the context of mistake analysis of particular approaches.

6.1. Sentence Level Corpus

6.1.1. Sentiment and Polar Facts

Within the hotel review corpus, 2540 of the 3476 sentences (73.1%) are polar, that is, they either contain an explicit expression of sentiment or cover a polar fact expression. In the digital camera corpus, the relative frequency of polar sentences is less, 2191 out of 3493 sentences (62.7%) contain polar language. Both numbers are quite high and underline the assumption that customer reviews exhibit a high proportion of subjective information. We can regard these numbers as lower bounds for the precision of a sentence level subjectivity classifier. A baseline method that simply classifies each sentence as subjective, would achieve precision values of 73% and 63%, respectively. We can further conclude that a topic classification alone (i.e., without sentiment detection) is a valuable tool for customer review mining: If more than 60% of sentences express sentiment, a system that only categorizes by topic, is already very helpful with structuring the review information. Figure 6.1 presents the detailed distribution[1] of polar sentences with respect to the overall rating of a review (ratings range from 1 $\sim$ worst to 5 $\sim$ best).

[1] Table B.1 contains the data that was used to create Fig. 6.1.



(a) hotel corpus

(b) digital camera corpus

Figure 6.1.: The distribution of polar sentences in the sentence level corpora.

Correlation between User-Provided Rating and Sentence Polarity

The data shows, not very surprisingly, a strong correlation between the overall rating of a review and the polarity of its contained sentences. The probability that a sentence in a review rated with 1 has a negative polarity is 0.52 (hotel) and 0.58 (camera), compared to a probability of 0.08 (hotel) and 0.06 (digital camera) for positive polarity. In contrast, for reviews with an overall rating of 5, the probability of a sentence to have positive polarity is 0.69 (hotel) and 0.54 (digital camera), whereas negative polarity is unlikely with probabilities of 0.07 and 0.10. In both corpora, the relative frequency of negative sentences prevails in reviews with ratings in the range of 1 to 3, whereas positive sentences are more likely in reviews with ratings 4 and 5. Furthermore, we find that the relative frequency of neutral sentences is higher in reviews with a mediocre overall rating of 3 (and 4 in the digital camera dataset). We can conclude that the overall rating of a review is a good indicator for the polarity of its contained sentences and appears to be a valuable feature with regard to a sentence level polarity classification task.

A natural assumption would be that polar sentences more likely occur in extreme reviews (i.e., with overall ratings 1 or 5). Whereas the assumption seems to be true for the camera review dataset, we find contradictory data for the hotel review corpus. For hotel reviews, we can observe that the relative frequency of polar sentences increases with the overall rating of a review. The share of polar sentences in reviews with an overall rating of 5 is much higher (78.5%) than in reviews with a rating of 1 (62.9%).

Imbalanced Data

Also at the sentence level, the distribution of polarity is skewed towards positive ratings. However, compared to the document level, the skew is less pronounced. The share of positive sentences of all polar sentences is 70.0% (hotel) and 62.5% (digital camera), compared to 32.2% and 32.7% for negative sentences. Thus, in both corpora, the ratio of positive polar versus negative polar sentences is around 2:1, whereas on the document level the ratios are around 3:1 and 8:1 (cf., Section 5.1). This lower "between class imbalance" reduces the effects that come with learning from imbalanced data [72, 162, 188], at least for the distinction between positive and negative sentences.

Neutral and Mixed Sentiment

Sentences with neutral sentiment or mixed sentiment occur very rarely. Reviewers expresses neutral sentiment in only 1.9% (hotel) and 1.3% (digital camera) of all sentences. It is apparent, in particular within the hotel review corpus, that neutral sentiment is more often expressed in overall neutral reviews (rating 3). Sentences with mixed polarity have a share of only 3.1% in the hotel corpus and 1.7% in the digital camera corpus. We can conclude that capturing both phenomena as part of a sentence level polarity classifier is less relevant. The majority of polar sentences is either positive or negative.

Polar Facts

Tables 6.1a and 6.1b depict the distribution of polar facts in both corpora. The phenomenon is significant in both corpora. In the hotel corpus, polarity is expressed by means of polar facts in 16.7% of all cases. In the digital camera corpus the number is 13.9%. It is noteworthy that polar facts are used far more often to express negative sentiment. We can observe the following: Reviewers may induce negative sentiment (as polar facts) by naming both, the occurrence of undesired or the lack of desired behavior. On the other hand, they may express positive sentiment by describing the occurrence of desired behavior. But it is very rarely that they evoke positive sentiment by naming the absence of undesired behavior. This is one reason, why reviewers tend to describe the facts that lead to negative sentiment more often than the facts that evoke positive sentiment. In consequence, we conclude that the phenomenon of polar facts is significant, in particular with regard to the expression of negative sentiment (> 30% of negative, polar sentences). It thus needs to be examined when building sentence level polarity classifiers. Simple lexicon-based classifiers are likely to have difficulties in detecting polar facts.

polarity	polar fact	¬polar fact
positive	134 (8.65%)	1415 (91.35%)
negative	285 (34.88%)	532 (65.12%)
neutral	1 (1.54%)	64 (98.46%)
both	5 (4.59%)	104 (95.41%)
all	425 (16.73%)	2115 (83.27%)

(a) hotel corpus

polarity	polar fact	¬polar fact
positive	87 (6.36%)	1282 (93.64%)
negative	213 (29.75%)	503 (70.25%)
neutral	1 (2.17%)	45 (97.83%)
both	4 (6.67%)	56 (93.33%)
all	305 (13.92%)	1886 (86.08%)

(b) digital camera corpus

Table 6.1.: The distribution of polar facts in the review corpora.

6.1.2. Topics

Within the hotel review corpus 2712 of 3476 sentences (78.0%) are on-topic. In the digital camera corpus we count 2573 of 3493 sentences (73.7%) which are on-topic. These numbers confirm our assumption that customer reviews are generally very focused documents and most of the provided information is relevant to the discussed product or one of its aspects. Table 6.2 lists the distribution[2] of topics for both corpora. In the hotel review domain, reviewers most often comment on the topics "room", "service", and "location". These three topics alone account for roughly 50% of all sentences in the corpus. Within the digital camera domain, the top five topics "picture quality", "ease of use",

[2] Take note that, since a sentence may be attributed to multiple topics, the sum of the frequency counts is greater than the number of sentences in the corpus. In consequence, summing up the relative frequencies of all topics plus the off-topic category, reveals a number greater than 100%.

topic	frequency (share)
room	700 (20.14%)
service	593 (17.06%)
location	475 (13.67%)
PRODUCT	413 (11.88%)
dining	281 (8.08%)
sleep quality	265 (7.62%)
facility	164 (4.72%)
cleanliness	163 (4.69%)
price	160 (4.60%)
internet	65 (1.87%)
decoration	60 (1.73%)
ON-TOPIC	2712 (78.02%)
OFF-TOPIC	764 (21.98%)

(a) hotel corpus

topic	frequency (share)
picture quality	489 (14.00%)
PRODUCT	473 (13.50%)
ease of use	334 (9.60%)
appearance	281 (8.00%)
battery	232 (6.60%)
optics	231 (6.60%)
settings	185 (5.30%)
price	161 (4.60%)
features	154 (4.40%)
video recording	128 (3.70%)
memory	95 (2.70%)
screen	90 (2.60%)
speed	87 (2.50%)
flash	70 (2.00%)
connectivity	68 (1.90%)
software	56 (1.60%)
ON-TOPIC	2573 (73.70%)
OFF-TOPIC	920 (26.30%)

(b) digital camera corpus

Table 6.2.: The distribution of topics in the review corpora.

"appearance", "battery", and "optics" account for around 45% of all sentences. In both corpora, the product itself (category *PRODUCT*) is addressed in around 12-14% of sentences.

Table 6.3 shows the correlation between polar and on-topic sentences in a combined hotel and camera review corpus. The contingency table shows a grand total of 6969 sentences in the combined corpus. A subset of 4329 sentences is both, polar and on-topic. In other words, around 62% of all sentences are relevant with regard to extracting on-topic, polar sentences. The other 40% are either off-topic, non-polar, or both. We can further observe a strong correlation between on-topic and sentiment bearing sentences. Computing the *phi coefficient*[3] reveals a value of $r_\phi = 0.53$, indicating a relatively strong association. Of the marginal total of 4731 polar sentences, 4329 (91.5%) are also on-topic — that is, we have a conditional probability $Pr\,(\text{sentence is on-topic} \mid \text{sentence is polar}) = 0.915$. On the other hand, 81.9% of all on-topic sentences are also polar (4329 of 5285 sentences). In consequence, we can conclude that the property of being polar, is a good indicator for identifying relevant (i.e., polar & on-topic) sentences. With an ideal polarity classifier (and without any specific topic classifier), we could extract relevant sentences with an already high accuracy of 91.5%.

	polar	**¬polar**	
on-topic	4329	956	5285
off-topic	402	1282	1684
	4731	2238	6969

Table 6.3.: A 2 x 2 contingency table showing the correlation between the topic dimension and the polarity dimension. The presented numbers are based on a conflation of the hotel and digital camera corpora.

[3] The *phi coefficient* r_ϕ is an index of association for two binary variables [97]. Similar to the *Pearson product-moment correlation coefficient*, its value ranges from -1 (perfect disagreement) to +1 (perfect agreement) and a value of zero indicates no association. Correlation coefficients greater than 0.5 are generally regarded as indicating a strong relationship.

The sentence level annotation scheme allows to attribute multiple topics to a single sentence. In both corpora, the relative frequency of on-topic sentences with multiple topics is around 18%. In the hotel corpus, we have 485 of 2712 (17.9%) on-topic sentences that are associated with multiple topics. In the digital camera dataset we count 458 of 2573 (17.8%) sentences. The numbers indicate that reviewers tend to address only a single topic within one sentence. Nevertheless, a relative frequency of 18% shows that the phenomenon cannot be ignored and needs to be tackled when classifying sentences with regard to topics. We conclude that the task of sentence level topic classification is indeed a multi-label classification problem.

6.1.3. Discourse Functions

The sentence's discourse function attribute specifies the function or role a sentence has in the discourse structure of a customer review. As part of the sentence level annotation scheme we defined $16 + 1$ different functions, which we identified to be relevant in the domain of customer reviews. Figure 6.2 presents the distribution[4] of these functions in both corpora.

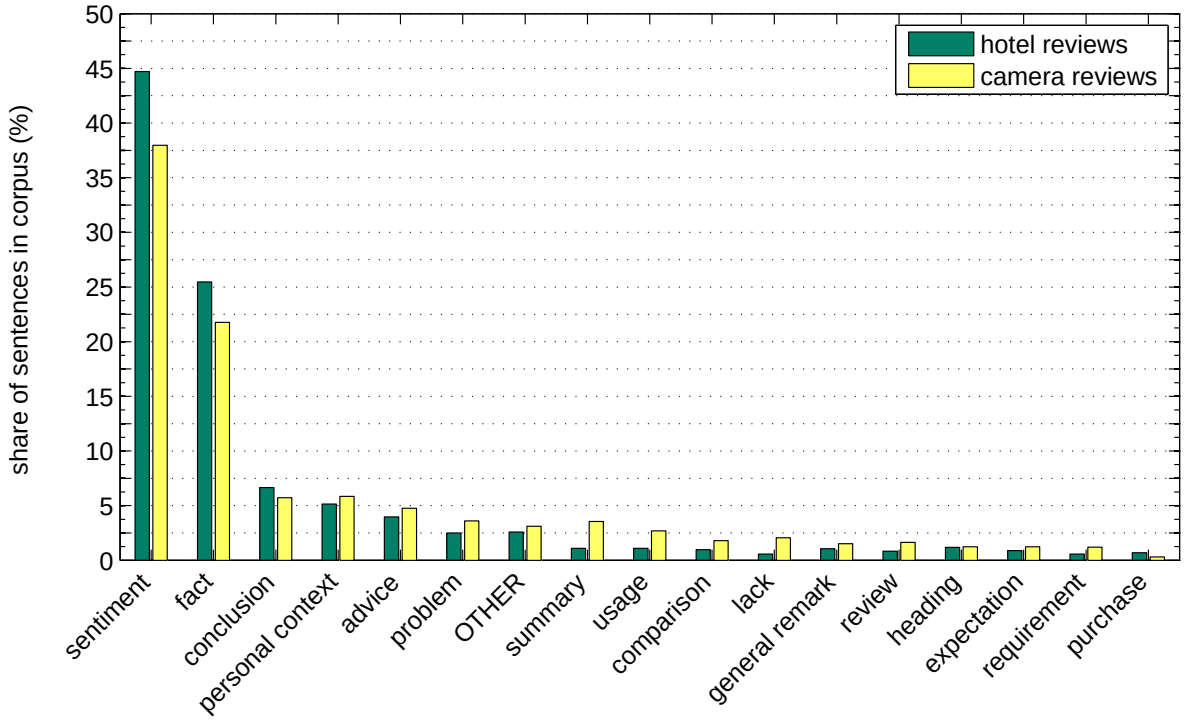


Figure 6.2.: The distribution of discourse functions in the review corpora.

First, we can observe that the coverage of the predefined discourse functions is quite good. A share of around 97% of all annotated sentences can be associated with one of the functions, only about 3% are associated with the "OTHER" category (examples for sentences that we annotated with the discourse function "OTHER" are shown in Appendix A.2.7). Not surprisingly, the major share of sentences (hotel: 44.7%, camera: 38.0%) is attributed to the discourse function "sentiment". Sentences with factual information (category "fact") that cannot be attributed to one of the other, more specific discourse functions, account for 25.5% (hotel) and 21.8% (camera) of all sentences. In consequence, the remaining functions (excluding the category "OTHER") make up for shares of 27.3% (hotel) and 36.7% (camera).

[4] Table B.3 contains the data that was used to create Fig. 6.1.

Relevance of the Discourse Function Dimension

In Section 4.2 we motivated the consideration of discourse functions by pointing out an information need with regard to this dimension. Relevant functions with respect to this information need are especially the types "conclusion", "personal context", "advice", "problem", and "lack". For example, conclusive passages are informative as they summarize the overall impression of a reviewer, and the "personal context" may provide information about the trustworthiness of the reviewer. In the hotel corpus, the discourse function "conclusion" has the maximum relative frequency (6.7%) among these functions. In the digital camera corpus it is the function "personal context", with a relative frequency of 5.8%. Considering the combined hotel and camera review corpus, we find that 51.3% of all reviews contain at least one conclusive sentence and in 39.3%, reviewers provide personal context. Sentences that contain advices account for 4.4%, problem descriptions for 2.9%, and passages that describe a lack (of something desired) for 1.3% of all sentences. From the reported numbers we conclude that examining the dimension of discourse functions is helpful in analyzing customer reviews. The aforementioned five most relevant discourse functions are associated in total with one fifth (20.2%) of all sentences.

6.1.4. Further Observations

The corpus analysis shows that only a relative small number (1.9%) of sentences contain a comparison. This number is relevant when considering whether an expression level model should cope with comparative sentences or not. Our current expression level annotation scheme does not capture comparisons. The model lacks the possibility to express "sentiment relations" between two sentiment targets. However, the low number of actual comparisons shows that reducing the complexity with regard to this phenomenon is acceptable.

The sentence level annotation scheme further contains the exploratory attributes "non focus entity" and "irrealis". The first attribute marks sentences that address an entity that is not the primary entity in focus of the review (most commonly as part of a comparison). For the domain of customer reviews, we find that the relative frequency of such sentences is very low, with below one percent (68 sentences, resulting in a share of 0.98%). Also this observation confirms our hypothesis that reviews are highly focused documents. We conclude that, at least in the examined sentence level corpora, it is of minor interest to apply named entity recognition. The share of polar sentences that were marked with the "irrealis" flag is 2.4%, showing that a detection of this phenomenon is also not necessarily needed to significantly improve the accuracy of a sentence level polarity classifier.

6.2. Expression Level Corpus

6.2.1. Aspect Mentions and Sentiment Targets

Distribution of Mention Types

Recall that we distinguish between *nominal*, *named*, *pronominal*, and *implicit* mentions of product aspects. We defined a nominal mention as a noun phrase that explicitly refers to the product or one of its aspects (e.g., "camera" or "image stabilization"). Named mentions refer to the product or an aspect by means of proper names (e.g., "Canon EOS 550D" or "SteadyShot"). The expression level annotation scheme distinguishes both types for each utterance of a mention, independent of whether it is targeted by a sentiment expression or not (pronominal and implicit mentions are only captured when addressed as sentiment targets). Table 6.4 shows the distribution of nominal versus named mentions. We read the information as follows: Within the hotel review corpus, we count 2,158 (2,066 + 92) nominal or named mentions. The majority (95.7%) is of nominal nature and only a small share is of type

"named". The table also shows that 48.9% of all nominal mentions are associated with at least one sentiment expression, whereas only 25 named mentions (27.2% of all named mentions) are marked as sentiment target. The digital camera review corpus contains in total 2,149 (1,918 + 231) nominal and named mentions. It is apparent that the relative frequency of named mentions is higher with 10.8% (although still low). The relative frequency of nominal mentions marked as sentiment target, is similar to the hotel review corpus (45.2%). With regard to named mentions, it is much less (only 13.9%).

	hotel		camera	
type	**all mentions**	**sentiment targets**	**all mentions**	**sentiment targets**
nominal	2066 (95.74%)	1011 (48.94%)	1918 (89.25%)	867 (45.20%)
named	92 (4.26%)	25 (27.17%)	231 (10.75%)	32 (13.85%)

Table 6.4.: The distribution of nominal and named mentions in the review corpora.

In Table 6.5, we examine the distribution of mention types with respect to sentiment targets. In the hotel review corpus, a total of 1,203 phrases are annotated as sentiment target. The major share (84.0%) of sentiment targets is expressed explicitly as a nominal mention. Only an insignificant share of 2.1% refers to named mentions. Pronominal mentions account for 6.4% of targets and implicit mentions for 7.5%. Considering the digital camera corpus, we observe that the relative frequency of pronominal and implicit mentions is higher, with 10.4% and 8.8%, respectively. Nominal mentions amount to roughly 78% in this corpus. Examining the numbers for the combined corpus, we can conclude that detecting nominal mentions is most important. Pronominal and implicit mentions account for about 16%. In other words, perfect identification of these two mention types has the potential to increase the recall level for sentiment target extraction by 16 percentage points. Considering them is thus generally desirable. Detecting named entities is of minor importance — only 2.4% of sentiment targets are expressed in this manner.

	hotel	**camera**	**combined**
nominal	1011 (84.04%)	867 (77.90%)	1878 (81.09%)
named	25 (2.08%)	32 (2.88%)	57 (2.46%)
pronominal	77 (6.40%)	116 (10.42%)	193 (8.33%)
implicit	90 (7.48%)	98 (8.81%)	188 (8.12%)
	1203	1113	2316

Table 6.5.: The distribution of sentiment target mention types.

In both corpora, we observe a similar distribution of sentiment targets. The average number of sentiment targets per document is slightly less in the digital camera corpus (7.4 compared to 8.0 occurrences). We find that three out of the 150 annotated digital camera reviews do not contain a sentiment target at all; in the hotel review corpus all reviews have at least one sentiment target. The maximum number of targets is counted in the digital camera corpus with 40 occurrences. Comparing these numbers to the basic dataset statistics presented in Table 5.1, we find that the differences correlate with the different average lengths of review documents. The smaller average length of digital camera reviews correlates to the lower average number of occurrences (the same is true for the higher variability).

In both corpora the proportion of sentences containing at least one sentiment target is more than 50%. In the hotel corpus 53.0% contain a target, whereas in the digital camera corpus the proportion

statistic	hotel	camera
avg. targets per document	8.02	7.42
std. targets per document	4.68	7.46
min. targets per document	1	0
max. targets per document	25	40
avg. targets per sentence	0.72	0.77
proportion of sentences with target	52.98%	55.61%

Table 6.6.: Statistics about the occurrences of sentiment targets in the review corpora.

is 55.6%. Both proportions are less than the 60% of on-topic, polar sentences we annotated on the sentence level. Although we cannot directly compare these numbers, as the expression level corpora only represent a subset of the sentence level corpora, the difference shows that implicit aspect mentions (as discussed earlier) and polar facts (which are not captured by the expression level scheme) represent a significant share. We conclude that it is desirable to analyze customer reviews on multiple levels of granularity. Phenomenons such as implicit mentions and polar facts are better addressed at the sentence level.

Nominal Aspect Mentions and Sentiment Targets

As nominal mentions of product aspects represent the greatest share of mention types, we examine these occurrences in more detail. Table 6.7 presents the basic statistics with regard to nominal aspect mentions. Comparing the numbers of the hotel review and digital camera review corpora, we find that the complexity of nominal mentions and targets is very similar. The average length of a nominal mention is 1.28 tokens and 1.40 tokens, respectively. The low standard deviation indicates that the majority of mentions is indeed composed of a single token (78.3% in the hotel corpus versus 71.6% in the digital camera corpus). The maximum length is seven tokens for the hotel dataset (e.g., "room in the back of the hotel") and nine tokes for the camera corpus. We count 490 distinct mentions[5] in the hotel corpus and 477 in the digital camera corpus.

statistic	hotel	camera
avg. tokens per mention	1.28	1.40
max. tokens per mention	7	9
std. tokens per mention	0.60	0.76
distinct mentions	490	477
root ttr of mentions	10.78	10.89

Table 6.7.: Statistics about the occurrences of the nominal mention type in the review corpora.

To further compare the complexity of the corpora, we compute a measure related to the *root type-token ratio* (root-ttr), which was proposed to examine the *lexical diversity* of a corpus. The measure is based on the simple *type-token ratio* [195], but uses the square root of the denominator to relieve the influence of the sample size. It is computed as $\textit{root-ttr} = \frac{\#\text{distinct tokens}}{\sqrt{\#\text{tokens}}}$. In our case, we consider mentions (instead of tokens) to compute the *root-ttr*. The sample size is the total number of (nominal) mentions. A lower score for the measure corresponds to a lower lexical variability. We can observe that the lexical diversity of nominal mentions is very similar in both corpora, being slightly lower in the hotel review corpus (10.78 compared to 10.89).

[5]Recall that mentions refer to normalized (i.e., lemmatized) occurrences of aspects.

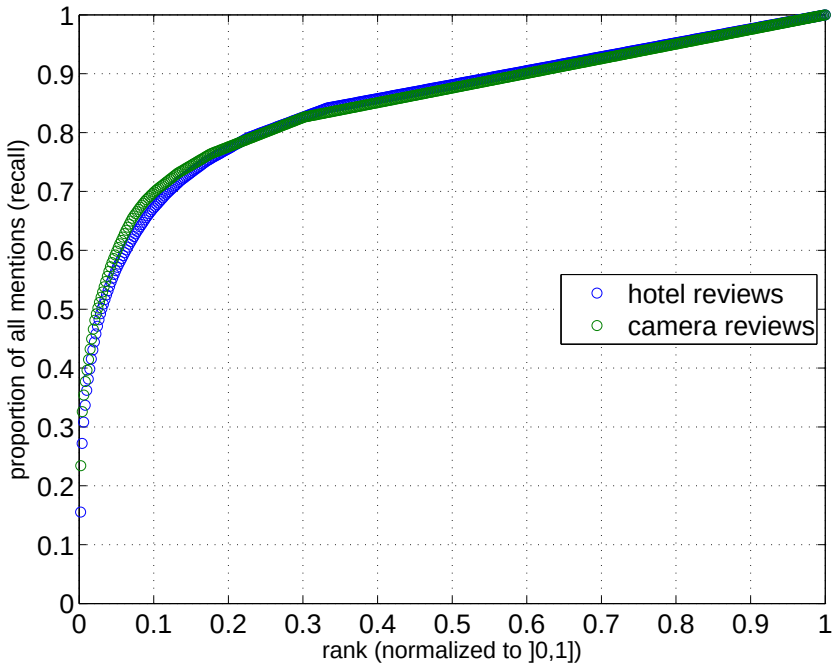


Figure 6.3.: The lexical diversity of nominal aspect mentions. The normalized rank is plotted against the recall.

Whereas the *root-ttr* is a single score, suited to compare the lexical diversity of two samples, the degree of lexical diversity can also be explained by the information given in Fig. 6.3[6]. We rank each distinct mention according to its frequency of occurrence (giving a higher rank to more frequent mentions). The horizontal axis of Fig. 6.3 displays the rank, normalized to the interval $]0;1]$. The figure then reads for example as follows: With 10% of the most frequent mentions (normalized rank 0.1) around 67% (70%) of all occurrences of mentions in the corpus are covered. The plot mirrors that the frequencies of occurrence basically follow a *Zipfian distribution* — that is, they are inversely proportional to their rank (in this plot expressed as a flattening of the curve with higher ranks). Again, we can observe that both corpora exhibit very similar numbers. The slightly steeper curve for the digital camera corpus expresses that the most frequent mentions exhibit a slightly higher proportion. The numbers are especially relevant when considering a lexicon-based approach to extract nominal aspect mentions. For example, with regard to the examined corpora, to achieve a recall of 80%, a lexicon of product aspects must contain around 25% of the total number of aspects occurring in the documents. We will examine the distribution more closely when we analyze the different approaches to product aspect and sentiment target identification.

Table 6.8 examines the distribution of part-of-speech patterns for nominal mentions of product aspects. It shows the relevant numbers for the ten most frequent combinations (along with an exemplary aspect mention that represents the pattern). Not surprisingly, noun phrases account for the major share of combinations. The five most frequent patterns refer to nouns (potentially pre-modified by an adjective) and exhibit a cumulated share of 92.8%. Verbs (VB) and adjectives (JJ) typically do not constitute a valid nominal mention, but occur in the statistic. Looking at the actual data reveals that all occurrences can be ascribed to mistakes by the automatic part-of-speech tagger. In conclusion, the numbers show that with relative few part-of-speech patterns, a great share of nominal mentions can be addressed.

[6] Also see Table B.2, which presents the same statistic, but setting focus on the recall levels from 30% to 100%.

rank	part-of-speech tag	example	frequency	share	cumulated share
1	NN	camera	2928	73.49%	73.49%
2	NN NN	picture quality	510	12.80%	86.30%
3	JJ NN	front desk	180	4.52%	90.81%
4	NN NN NN	burst shooting mode	50	1.26%	92.07%
5	JJ NN NN	front desk staff	36	0.90%	92.97%
6	NN IN DT NN	size of the camera	29	0.73%	93.70%
7	VB NN	build quality	27	0.68%	94.38%
8	VB	shower	26	0.65%	95.03%
9	NN IN NN	ease of use	21	0.53%	95.56%
10	JJ	safe	20	0.50%	96.06%

Table 6.8.: The distribution of the ten most frequent part-of-speech tags of nominal aspect mentions. The numbers are based on a conflation of the hotel and camera review corpora.

6.2.2. Sentiment Expressions

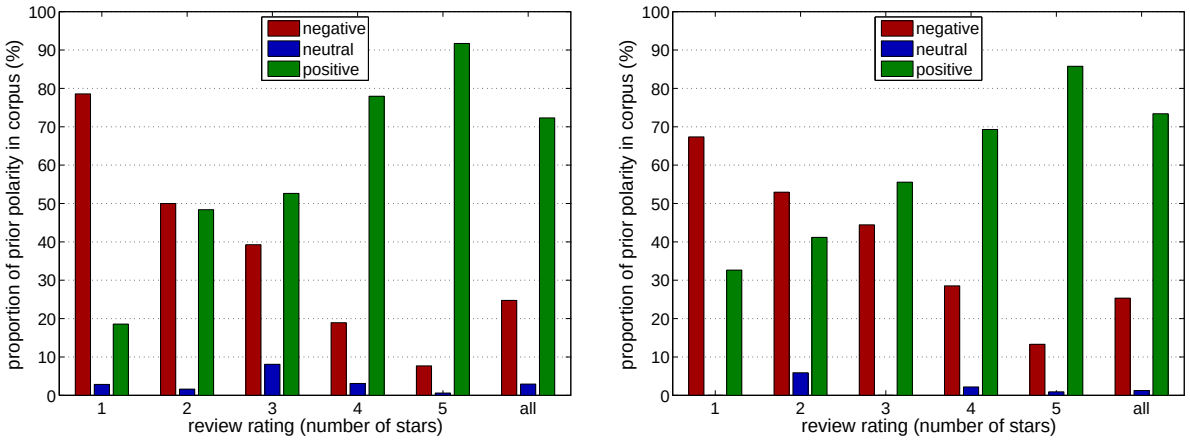
Table 6.9 summarizes the statistics with regard to sentiment expressions. In total, the hotel review corpus contains nearly 1,400 sentiment expressions. In the digital camera corpus we annotated 1,196 expressions. As with sentiment targets, the average number of sentiment expressions per document is higher in the hotel review dataset. Again, this observation can be ascribed to the greater average length of hotel reviews. Since we defined sentiment targets and sentiment expressions as existentially dependent on each other, the proportion of sentences containing explicit sentiment expressions/targets is the same[7]. The average number of tokens per sentiment expression is low with 1.23 and 1.32 tokens. In the hotel corpus, 86.7% of sentiment expression are composed of a single token, in the digital camera dataset the proportion is 81.0%. Regarding the proportion of sentiment expressions with strong intensity, both corpora exhibit a similar distribution. In the hotel review corpus around 12.5% of sentiment expressions have strong intensity compared to 16.0% in the digital camera dataset.

statistic	hotel	camera
sentiment expressions	1402	1196
avg. sentiment expressions per document	9.35	7.97
std. sentiment expressions per document	5.53	8.03
min. sentiment expressions per document	1	0
max. sentiment expressions per document	29	45
sentences with sentiment expression	53.04%	55.68%
avg. tokens per sentiment expression	1.23	1.32
unique sentiment expressions	488	473
root ttr	13.03	13.68
proportion of expressions with multiple targets	3.50%	3.68%
proportion of expressions with shifters	29.53%	33.28%
proportion of target specific expressions	20.54%	19.23%
proportion of expressions with strong intensity	12.48%	15.97%

Table 6.9.: Basic statistics about the occurrences of sentiment expressions in the review corpora.

Overall, we find 488 and 473 different types of sentiment expressions, resulting in *root-ttr* scores of 13.0 and 13.7. The numbers show that the lexical diversity of sentiment expressions is higher than

[7]The minimal difference is due to pronominal targets, which need not occur in the same sentence as the sentiment expression.



(a) hotel corpus

(b) camera corpus

Figure 6.4.: The correlation between the user-provided review ratings and the occurrence of polar expressions.

the variability of nominal aspect mentions. This indicates that sentiment expression may be more difficult to identify than nominal aspect mentions.

We further find that sentiment expressions that target multiple sentiment targets, represent only a small share of all utterances of sentiment (3.5% and 3.7%). Reducing the complexity of a sentiment analysis systems with regard to this phenomenon can thus be regarded as acceptable. When analyzing the number of sentiment expression that are affected by at least one sentiment shifter, we can observe that shifters play a major role. Nearly a third of all sentiment expressions in both corpora (29.5% and 33.3%) is associated with one or more shifters. We examine sentiment shifters more closely in the next section.

Another important number is the proportion of target-specific sentiment expressions. Recall that these are expressions that, depending on the context (i.e., the sentiment target), may shift polarity or may be polysemous with respect to subjectivity/objectivity. Such expressions account for a relatively large share of around 20% in both corpora. We thus conclude that recognizing target-specific sentiment expressions is an important factor in aspect oriented sentiment analysis. We further find that, in both datasets, adjectives account for more than 90% of all target-specific expressions (hotel: 93.8%, camera: 91.3%).

Figure 6.4 examines the distribution of prior polarities (for the illustration we only count sentiment expressions which are not shifted by a neutralizer or negator). The distribution is highly skewed towards positive prior polarity. In both corpora, positive polarity sums up to around 73%, whereas negative polarity has a share of around one quarter (25.4% and 26.5%). Neutral polarity accounts for only 2.9% and 1.1%, respectively. Not surprisingly, we find strong correlations between the overall rating of a review and positive/negative polarities. For example, more than 90% of sentiment expression in "five star" hotel reviews have positive polarity. On the other hand, more surprisingly, we observe that sentiment expressions with positive prior polarity also occur with a relative high proportion in overall negative reviews. Positive sentiment expressions in "one-star" camera reviews account for nearly a third and in two-star reviews the share of positive expressions nearly equals the share of negative expressions (in both corpora). Also the amount of expressions with negative prior polarity in overall positive reviews (four and five stars) is significant. This observation underlines the need for fine-grained sentiment analysis — for example, on the sentence or expression level.

We also compare the lexical diversity of positive versus negative sentiment expressions and find that expressions with negative prior polarity have a higher *root-ttr* in both corpora. In the hotel corpus

lemma	frequency	share
great	82	8.17%
good	66	6.57%
clean	59	5.88%
nice	45	4.48%
friendly	42	4.18%
comfortable	36	3.59%
excellent	31	3.09%
helpful	29	2.89%
free	27	2.69%
recommend	16	1.59%
	433	43.13%

(a) hotel corpus: positive

lemma	frequency	share
small	38	10.67%
problem	17	4.78%
no	14	3.93%
rude	10	2.81%
bad	10	2.81%
noisy	9	2.53%
not work	6	1.69%
awful	6	1.69%
slow	6	1.69%
complaint	6	1.69%
	122	34.27%

(b) hotel corpus: negative

lemma	frequency	share
great	105	12.25%
good	90	10.50%
love	35	4.08%
like	27	3.15%
nice	27	3.15%
easy	25	2.92%
easy to use	18	2.10%
excellent	18	2.10%
well	16	1.87%
happy	15	1.75%
	376	43.87%

(c) camera corpus: positive

lemma	frequency	share
problem	17	5.30%
bad	10	3.12%
small	7	2.18%
no	7	2.18%
blurry	7	2.18%
grainy	6	1.87%
issue	6	1.87%
not work	6	1.87%
expensive	4	1.25%
slow	4	1.25%
	74	23.05%

(d) camera corpus: negative

Table 6.10.: The ten most frequent positive and negative sentiment expressions in the review corpora.

the numbers are 9.28 (positive) versus 9.61 (negative) and in the digital camera corpus we have 9.47 versus 10.73. The higher lexical variability indicates that it is more difficult to recognize negative sentiment expressions.

Table 6.10 presents the ten most frequent positive and negative sentiment expressions in both corpora. The difference in lexical diversity is mirrored by the higher proportion that the ten most frequent positive expressions exhibit, compared to the ten most frequent negative expressions (43.1% and 43.9% versus 34.3% and 23.1%). The tables further show some examples of sentiment expressions with target-specific prior polarity, such as "free", "small", "slow", "grainy", or "blurry". The negation "no" occurs in the list of negative expressions, as it is often used to express the lack of a desired aspect or behavior.

As the lists of the most frequent sentiment expressions already indicate, the major share of sentiment bearing expressions belongs to the word class of adjectives. This is examined more closely in Table 6.11. The first four rows of the table refer to the counts for single-token sentiment expressions with part-of-speech being either adjective, noun, verb, or adverb. The fifth row refers to single-token sentiment expressions of any other word class, as well as multi-word expressions. In the hotel dataset, we can observe that the skew towards adjectives is more pronounced. 67.3% of all sentiment expressions are adjectives, compared to 54.0% in the digital camera dataset. The dominance of adjectives as part-of-speech is not surprising, since adjectives are the primary word class used to *describe* things. Nevertheless, other parts-of-speech have a significant share and should be covered when detecting sentiment expressions (e.g., by means of a sentiment lexicon). The table further examines the lexi-

	hotel			camera		
word class	**share**	**frequency**	**root ttr**	**share**	**frequency**	**root ttr**
adjective	67.26%	943	6.74	54.01%	646	7.24
noun	8.99%	126	6.77	9.87%	118	5.71
verb	6.35%	89	4.13	9.95%	119	3.85
adverb	4.78%	67	5.74	7.44%	89	5.30
other	12.62%	177	–	18.73%	224	–

Table 6.11.: The distribution of parts of speech of sentiment expressions.

	hotel		camera	
shifter type	**affected expressions**	**frequency**	**affected expressions**	**frequency**
amplifier	19.26%	270	16.89%	202
negator	5.35%	75	7.19%	86
neutralizer	3.07%	43	8.19%	98
downtoner	2.21%	31	2.17%	26
softener	1.00%	14	1.92%	23
solidifier	0.50%	7	0.50%	6

Table 6.12.: The distribution of varying sentiment shifter types in the review corpora.

cal diversity of sentiment expressions with respect to the four most important word classes. In both corpora, the *root-ttr* is lowest for verbs and adverbs. In fact, the five most frequent verbs ("recommend", "love", "enjoy", "appreciate", "like") in the hotel corpus account for 47.2% of all occurrences of sentiment expressions. In the camera dataset, the top five verbs ("love", "recommend", "like", "enjoy", "die") amount to 54.2% of occurrences.

6.2.3. Sentiment Shifters

In this section we analyze the corpora with regard to the distribution of different sentiment shifter types. The statistic presented in Table 6.12 describes the relevance of each type in terms of the number of modified sentiment expressions. Earlier we learned that around 30% of all sentiment expressions are affected by at least one sentiment shifter[8]. The great majority of these expressions is modified through an amplifier. Nearly 20% of sentiment expressions in both corpora (19.3% and 16.9%) are amplified. In consequence, *amplifiers* account for around 60% of all sentiment shifters in the hotel review corpus and 47% in the digital camera dataset. The shifter types *downtoner*, *softener*, and *solidifier* only play a marginal role – in sum they amount to less than 5% of affected expressions in both corpora. With regard to *neutralizers* and *negators* we observe a higher relevance in the digital camera corpus compared to the hotel corpus. In sum, they modify 15.4% of sentiment expressions in the camera dataset and only 8.4% in the hotel review dataset. It is thus more difficult in the camera corpus to detect the correct contextual polarity. Due to the greater effect of negators (potential "polarity flip") and neutralizers ("nullifying" polarity) on contextual polarity, as well as the higher number of affected expressions, these shifter types are of greater importance and should be recognized by sentiment analysis systems.

Tables 6.13a to 6.13c list the five most frequent occurrences of the shifter types *amplifier*, *negator*,

[8] Take note that the sums of the individual relative frequencies presented in this table are greater than the overall proportions of affected sentiment expressions. The reason is that a single sentiment expression may be affected by multiple different shifter types.

lemma	frequency	share
very	226	47.98%
really	36	7.64%
pretty	22	4.67%
extremely	20	4.25%
just	16	3.40%
	320	67.94%

(a) amplifier

lemma	frequency	share
not	119	74.84%
no	13	8.18%
never	4	2.52%
nothing	3	1.89%
without	3	1.89%
	142	89.31%

(b) negator

lemma	frequency	share
if	31	22.63%
look for	11	8.03%
want	9	6.57%
when	7	5.11%
would	7	5.11%
	65	47.45%

(c) neutralizer

Table 6.13.: The five most frequently used phrases for the sentiment shifter types, amplifier, negator, and neutralizer. The numbers are based on a conflation of the hotel and camera review corpora.

and *neutralizer* (we do not present statistics for the other types as the frequency of occurrence is too low). Amplifiers and negators exhibit a very low lexical variability. Nearly 50% of all occurrences of amplification are associated with the adverb "very". With respect to negation, the word "not"[9] (not surprisingly) accounts for even three quarters of all occurrences. The *root-ttr* for amplifiers is 3.45 compared to only 1.51 for negators. The major share (22.6%) of neutralizing modification of sentiment expressions can be ascribed to the conjunction "if", which generally introduces a conditional clause. With a *root-ttr* of 4.61, the lexical variability of neutralizers is higher than the other examined shifter types. We summarize, that in the examined corpora the shifter types amplification, negation, and neutralization play a major role in defining the contextual polarity and intensity of sentiment expression. The lexical variability of amplifiers and negators in particular, but also of neutralizers is rather low, so that relative simple lexicon-based approaches may suffice to detect these kinds of modification.

6.3. Summary

In this chapter we presented the results of an annotation study conducted on our manually labeled sentence and expression level corpora. The major goals of this study were (1) to examine the relevance of the different phenomenons described by our models and (2) to determine whether significant differences between the two considered domains (hotel vs. digital camera) exist. The following enumeration summarizes our most relevant findings:

- Corpus analysis confirms that customer reviews are highly focused documents.
  About 60% of all sentences contain relevant information, that is exhibit a polar expression on at least one of the predefined topics. Strong correlations between polarity and topic relevance are found. 93% of polar sentences are on-topic. A system that only analyzes the polarity dimension thus already achieves a very high accuracy with regard to topic relevance.
- The overall rating of a review is a good indicator for sentiment classification. For example, the probability that a sentence exhibits a positive sentiment is about ten times higher in a five star rated review compared to a one star rated review.
- The imbalance of positive versus negative sentiment is less pronounced when examined on the sentence and/or expression level. On the sentence level, positive sentences occur about twice as often as negatively attributed sentences. Significant shares of sentences in overall negative rated

[9] Our annotation guidelines (cf., Appendix A.3) require that the annotation of contracted negations, as for example in "doesn't", "don't", or "isn't", only covers the *n't* part of the contraction. As this contraction is lemmatized to the lemma "not", all contracted negations are also counted as occurrence of this lemma.

reviews are positive and vice versa. This observation further stresses the need for fine-grained analysis on the sentence or expression level.

- Roughly 15% of all on-topic sentences refer to multiple topics. We conclude that topic categorization on the sentence level is indeed a multi-label, multi-class classification problem.
- Nearly 97% of all sentences can be attributed to one of the predefined discourse functions, showing a very high coverage of the set. The two functions "sentiment" and "fact" cover about 65% of the sentences, 20% correspond to functions that represent an additional information need ("conclusion", "personal context", "advice", "problem" and "lack"). We conclude that it is worth considering the discourse function dimension in a review mining system (but this may depend on the concrete application scenario).
- With regard to sentiment targets in the expression level model, nominal mentions are most important, making up a share of over 80% of all mentions. Detecting pronominal and implicit mentions can improve the recall of a review mining system by at maximum 16%. Named mentions play only a marginal role with a share of less than 2.5%.
- The occurrence frequency of sentiment targets approximately resembles a Zipfian distribution. We find about 500 distinct mentions in both corpora, where the majority of these types occurs only once. The 10% most frequent types make up for roughly 70% of all occurrences of sentiment targets. On the one hand, this favors lexicon-based approaches as small lexicons can already capture the major share of occurrences. On the other hand, it shows the limits of such approaches, as a lexicon can never be large enough to capture the types in the "long tail".
- More than 80% of all sentiment expressions refer to a single word and can be attributed to one of the major parts of speech ("adjective", "noun", "verb", or "adverb") indicating that even simple lexicon-based techniques promise reasonably good results.
- We find that more than 20% of all sentiment expressions exhibit a target-specific polarity. Adjectives account for over 90% of these expressions. We conclude that it is of importance for a review mining system to cope with this phenomenon and it is most reasonable to set priority on detecting target-specific polarity of adjectives.
- Concerning sentiment shifters, we observe that 30% of all sentiment expressions are shifted, where the majority (around 60%) is related to amplification. It is worth considering negation and neutralization, as these types occur significantly often and have great influence on the contextual polarity. The shifter types "downtoner", "solidifier", and "softener" occur very rarely and may thus be disregarded. The lexical diversity of the types "amplifier", "negator", and "neutralizer" is relatively low, so that again simple lexicon-based approaches may suffice.
- The annotation study did not reveal major differences with regard to the two different domains (hotel vs. digital camera).

Part III.

Tasks and Approaches

Not everything that counts can be counted, and not everything that can be counted counts.

William Bruce Cameron

7. Automatic Acquisition of Product Aspect Lexicons

7.1. Overview

Given a review document, the core task of an aspect-oriented customer review mining system is to extract all mentions of product aspects the reviewer has commented on. The relevance or validity of such an extraction can be defined along two dimensions. On the first dimension, relevance is defined in terms of distinguishing factual information from evaluative information. Normally, the task is to extract only mentions that are of evaluative nature. However, as we learned earlier (cf., Section 6.1.2), not all entities that are evaluated are also relevant with regard to the target domain. For instance, when mining hotel reviews, it is irrelevant to know that the reviewer loved the boat trip he experienced on his first day of visiting New York City, but it is relevant to detect that he hated the noisy air conditioning unit. So, on the second dimension, validity of an extraction is defined in terms of its relevance to the target domain. In this chapter, we focus on the second subtask, namely to identify mentions of relevant product aspects in customer review documents. We study unsupervised, lexicon-based approaches. In particular, we cast the task of automatically generating a product aspect lexicon as a *terminology extraction problem*.

The remainder of this chapter is organized as follows. In Section 7.2, we discuss related work concerning the acquisition and application of product aspect lexicons in the context of customer review mining. With regard to the acquisition process, we distinguish unsupervised and supervised methods. Section 7.3 briefly reviews the main concepts in terminology extraction and gives a detailed description of our approach. In Section 7.7, we present an extensive study of the approach with many different configurations and in various evaluation scenarios. Section 7.8 summarizes the most relevant findings and presents our conclusions.

7.2. Related Work

Most basically, the shape of a product aspect lexicon and the process of gathering it can be distinguished by the following four characteristics:

- **Degree of supervision**: This is the most fundamental distinction. A knowledge base can be either manually compiled (i.e., under human supervision) or may be automatically derived by means of an unsupervised[1], algorithmic approach. Of course, automatic acquisition and human supervision can be combined — for example, when manually refining or enriching an automatically extracted lexicon.

- **Definition of domain relevance**: The domain relevance may be either defined with regard to a specific product (e.g., "Canon PowerShot S100") or a whole class of products (e.g., digital cameras). Both definitions are reasonable and depend on the specific task a review mining system should fulfill. Naturally, a product centric definition of relevance allows for a more fine-grained analysis of an individual product. Specific aspects that are only relevant to this product, but not necessarily to the product class in general, can be captured. For example, in

[1] Take note that we distinguish the process of *creating* a lexicon from the the process of *applying* the lexicon. While the creation may be conducted in an unsupervised manner, application might indeed be implemented as part of a supervised approach to product aspect extraction (e.g., as additional machine learning feature).

the domain of restaurant reviews, the taste of the "tagliatelle" may be a relevant aspect for a specific Italian restaurant, but not for any Japanese restaurant or the class of restaurants as a whole. On the other hand, when the task is for instance to provide aspect-based comparisons of different products (of the same genre), a product class/genre centric definition of relevance is preferable. As indicated earlier, our corpora and approaches all follow a class/genre centric definition.

- **Degree of structuring**: A knowledge base for the task of product aspect extraction might be composed of a simple list of relevant terms (aspects) or may exhibit a more complex structuring. Following Buitelaar and Magnini [60], we distinguish five levels of complexity with regard to product aspect knowledge bases. The most basic level is a simple dictionary of terms. More complexity is added when individual aspects are arranged in groups of synonyms. For example, a knowledge base for mp3 players may contain the information that the aspects "headphone" and "earplug" are synonymous. On the next level of complexity, synonym groups are clustered into a set of (abstract) concepts. Such a lexicon might for instance provide the information that the aspects "headphone" and "speaker" both belong to the concept "sound". Organizing concepts or synonym groups hierarchically (as a taxonomy) determines the fourth level of complexity and defining fine-grained relations between concepts refers to the most complex (fifth) level.

- **Coverage**: Corpus analysis has shown that the great majority of nominal product aspects (> 92%) becomes manifest in form of simple noun phrases[2]. However, more complex noun phrases occur and implied product aspects typically do not become manifest as a noun phrase at all. Thus, we can distinguish the coverage of a knowledge base with regard to the type of its entries. A lexicon may comprise only simple noun phrases or other parts of speech. It may also cover spelling variations and common spelling mistakes of its entries. The coverage may be further enhanced by incorporating specific abbreviations that refer to a product aspect.

7.2.1. Unsupervised Approaches to Lexicon Creation

Manually crafting lexical resources for the task of aspect extraction is costly and time-consuming and thus does not scale well to multiple target domains. Unsupervised techniques make the pledge to overcome this disadvantage and meanwhile many studies can be found in the literature that address such approaches — mostly in conjunction with sentiment analysis. In this section, our goal is to distill the most substantial ideas and methods found in related work.

Most unsupervised approaches are corpus-based and in one or another way statistically analyze simple occurrence counts derived from the corpus. Input to each approach is a domain relevant corpus and output is a knowledge base containing identified product aspects. The different proposed techniques can basically be distinguished by the degree of linguistic and domain-specific[3] knowledge that is incorporated, as well as by the specific statistical methods applied.

Frequent Itemset Mining

Hu and Liu [176, 177] present one of the earliest works on aspect-oriented customer review mining. They cast the task of identifying relevant product aspects as a *frequent itemset mining* problem and apply the well-known *Apriori algorithm* [4]. In this context, they define an itemset to be a set of words that have been identified as term candidates and regard each sentence of the input corpus as a single *transaction*[4] — this way, they ensure that only words that occur jointly in a sentence can generate

[2] In this case, we regard noun phrases composed of a single noun (e.g., "breakfast"), a compound noun (e.g., "breakfast buffet"), or a single/compound modified by adjectives (e.g., "continental breakfast"), as simple.
[3] Here, domain-specific refers to the generic domain of customer reviews (e.g., instead of newswire text).
[4] With respect to association rule mining, a transaction is the set of items that happen to occur together (e.g., all items in a single purchase).

multi-word terms. Candidate words are generated by applying a linguistic filter based on part-of-speech tagging and syntactic chunking. Only nouns and noun phrases pass the filter. A minimum support of 1% is applied to extract the frequent itemsets. Although casting the task as a data mining problem and then applying a standard algorithm seems elegant, we believe that frequent itemset mining does not fit well the problem of extracting domain specific terms, for mainly two reasons: First, frequent itemset mining does not consider the order of items, but order of words is obviously important in natural language. And second, subsets of frequent itemsets are inevitably also frequent itemsets (this is basically the *Apriori property*). However, sub-terms of terms are not necessarily valid terms in natural language (e.g., the sub-term "size bed" of the term "king size bed" is not a valid term on its own). Hu and Liu propose to overcome these inherent drawbacks by post-processing the obtained results with two heuristics that try to restore the validity of terms. The complete algorithm produces an unordered list of nouns/noun phrases that are frequent within the target domain. Its evaluation is performed extrinsically (as part of sentiment target extraction) on sentence level annotated corpora (the "Hu and Liu Dataset of Consumer Electronics", cf., Section 5.4.3). The basic frequent itemset mining approach shows a relatively low precision of 56% at a recall of 68%. With application of the two proposed heuristics the precision is increased to 79% at the expense of a marginal decrease in recall of one percentage point. The presented approach relies on several tuning parameters, such as the minimum support, but the effect of these thresholds is not examined. Also, no information about the size and intrinsic accuracy of the extracted lexicons is provided.

Web as a Corpus

Whereas the method by Hu and Liu exclusively examines documents of the target domain, Popescu and Etzioni [304] propose to additionally incorporate the Web as a corpus. To gather term candidates, the target corpus is parsed and nouns and noun phrases are extracted. Only those candidates that occur with a higher frequency than an experimentally set threshold are retained. To increase precision, they utilize a component of their open domain information extraction system "KnowItAll" [123]. This component basically assesses a term candidate's domain relevance by computing the *pointwise mutual information* (PMI) [458] between the candidate term and predefined *meronymy discriminators*[5] that are associated with the target product class. The computed *PMI* score is used as an additional filter to prune irrelevant term candidates. The effectiveness of their method is evaluated by comparison to Hu's results (using the Hu/Liu dataset). They find that the *PMI* assessment, when only using the target domain as a corpus, improves precision in average by six percentage points, however at the same time lowering the recall by 16 percentage points. Additionally incorporating the Web as a corpus[6], reveals an increase of 20 percentage points in precision, whereas recall decreases by seven percentage points. Also this unsupervised approach relies on several tuning parameters (e.g., the minimum support or minimum *PMI* score), but this is not evaluated. It also remains unclear to which extent other components of the "KnowItAll" system are employed during feature extraction process. The reported results are thus very hard to verify.

Statistical Language Models

Wu et al. [445] propose to use *statistical language models*[7] for the purpose of assessing product aspect candidate terms. Again, candidate terms are retrieved by preprocessing the target corpus with a natural language parser and extracting identified nouns and noun phrases (without further justification, Wu et al. additionally extract verb phrases as candidates). A statistical language model is learned on

[5] Meronymy discriminators are lexical patterns that indicate a meronymy (part-of) relation between a term and a product class (e.g., "X of the digital camera", "digital camera has X", or "digital camera comes with X").
[6] As proposed by Turney [389], they query a Web index and utilize obtained page hits as frequency estimates.
[7] see for example Manning and Schütze [249, chap. 6]

an unlabeled corpus of the target domain. Then each candidate term is evaluated, taking its probability of occurrence (according to the learned model) as a score. No specific information about the type of language model (e.g., which *n-gram order* or whether *smoothing* is applied) is given. Also the determination of the threshold parameter for pruning unlikely candidates remains unclear. In case an unigram language model is applied, the method is similar to the frequent itemset mining approach. An unigram language model also disregards the order of words and its probabilities basically resemble the relative frequencies of words. The authors evaluate their approach on the extended version of the Hu/Liu dataset and report a relatively low average precision at 42.8%. Recall is measured with 85.5%, leading to an f-measure of only 57.0%. No mistake analysis is conducted, thus reasons for the low precision are not explained.

Deviation from Background Corpus

Also Scaffidi et al. [332] propose the application of a statistical language model. But opposed to Wu et al., they assess the domain relevance of candidate terms by comparing token probabilities in the target corpus with probabilities in a *background corpus* of generic English[8]. Candidate terms are extracted by a simple part-of-speech tag filter. Only single nouns or two-token compound nouns are considered. Both types are assessed separately, but with the same approach. The actual frequency of occurrence in the target domain is compared with the expected frequency[9] when considering the background corpus. Candidates that occur more often in the target domain than expected are assumed to be domain relevant. A ranking of candidates is calculated by means of the probability that the actual occurrence frequency is truly observed in the target domain. The basic idea of comparing term distributions in a foreground and background corpus is a common approach in NLP. However, Scaffidi et al. do not consider hypothesis testing methods (as for instance proposed by Dunning [108]) to determine the significance of computed probabilities. Also, they only consider unigrams and bigrams, and as they assess them separately, do not handle the case that an unigram only appears as part of a bigram term (e.g., "life" in "battery life"). They only evaluate the precision of their approach (by manual inspection of the generated extractions from a dataset of 5000 reviews). The presented results are quite as at maximum a number of 12 product aspects is extracted for a single product class. This obviously promotes a high precision (85%) at the expense of a very low recall, which however cannot be measured with their evaluation method.

Yi et al. [452] also propose to utilize a contrastive background corpus to determine domain relevant terms. They assess approaches based on *mixture language models* and *statistical hypothesis testing* for candidate selection and find that the latter consistently outperforms the former one. As term candidates they extract noun phrases which adhere to some domain specific, high precision/low recall patterns. Since we use Yi's approach as a baseline system for our experiments, we will provide more details in following sections.

Latent Semantic Analysis

Whereas the previously discussed approaches all generate a simple list of relevant product aspects, Su et al. [368] propose an unsupervised approach that groups extracted aspects to more general concepts. As before, nouns or compound nouns are extracted as candidates. Heuristics based on Web hits combined with a measure similar to the *PMI* score, as well as language specific (Chinese) characteristics prune spurious candidates. Based on the assumption that the use of individual sentiment bearing words (here only adjectives) is highly correlated to specific product aspect concepts, aspects and opinion words are clustered jointly. Similar to the basic idea of *latent semantic analysis* (LSA) [99],

[8] The 100 million word "British National Corpus of spoken and written conversational English" is used [46] as a background corpus.

[9] Expected frequency counts are estimated by assuming that the counts follow a binomial distribution.

an "aspect-by-opinion-word matrix" is constructed that models the co-occurrence statistics. Aspects are similar if they occur with similar opinion words and vice versa. A mutual reinforcement approach is conducted that iteratively clusters aspects, updates co-occurrence statistics, and then clusters opinion words and updates statistics again. For the extraction of product aspects, a very low precision of 52.5% and a relatively high recall of 86.5% is reported. However, the meaningfulness of their experimental evaluation is questionable for mainly two reasons: First, recall is computed as the sensitivity of the pruning steps, but not as the sensitivity of the whole extraction process. In fact, no annotated corpus is used for evaluation. Second, it is unclear whether the language specific heuristics are transferable, for example to English language. The effectiveness of their clustering approach is measured by manually labeling extracted aspects and calculating the similarity between automated and manual labeling using the *Rand index*[10] [311]. Experimenting with different tuning parameters, a maximum accuracy of 75% is achieved for this task.

Further work is for example by Zhang et al. [459] and Qiu et al. [306]. They propose to use a *double propagation approach* to leverage the correlation between aspects and sentiment bearing words. Holzinger et al. [168] use a wrapper algorithm to extract product aspects from tabular data on Web pages and refine their results with an ontology reasoning approach. Blair-Goldensohn et al. [40] basically apply the approach of Hu and Liu [177], but propose additional heuristics for filtering term candidates. Guo et al. [153] extend the basic idea of Su et al. [368] by employing a multi-level LSA approach.

7.2.2. Supervised Approaches to Lexicon Creation

Liu et al. [236] set focus on extracting product aspects from the pros and cons parts of customer reviews, which generally exhibit a high information density and relatively simplistic language. They propose a supervised approach to learning extraction patterns that exploit lexical and shallow parsing features. A human annotator creates a training corpus by identifying product aspects in a set of pros/cons documents. Each aspect (simple or multi-word term) is masked by a generic symbol and stored together with its contextual information of two adjacent words in a transaction file. In a next step they apply *association rule mining* [4] to find frequent patterns of tokens and POS tags that indicate the occurrence of a feature. But as they only use support (and not confidence) to detect rules, what they actually do is just to find frequent itemsets. Again, since frequent itemset mining does not consider the ordering of words, they need to perform extensive post-processing steps to discover valid patterns. As mentioned before, we believe that the association rule mining approach does not fit well in natural language processing tasks. In particular, this approach entails many heuristics and empirically defined thresholds that would render unnecessary with dedicated methods to supervised sequence tagging, such as techniques based on hidden Markov models [32, 310] or conditional random fields [223]. Pros and cons are analyzed separately, that is different sets of patterns are generated. Evaluation is performed by splitting the annotated corpus in a training and test set, but no information about the size of the datasets is provided. Liu et al. [236] report significantly worse results for extraction from cons than for pros documents, which they attribute to the higher lexical variability in cons texts. An approach to grouping synonymous aspects by means of the WordNet dictionary is not evaluated separately.

Also Feiguina and Lapalme [127] propose a supervised method to the creation of product aspect dictionaries. Their approach is basically an application of the terminology extraction system presented in [299]. Based on a training corpus, they first learn a language model on part-of-speech sequences which represent manually labeled product aspects. Then the most likely part-of-speech patterns are used to identify a set of candidate terms. For each candidate several scores are computed (e.g., TF/IDF, raw frequency, term cohesion). These scores, as well as the probability provided by

[10] The *Rand index* examines clustering as a set of pairwise decisions. It calculates the percentage of pairwise correct decisions (true positives and true negatives) and thus can be regarded as a measure of accuracy.

the language model are used as features for a binary classifier that is trained on the labeled corpus. The classifier is then applied to select valid terms from the set of candidates. Extracted terms are automatically grouped by looking for common keywords and by means of their semantic similarity as defined in WordNet. Unfortunately, evaluation is only performed by measuring the accuracy of the created dictionaries. Reported results are relatively mixed, varying from 10% to 88% accuracy, depending on the configuration. Due to the evaluation strategy, no results for recall are provided, thus the applicability of the approach remains unclear.

Yu et al. [457] present a combination of unsupervised and supervised methods that extract and group product aspects hierarchically. For extracting product aspects, they propose to leverage the pros and cons parts of reviews. Frequent nouns and compound nouns from pros and cons are used as samples for training a *one-class support vector machine* (one-class SVM) [248]. The learned model is applied to identify valid terms in a set of candidates extracted from the free text part of the reviews. Unfortunately, they do not provide further information about the feature set and feature representations used to learn the model. It is thus very hard to understand why and in which way the one-class SVM helps to filter term candidates. The achieved results cannot be verified. Furthermore, the proposal to automatically generate a training set by finding frequent nouns and compound nouns in pros and cons is debatable. Although citing Liu et al. [236], in contradiction to Liu's results, they assume a high accuracy of the frequent noun heuristic. For grouping extracted aspects hierarchically, a two-step approach is conducted. First, an initial hierarchy is constructed by exploiting parent-child relations in product specifications. For this task, they rely on an existing wrapper generation approach [449] to extract product specifications from web pages. The second step is to allocate extracted product aspects to appropriate positions in the initial hierarchy. This is performed incrementally, applying a multi-criteria optimization approach which finds the most similar aspect already present in the hierarchy. All criteria assess the similarity of terms based on linguistic features. Evaluation of aspect extraction and hierarchy induction is performed separately on a dataset comprising reviews of 11 different products. Unfortunately, they do not give information about their annotation scheme and the labeling process; for example, it remains unclear whether annotations refer to the sentence or expression level. For the task of aspect extraction, they report an f-measure of 73% in average, improving on the approaches of Hu and Liu [177] and Wu et al. [445] on their dataset. Automated hierarchy generation is evaluated against a manually labeled gold standard, examining pairwise correctness of parent-child relations. Here, an f-measure of 72% is reported in average.

study	supervision	structuring	coverage	linguistic information/features	basic approach
Hu and Liu [176, 177]	unsupervised	list of terms	nouns / noun phrases	part-of-speech tagging, sentiment lexicon	frequent itemset mining + nearest noun phrase heuristic
Popescu and Etzioni [304]	"	"	"	shallow parsing, meronymy discriminators, morphological cues	PMI + Web as a corpus
Wu et al. [445]	"	"	noun / verb phrases	dependency parsing	statistical language models
Scaffidi et al. [332]	"	"	noun uni/bigrams	part-of-speech tagging	statistical language models + contrastive background corpus
Yi et al. [452]	"	"	nouns / noun phrases	part-of-speech tagging + beginning definite noun phrase heuristic	likelihood ratio test + contrastive background corpus
Su et al. [368]	"	concepts	nouns / compound nouns	part-of-speech tagging + language specific heuristics	PMI + Web as a corpus + aspect-sentiment co-occurrence analysis
Liu et al. [236]	supervised	synonyms	arbitrary n-grams	part-of-speech tagging + WordNet	pattern learning with frequent itemset mining
Feiguina and Lapalme [127]	supervised	concepts	arbitrary n-grams	"	statistical language models + binary classifier
Yu et al. [457]	unsupervised / supervised	hierarchy	nouns / noun phrases	full parse	one-class SVM + multi-criteria optimization (for hierarchy generation)

Table 7.1.: Unsupervised and supervised approaches for product aspect lexicon creation.

7.3. Terminology Extraction

The overall goal of a terminology extraction system is to identify domain relevant terms in a given text corpus. The input to such a system is a collection of documents of the target domain and the output is a (typically ranked) list of identified, relevant terms. Often, and as is the case in our context, the task of terminology extraction is a preceding step to more complex tasks. For example, common fields of application are the creation of specialized glossaries [220, 398], ontology learning [60, 277], or text mining in biomedical corpora [12, chap. 4].

7.3.1. Pipeline Architecture

A typical system for terminology extraction follows the pipeline architecture depicted in Fig. 7.1. The extraction process is decomposed into the following five steps:

1. **Linguistic pre-processing:** Depending on the degree of linguistic analysis, different steps of pre-processing are performed on the document collection. Commonly, the textual data is tokenized, split into sentences, tagged with part-of-speech symbols, and individual tokens are stemmed or lemmatized.
2. **Acquisition of candidate terms:** Based on the results of the first step, application specific, linguistic filters can be applied to find appropriate candidate terms. For example, part-of-speech tag patterns are often used to extract all nouns and noun phrases. Domain dependent heuristics may be used to increase the precision of the acquisition step.
3. **Candidate filtering:** In this step, additional domain specific heuristics are applied to prune candidate terms — for instance, all named entities may be discarded or known "stop terms" may be removed.
4. **Aggregation of term variants:** A single term may become manifest in different forms on the textual surface — for example, a term may be misspelled or occur in plural and singular form. Such variants are aggregated and mapped to a single canonical form of a term.
5. **Candidate ranking and selection:** The system computes the relevance of each acquired candidate and ranks the terms according to this score. Most commonly, statistical measures based on occurrence counts are employed to define relevance, but also domain dependent heuristics may play a role. Finally, a criterion for selecting the most relevant terms is applied.

Because the definition of term relevance is *the* central aspect in terminology extraction, and since the underlying concepts are applicable across domains, we elaborate on them in more detail.

7.3.2. Definitions of Term Relevance

A multitude of statistical measures for defining term relevance is proposed in the literature on terminology extraction[11]. Whereas the individual approaches differ widely, most basically, we can identify the following three underlying concepts of term relevance:

Contrastive Domain Relevance

This concept defines the relevance of a term by means of contrastive analysis. The basic idea is that a term has a higher (relative) domain relevance if it occurs frequently in the target domain and relatively infrequently in a collection of contrastive documents. Typically, the collection of domain relevant documents is denoted as *foreground corpus*, whereas the contrastive documents form the *background corpus*. The most obvious measure to determine the relative domain relevance of a term is to

[11] For example, Kageura and Umino [199] or Wong [440] provide literature surveys of terminology extraction.

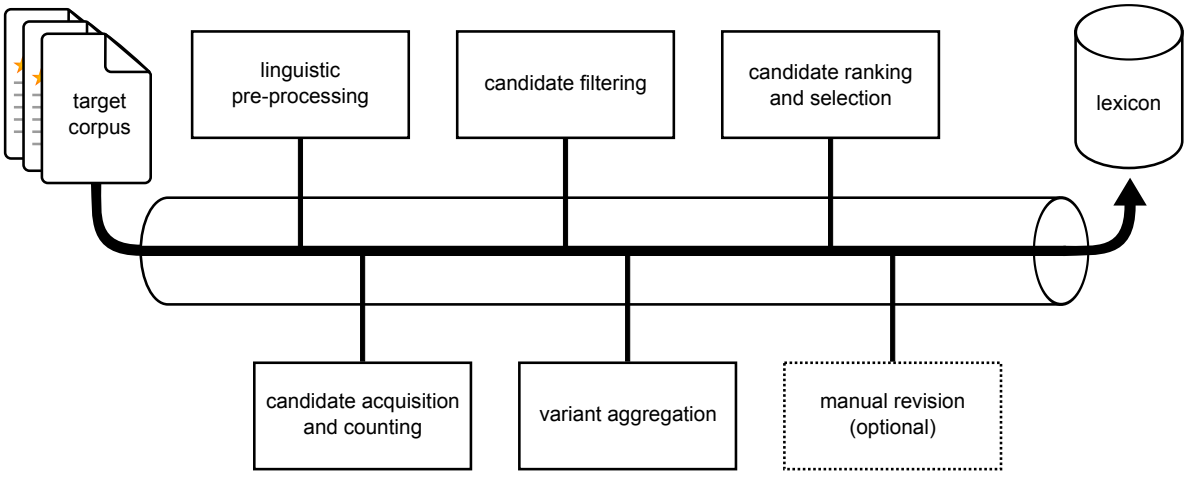


Figure 7.1.: A pipeline architecture for terminology extraction.

utilize the ratio of relative frequencies in both corpora as a score [94, 110]. Ironically, the literature on terminology extraction does not provide a consistent term for the concept of relative domain relevance. It is also referred to as *domain specificity* [298], *domain relevance* [398], *domain pertinence* [336], or *informativeness* [385].

Intra Domain Relevance

Opposed to a contrastive analysis, *intra domain relevance* (e.g., Wong [440]) refers to measuring the significance of a term by studying its distributional behavior within the target corpus. For instance, the most simplistic measure would be the raw frequency of a term candidate in the corpus; significant terms are assumed to occur with higher frequency than other terms. Other measures rely on the cross document distribution of term candidates. For example, Navigli and Velardi [276] assume that a candidate exhibits a high relevance if it is evenly distributed across the corpus. They propose an entropy based measure, which they denote as *domain consensus*.

Term Cohesion

Whereas both previously mentioned relevance concepts are applicable to simple terms (single word) as well as complex terms (multiple words), the concept of *term cohesion* addresses only the significance of complex terms. Kageura and Umino [199] refer to this notion as *unithood* of terms and define it as the "degree of strength or stability of syntagmatic combinations or collocations". That is, term cohesion measures the strength of association between the individual words that compose a complex term. The most common statistical methods used to determine a score for the lexical cohesion are thus closely related to techniques for finding *collocations* in natural language text. Hypothesis testing, such as *likelihood ratios* [108], or information theoretic measures, such as *PMI* [458], are frequently proposed. But also more linguistic informed methods, such as the *NC-value* approach [132] or measures that build upon the limited *compositionality*[12] of collocations [407], are proposed.

7.4. Terminology Extraction for Product Aspect Detection

We now discuss our terminology extraction approach to product aspect detection in detail. Sections 7.4.1 to 7.4.6 describe how we implement the individual steps of the extraction pipeline (for

[12] cf., Manning and Schütze [249, chap. 5]

the majority of steps, we propose several alternative approaches, which will be subject to experimentation in Section 7.7).

7.4.1. Linguistic Preprocessing

We pre-process all text documents by means of the "Stanford CoreNLP" natural language analysis tool[13]. In particular, we employ the POS tagger component [387], which performs tokenization, sentence splitting, POS tagging, and lemmatization of the input text. We do not train a new model for the POS tagger component on the collection of review documents, but rely on the default model, which is trained on the *Penn Treebank* [251] — consequently, we use the *Penn Treebank tag set*[14]. All tokens are further normalized by lowercasing.

7.4.2. Candidate Acquisition

The candidate acquisition component initially decides which phrases are further considered and which of them are directly discarded. The definition of these filters potentially has a great influence on the recall and precision of the whole extraction process. Defining too restrictive filters may significantly lower the recall, whereas too unconstrained filters may decrease the system's precision. We therefore experiment with two different filters and compare their performance.

Part-of-Speech Tag Filter

Based on the observation that nominal aspect mentions are nouns or noun phrases, we define appropriate linguistic filters as part-of-speech tag patterns. The patterns are defined as regular expressions[15] over strings of POS tags. We define two different patterns for candidate extraction:

The first "base noun phrase pattern" (BNP1) corresponds exactly to the POS tag filter applied in [453]:

```
BNP1 := NN |NN NN |JJ NN |NN NN NN |JJ NN NN |JJ JJ NN
```

This pattern restricts candidates to be composed of at maximum three words. The last word must be a noun (NN) which may be preceded by other nouns or adjectives (JJ). Adjectives must only occur as pre-modifiers to nouns. This pattern would for example match the term "intelligent/JJ auto/JJ mode/NN".

We further examine the utility of a more relaxed pattern (BNP2). The pattern matches terms of arbitrary length. It also allows for plural forms and matches proper nouns (identified by the tags NNP or NNPS):

```
BNP2 := (JJ )*(NN\w{0,2} )+
```

Domain Specific Heuristics

All presented heuristics follow the principle of applying *low-recall/high-precision* patterns to very large corpora (the basic assumption is that large corpus sizes compensate for low recall). Again, we follow Yi and Niblack [453]. They propose a set of domain specific heuristics for candidate identification which extend the basic part-of-speech tag filter. The heuristics are assumed to increase the precision of the POS tag filter.

[13] http://nlp.stanford.edu/software/corenlp.shtml
[14] ftp://ftp.cis.upenn.edu/pub/treebank/doc/tagguide.ps.gz
[15] We use the regular expression syntax as defined for the Java programming language (see http://docs.oracle.com/javase/6/docs/api/index.html).

- **Definite Base Noun Phrase (dBNP)**: Originally, this heuristic restricts the *BNPs* to phrases which are preceded by the definite article "*the*". Based on an analysis of the held out dataset[16], we extend the heuristic to also allow for demonstrative determiners such as "this" and "that".

- **Beginning Definite Base Noun Phrase (bBNP)**: Based on the observation that a shift in focus with regard to the discussed product aspects is often introduced by naming the new aspect at the beginning of a sentence, Yi and Niblack [453] propose the *bBNP* heuristic. It further restricts valid candidates to *dBNPs* that occur at the beginning of a sentence and that are followed by a verb phrase (e.g., "The *picture quality* is great.").

As our corpus analysis has shown, a strong correlation between mentions of product aspects and sentiment bearing phrases exists. We therefore hypothesize that incorporating knowledge about the existence of sentiment expressions in the context of a candidate can improve the precision of the extraction process. We employ a small, hand-crafted and domain-independent sentiment lexicon for the purpose of finding sentiment bearing expressions. The lexicon consists of 520 adjectives and verbs with strongly positive (170) or strongly negative (350) prior polarity. We assume that such a small lexicon guarantees a high precision for determining sentiment expressions (possibly at the expense of coverage). With regard to incorporating this knowledge, we experiment with two different strategies:

- **Occurrence in Sentiment Bearing Sentence (SBS)**: Using the sentiment lexicon, we mark each sentence as either sentiment bearing or factual. Only candidates that occur in sentiment bearing sentences are extracted.

- **Occurrence in Sentiment Bearing Pattern (SBP)**: For this heuristic, we define a set of very simple syntactic patterns that relate candidate terms to sentiment expressions. Only candidates that match one of these patterns are further considered. We employ the following patterns:

  – Preceding adjective: A sentiment bearing adjective directly precedes the candidate term. For instance, this pattern can extract phrases such as "*horrible* battery life".

  – Succeeding adjective: A sentiment bearing adjective directly succeeds (potentially modified by an adverb) the candidate term. This pattern takes care of the fact that reviewers often use informal, grammatically incorrect language. For example, in a sentence such as "They have been always very helpful, service really *fantastic*." the pattern can match the candidate "service".

  – Simple predicative construction: A sentiment bearing adjective (possibly modified by an adverb) is connected to the candidate term via a simple predicative construction. This pattern can match constructions such as
  "front desk staff was extremely *rude*".

  – Simple verb construction: A sentiment bearing verb directly precedes (ignoring any determiner) the candidate term, matching phrases such as
  "... *loved* the breakfast buffet".

7.4.3. Candidate Filtering

Although the heuristics applied for candidate acquisition focus on high precision, they generate a considerable number of irrelevant candidates that can be pruned by domain specific filters. We employ the following types of filtering:

[16] cf., Section 7.6

Review Stop Word Filter

We compile a list of review specific *stop words* and discard each candidate term that contains at least one of the words. Opposed to general purpose stop word lists which comprise all types of word classes and which are domain independent, we can restrict the stop words to a set of nouns relevant for the domain of customer reviews. The list has been constructed based on observations on the held out dataset and by (intelligent) extrapolation of these findings. It has a size of 176 entries. Roughly categorized, it includes the following types of nouns:

- Sentiment bearing nouns: complaint, con, disappointment, drawback, problem, advantage, plus, positive, pro, recommendation, etc.
- Review related: bottom line, competition, consumer, review, reviewer, rating, test, etc.
- Purchase related: deal, delivery, product, purchase, retailer, shipping, salesman, sales person, seller, vendor, etc.
- Mentioning of persons: baby, folk, friend, guy, husband, people, wife, family, lady, kid, etc.
- Review related websites: amazon, epinion, buzzillions, tripadvisor, etc.
- Reasoning on product: reason, decision, surprise, question, idea, etc.
- Others: past, future, need, lack, ...

Measuring Unit Filter

Frequently, numerical modifiers of a product aspect (e.g., "*3x* optical zoom", "*512MB* memory card", or "*10MP* resolution") pass the part-of-speech tag filter. One may argue that they represent a valid part of the aspect, but we believe that they overspecify the particular aspect. We thus remove any numerical modifier of a candidate term that refers to a measuring unit. To do so, we employ simple regular expressions.

Pre-Modifier Filter

Both presented part-of-speech filters for candidate acquisition allow nouns to be modified by multiple adjectives. Whereas these kind of patterns are necessary to acquire candidates such as "intelligent/JJ auto/JJ mode/NN", it leads to the extraction of many invalid terms (e.g., "great/JJ design/NN", "modern/JJ design/NN", or "new/JJ design/NN").

In our context, we can mainly distinguish between two types of pre-modifiers that need to be filtered out. The first type refers to *sentiment bearing adjectives* with stable (i.e., domain/target independent) prior polarity (e.g., "great", "fantastic", "bad", or "horrible"). For these we can simply use a stop word list, typically based on a general purpose sentiment lexicon.

The other type is related to adjectives that act as *universal modifiers* in term candidates (e.g., "new", "long", "other", or "red"). The main problem is that the status of these adjectival pre-modifiers is specific to the domain or even a concrete product aspect. For instance, the adjective "intelligent" is part of the term when referring to the word sequence "intelligent design" as an (odd) creationists' proposition, but is not part of a term when referring to the intelligent design of a digital camera. Also the modifier "red" is typically not part of a term (e.g., "red camera"), but in the term "red eye reduction" it is. As a consequence, we cannot simply provide a stop word list for these type of modifiers. We experiment with two different approaches for filtering product aspect specific adjectival modifiers. The first method is based on a filter proposed by Kozakov et al. [220] as part of their *GlossEx* glossary extraction system. The second technique is an indirect crowdsourcing approach that uses signals from pros and cons summaries of customer reviews (see Section 7.5).

GlossEx Filter This filter measures the *contrastive domain relevance* of the pre-modifier and the *lexical cohesion* between modifier and head noun. The contrastive relevance is calculated as the ratio of relative frequencies. Let $f_+(pm)$ be the frequency of the pre-modifier pm in a foreground corpus C_+ and $f_-(pm)$ be the frequency of pm in a background corpus C_-. Then the ratio of relative frequencies *rrf*(pm) is defined as

$$\mathbf{D} := \mathit{rrf}(pm) = \frac{\frac{f_+(pm)}{|C_+|_t}}{\frac{f_-(pm)}{|C_-|_t}}, \tag{7.1}$$

where $|C_+|_t$ and $|C_-|_t$ refer to the size (in tokens) of the foreground and background corpus. The lexical cohesion is computed as the conditional probability $Pr(head|pm)$, where $head$ is the head noun, which is defined as the first noun succeeding the pre-modifier pm. Using maximum likelihood estimates, we compute the conditional probability as

$$\mathbf{A} := Pr(head|pm) = \frac{f_+(head, pm)}{f_+(pm)}, \tag{7.2}$$

where $f_+(head, pm)$ denotes the joint frequency of the head noun and the pre-modifier in the foreground corpus. Kozakov et al. [220] suggest to apply Algorithm 7.1 to decide on the status of the pre-modifiers:

Algorithm 7.1 GlossEx pre-modifier filter

if $D < \delta_{lower}$ and $A < \alpha_{upper}$ **then**
  remove pre-modifier
else if $D \geq \delta_{upper}$ and $A \geq \alpha_{lower}$ **then**
  keep pre-modifier
else if $A \geq \alpha_{intermediate}$ **then**
  keep pre-modifier
else
  remove pre-modifier
end if

A pre-modifier is removed if its domain relevance D is very low (smaller than the low threshold δ_{lower}), unless the lexical cohesion (association) A is very strong (greater than the upper threshold α_{upper}). If the domain relevance D is strong, that is, greater or equal than the upper threshold δ_{upper}, we keep the pre-modifier, unless the lexical cohesion is very low (i.e., lower than the threshold α_{lower}. In case the domain relevance is within the interval $[\delta_{lower}, \delta_{upper}[$, we retain the pre-modifier only if its lexical cohesion is greater than the intermediate threshold $\alpha_{intermediate}$, otherwise it is removed. We present the concrete parameter settings in Section 7.6.

Product Name Filter

For the task of creating a lexicon of product aspects we are only interested in finding nominal mentions. We thus want to discard all candidate terms that refer to product or brand names. Again, we use a list of stop words for this filtering step. The lists are automatically generated, based on meta data that is associated with the collection of crawled customer reviews. Each review in our collection is accompanied with information about the product name and optionally (depending on the review site) with the name of the producer. Applying simple regular expression patterns, we extract for each product class a list of brand and product names. Whenever a term candidate contains a token that is present in the appropriate stop word list, the candidate is discarded.

7.4.4. Variant Aggregation

The goal of this step is to find all variants of a term and to identify a canonical representation. For example, the variants "auto-focus", "auto focus", "autofocus", or "auto focuss" should all be mapped to the canonical form "auto focus". The purpose of this step is twofold: First, by morphologically clustering variants and associating them to a single representative, a resulting term lexicon has a higher coverage and we can thus assume that it exhibits a higher recall when utilized for extracting terms. Second, as we can aggregate and combine the frequencies counted for all variants of a term, we prevent potential problems with data sparseness and can achieve more significant results during candidate ranking and selection. Following Park et al. [298], we distinguish *inflectional*, *symbolic*, *compounding*, and *misspelling* variants. In addition, we consider *compositional* variants:

Inflectional Variants This type of variant addresses the modification of words caused by the grammatical context. With regard to nouns, inflection refers mostly number (singular or plural) and case (nominative or possessive) in English. Adjectives in English are generally not declined.

We utilize the lemmatization provided by the "Stanford CoreNLP" tool chain. For single word terms, we use the lemma of a word as the canonical representative and associate all inflected forms as variants (e.g., "hotel", "hotels", and "hotel's" are mapped to the lemma "hotel"). Take note that our part-of-speech tag patterns restrict single word terms to be nouns. For complex terms, we only lemmatize the last word in the multi word sequence (e.g., we normalize "parking garages" to "parking garage" and not to "park garage"). As our POS tag filters restrict adjectives and verbs to be premodifiers, in effect, only nouns are affected by this variant aggregation step.

Symbolic Variants We try to normalize orthographic variants that are caused by the diverse usage of symbols for concatenating words. For instance, the forms "heating/air-conditioning unit", "heating&air-conditioning unit", or "heating/air-conditioning-unit"
need to be recognized as symbolic variants of the same term. To identify this type of variant, we employ a simple heuristic: We first remove all relevant symbols (e.g., "/", "-", "&", or "+") and then split the word sequence at the boundaries defined by these symbols (e.g., "air-conditioning-unit" is normalized to "air conditioning unit"). Then, we group all variants with the same normalized word sequence and choose the variant with the highest frequency as the canonical form. In case of two or more forms exhibiting the same frequency, we decide for the form with the least amount of symbols.

Compounding Variants Other orthographic variants are caused by varying use of word boundaries. For instance, reviewers may spell "air conditioning unit" or "airconditioning unit". A further example is "auto focus" or "autofocus". We find such variants with an algorithm that simply removes all whitespace characters in multi word candidates and looks for "join partners". For instance, both "air conditioning unit" and "airconditioning unit" are mapped to "airconditioningunit" and thus can be joined (i.e., can be recognized as variants). Again, we choose the variant which occurs most often as canonical form. In case of a draw, we decide for the most separated form.

Misspelling Variants A third type of orthographic variation is due to misspellings of words. We identify out-of-vocabulary words and try to match them to a correct variant. To do so, we apply Algorithm 7.2.

Algorithm 7.2 Misspelling Variant Aggregation

train spell checker S on the target corpus
$C \leftarrow set\ of\ normalized\ candidate\ terms\ ordered\ by\ frequency$
for all $c \in C$ *in ascending order* **do**
  if c *is out of vocabulary and* $char\_length(c) \geq \gamma$ **then**
    $c^* \leftarrow$ CORRECT(c)
    if $c^* \neq NULL$ **then**
      add c as variant to c^*
      $C \leftarrow C \setminus \{c\}$
    end if
  end if
end for
function CORRECT(candidate)
  $V \leftarrow top\ five\ corrections\ from\ S$
  $V^* \leftarrow \emptyset$
  for all $v \in V$ **do**
    if $v \in C$ *and* $edit\_distance(v, candidate) \leq \lceil \epsilon * char\_length(candidate) \rceil$ **then**
      $V^* \leftarrow V^* \cup \{v\}$
    end if
  end for
  if $V^* = \emptyset$ **then**
    return *NULL*
  else
    return variant $v \in V^*$ with maximum frequency
  end if
end function

First, we train a spell checker on the whole target corpus. For this purpose we use the tools provided by the *LingPipe*[17] text processing tool kit. In particular, we train a *context-sensitive spell checker* based on *character language models* and *weighted string edit distance*[18]. Then, for each candidate term, in ascending order of their frequency, we check whether it needs to be corrected. We only try to correct candidates that are composed of at least $\gamma = 4$ characters and which are out-of-vocabulary. We find out-of-vocabulary candidates by looking up each word of the candidate in an English language dictionary (we use WordNet [263]). If at least one word is not contained in the dictionary, the candidate is assumed to be out-of-vocabulary.

Our procedure for correcting candidates is as follows: We use the trained spell checker to acquire the five most probable corrections. For each proposed correction, we look up whether it is a known candidate term, all other corrections are discarded. If the string edit distance between the correction and the original candidate is less than a threshold, we add it to a set of valid corrections. The applied threshold value is dependent on the length (in characters) of the candidate term. The intuition is to allow for more misspellings in longer terms. We set the parameter $\epsilon = \frac{1}{8}$, so that the allowed edit costs increase by one every eighth character. From all valid corrections, we choose the correction with the highest frequency. In case of a draw, we choose the one which is most probable according to the spell checker. If our correction procedure successfully finds a correction, we add the original candidate as variant to the correction and remove the candidate from the set of candidate terms.

[17] http://alias-i.com/lingpipe/index.html
[18] Kukich [222] provides a summary of techniques for correcting words in text.

Compositional Variants In addition to the morphological variants considered in Park et al. [298], we process variants that are based on different forms of composition used to express a semantic relation between terms. For instance, to express the meronymy relation (part-of) between the terms bag and camera we may use the compositional forms "bag of the camera" or "camera bag". We have identified this type of variation to be very common in customer review documents. To find these kinds of compositional variants, we apply the following algorithm:

We decompose each complex term into the single last noun[19] *C2* and the preceding modifiers *C1*. For example, we decompose "front desk staff" into *C1* = "front desk" and *C2* = "staff". Then we compose a variant pattern *V* = "*C2 of (the | this | that)? C1(s)?*", which effectively reorders *C1* and *C2* and concatenates both parts with the expression "of" plus an optional determiner. This matches for instance the phrase "staff of the front desk" (which admittedly sounds odd in this particular case). We also match phrases in plural form by simply adding the letter "s" — for example, we can derive "detection of red eyes" from the term "red eye detection". We count the number of word sequences in the target corpus that match the generated pattern. If the aggregated counts have a minimum support of σ, we add the composition with all possible forms (different determiners, as well as singular or plural form) as variants to the complex term. Take note that for our application scenario (aspect detection in reviews) this frequency-based pruning step can perfectly be omitted. False conversions, such as "card of the memory" ("memory card"), do not harm the product aspect detection task. Such lexicon entries are simply very unlikely to ever match. However, we prune such candidates for the purpose of a faster detection process, which is guaranteed with smaller lexicon sizes.

7.4.5. Candidate Counting

It is not directly obvious how to count candidate terms. In our context, the basic question is to which extend we interweave the process of generating candidate terms with counting them. In the following, we consider three alternatives:

Filtered Counting The previous steps, candidate acquisition, candidate filtering, and variant aggregation create a final list of candidate terms. During this process several heuristics are applied to increase the precision of the generated list (e.g., the *bBNP-filter* or the *SBP-filter*). If we count only those occurrences that pass these filters, we speak of filtered counting. For example, with regard to the *bBNP-filter*, we recognize and count the occurrence of the term "picture quality" in the sentence "The picture quality is ...", but do not count it in a sentence such as "I really like the picture quality and ...". In case of filtered counting, we can directly interweave the counting of candidates with the pipelined processing of the target corpus. Observe that applying filtered counting is contraindicated in case a background corpus is employed to determine contrastive relevance scores. To achieve accurate statistics, the filter would need to be applied also in the background corpus. But as the filter is domain specific (e.g., searches for sentiment words in context) the statics are likely to be heavily biased.

Open Counting We may also strictly decouple candidate generation and counting. In this case, we regard the generated list of candidate terms as a dictionary and count all occurrences of dictionary entries in the corpus — disrespecting any of the previously applied filters except the POS tag filter[20]. We denote this method as open counting. Considering the previous example, we also count the occurrence in the second sentence with open counting. Assuming that the candidate generation process exhibits a good precision, the intuition for open counting is that it avoids data sparseness and can increase the confidence of statistical methods applied for ranking.

[19] Take note that the choice of our part-of-speech tag filters guarantees that the last word of a complex term is a noun.
[20] For the purpose of a very shallow word sense disambiguation method, we do not disregard the part-of-speech filter in open counting.

Weighted Counting Weighted counting basically represents a compromise between filtered and open counting. We count any occurrence of a candidate term as in open counting, but weigh occurrences that match the low-recall/high-precision filters with a higher score. Considering the previous example, we may count the occurrence in the first sentence with a value of 2 and in the second sentence with 1.

7.4.6. Candidate Ranking and Selection

The previous steps generate an unordered list of normalized term candidates and associated variants. We denote this list as T. The goal of this step is to rank the candidates, so that we can select those which most likely represent valid product aspects with regard to the target domain. In the following, we present the statistical measures we apply and compare in our experiments. Tables 7.2 and 7.3 define the notation we use for subsequent descriptions. Additional notation is introduced as necessary.

	$\mathbf{C}_+$	$\mathbf{C}_-$
ws	$D_{11}(ws)$	$D_{12}(ws)$
$\neg$ **ws**	$D_{21}(ws)$	$D_{22}(ws)$

(a) observed document frequency

$\mathbf{C}_+$	$\mathbf{C}_-$
$f_+(ws)$	$f_-(ws)$

(b) observed word frequency

Table 7.2.: Notation for frequencies of word sequences (ws) in a foreground corpus C_+ and a background corpus C_-. The notation "$\neg$ ws" refers to the set of documents that does not contain the word sequence.

notation	description
T	List of candidate terms.
$\|C\|_d$	Size of corpus C in documents.
$\|C\|_t$	Size of corpus C in tokens.
$\|ws\|_t$	Size of word sequence ws in tokens.
$\|ws\|_c$	Size of word sequence ws in characters.
T_{ws}	Set of candidate terms that contain the word sequence ws, excluding ws.
T^*_{ws}	Set of candidate terms that contain the word sequence ws, including ws.

Table 7.3.: Notations used in the context of candidate term ranking.

Raw Frequency (Intra Domain)

This represents a base line with regard to more elaborated methods. We simply define the ranking order by descending raw frequency — that is, we rank candidate terms c in descending order by $f_+(c)$.

Relative Frequency Ratio (Contrastive)

As we have noted earlier, relative frequency ratios measure term relevance as contrastive domain relevance. We defined the measure for a single word in Eq. (7.1). Following Park et al. [298], we

extend the measure by defining the relative frequency ratio for a multi-word term ws as

$$\text{mrrf-score(ws)} = \frac{\sum_{w_i \in ws} \frac{\frac{f_+(w_i)}{|C_+|_t}}{\frac{f_-(w_i)}{|C_-|_t}}}{|ws|_t} = \frac{\sum_{w_i \in ws} \frac{Pr_+(w_i)}{Pr_-(w_i)}}{|ws|_t}. \tag{7.3}$$

In case of a multi-word term, the measure calculates the average relative frequency ratio of all words contained in the term. For a single word term it equals the definition of Eq. (7.1).

Likelihood Ratio Test (Contrastive)

The *likelihood ratio test* (LRT) is a parametric hypothesis test which is used to compare two different hypotheses H_0 and H_1 with regard to the fit to observed data. The LRT computes the ratio $\lambda = \frac{L(H_0)}{L(H_1)}$ of the likelihood $L(H_0)$ for the null hypothesis and the likelihood $L(H_1)$ for the alternative hypothesis. The value λ expresses how much more likely it is to make a certain observation under the assumption H_0 than under the assumption H_1. As it is known that the value $-2 \ln \lambda$ is asymptotically χ^2-distributed [430], the likelihood ratio can be used to reject H_0 at a given confidence level. Dunning [108] proposes to use the LRT to find collocations in text corpora and argues that in this context the test is more robust with respect to data sparseness than for instance the χ^2-test or the t-test. For ranking term candidates, we apply the LRT to compute a measure of contrastive relevance as proposed by Yi et al. [452]: We are interested in the two conditional probabilities

$$\begin{aligned} p_1 &= Pr(d \in C_+ | c \in d) \\ p_2 &= Pr(d \in C_+ | c \notin d), \end{aligned}$$

where p_1 expresses the probability that a document d stems from the foreground corpus, given that the candidate term c is contained in d and p_2 refers to the probability that d stems from the foreground corpus, given that c is not contained in d. As null hypothesis H_0, we assume that $p_1 = p = p_2$, that is the probability that a document is domain relevant is independent of whether a candidate is contained in the document or not. As alternative hypothesis H_1, we assume that $p_1 \neq p_2$, that is we assume a dependence on the occurrence of c in d. Using maximum likelihood estimates for p, p_1, and p_2 we calculate the LRT-score as:

$$\text{LRT-score} = \begin{cases} -2 \log \lambda & \text{if } p_1 > p_2 \\ 0 & \text{if } p1 \leq p_2 \end{cases} \tag{7.4}$$

Appendix C.1.1 describes more details on how to estimate the probabilities and how to calculate λ. The relation $p_1 > p_2$ basically expresses that it is more likely that a document d stems from the foreground corpus when the candidate term c occurs in d, than when it not occurs in d. That is, terms for which $p_1 > p_2$ holds are likely to be domain relevant. As we are only interested in these terms, we set the LRT-score to zero for all other terms. The higher the LRT-score, the higher is the confidence that $p_1 > p_2$ (i.e., the more confident we are that c is domain relevant).

Generalized Dice Coefficient (Term Cohesion)

To measure the association between words of a complex term, Park et al. [298] introduce a measure that generalizes and adapts the Dice coefficient [101]. The measure aims at giving higher scores to terms which exhibit high co-occurrence frequencies:

$$\text{GDC-score(ws)} = \frac{|ws|_t * \log_{10} f_+(ws) * f_+(ws)}{\sum_{w_i \in ws} f_+(w_i)} \tag{7.5}$$

Take note that for single word terms the GDC-score equals the decadic logarithm of the term frequency. Park et al. [298] propose to reduce the score for single word terms by multiplying with a factor $0 < \rho < 1$ (we define ρ in Table 7.4).

Diversity Value (Intra Domain)

Based on the observation that nested word sequences, which appear frequently in longer terms, are likely to represent the key parts or features of a product, we propose a measure that gives higher scores to such "key terms". For example, consider the word "lens". In the domain of camera reviews, we find it nested in terms such as "autofocus lens", "zoom lens", "macro lens", "lens cap", or "lens cover". On the other hand, words such as "cap" or "cover" are less likely to occur in many different contexts. Under the assumption that reviewers tend to comment on the key aspects of a product, it is reasonable to give such terms a higher weight. We hypothesize that the diversity of contexts, that is, the amount of longer terms that contain a candidate c, is a good indicator for the association of c with the product class under consideration. Inspired by the *C-Value score* proposed by Frantzi and Ananiadou [133], we define the measure as

$$\text{diversity-score(ws)} = \underbrace{\log_2(|ws|_t + 1)}_{\mathbf{A}} * \underbrace{\frac{\sum_{w_i \in ws} (f_+(w_i) * \log_2(|T^*_{w_i}| + 1))}{|ws|_t}}_{\mathbf{B}}, \tag{7.6}$$

where the notation T^*_x refers to the set of candidate terms that contain x as nested term, including x itself, and $|T_x|$ is the cardinality of that set. The *diversity-score* has the following properties: The factor *B* in Eq. (7.6) represents the average frequency of words w_i contained in the word sequence ws, weighted by the diversity of each component w_i. The factor exhibits a high value if the individual frequencies and diversity values are high. For a single word term, the factor *B* represents the raw frequency weighted by the diversity. Take note that the factor does not measure term cohesion; we do not consider the co-occurrence counts. To compensate the fact that *B* favors single word terms (*B* for a complex term cannot be larger than *B* for any of the term's components), we multiply with factor *A*, which is higher for longer terms.

Combining Ranking Measures

As the presented ranking measures are based on different definitions of term significance, it is reasonable to compute a combined score (e.g., combining a term's contrastive relevance with its strength of cohesion). We consider two different methods to combine ranking measures:

- **Weighted sum:** The final score is computed as a linear combination of the n selected individual scores
$$score_{final} = \sum_{i=1}^{n} \omega_i * score_i \text{ , where } \sum_{i=1}^{n} \omega_i = 1. \tag{7.7}$$
- **Weighted rank:** Let $R_i(t)$ be a ranking function that orders candidates $t \in T$ based on the score $score_i$. Considering n selected measures, we compute the final rank of a candidate t as
$$R_{final}(t) = \sum_{i=1}^{n} \omega_i * R_i(t) \text{ , where } \sum_{i=1}^{n} \omega_i = 1. \tag{7.8}$$

Observe that both methods indeed result in different rankings of candidates. For instance, suppose that $score_1(t_1) = 30$, $score_1(t_2) = 20$, $score_1(t_3) = 10$, $score_2(t_1) = 100$, $score_2(t_2) = 300$, and $score_2(t_3) = 200$. Assuming $\omega_1 = \omega_2 = 0.5$, the weighted sum ranks t_2, t_3, t_1 and the weighted

rank results in the order t_2, t_1, t_3. Defining the weights for the weighted rank is more intuitive as the ranking functions take values in the same co-domain. Further observe that we may use a stacked variant of both methods — that is, we may calculate different weighted sums and combine these in a weighted ranking.

Selection Strategies

We distinguish three selection strategies:

- **Top-k:** The most straightforward strategy is to simply select the top-k candidates according to the ranking. In this case, the parameter k predefines the size of a lexicon. As terms are incorporated into a lexicon disregarding their absolute score, the advantage of this strategy is that it can be applied independent of the chosen candidate ranking approach.
- **Fixed threshold:** In contrast, candidates may be selected by predefining a threshold value for the minimum score. Whereas in the case of the LRT-approach the score is directly interpretable (translates to a confidence level regarding the statistical test), other scores (e.g., diversity-score) are based on heuristics and thus are not directly interpretable. In such cases, it is difficult to define a reasonable threshold parameter for a minimum score. A test corpus or direct human supervision is needed to empirically find optimal values.
- **Dynamic threshold:** As a third strategy, we refer to Jakob et al. [185], who propose a method to dynamically calculate a threshold value. The method is derived from an algorithm for outlier detection (Wilcox [429, chap. 3]). The algorithm basically computes a linear combination of the mean and standard deviation of the LRT-score:

$$threshold_{wilcox} = \mu_{score} + \sigma_{score}, \tag{7.9}$$

  where μ_{score} refers to the mean of all the scores in a ranking and σ_{score} refers to the standard deviation of these scores. With regard to the LRT-approach, Jakob et al. [185] report improved results when employing the dynamically calculated threshold instead of a fixed threshold. The method is generally applicable, independent of the approach chosen for scoring the candidates.

7.5. Incorporating Weakly Labeled Data

Earlier we proposed to filter sentiment bearing pre-modifiers with stable prior polarity by means of a simple stop word list. But we also encounter pre-modifiers with target-specific polarity (e.g., "*long* battery life"). As the *GlossEx filter* (see Section 7.4.3) is a generic method, it cannot cope with this type of modification. For example, the pre-modifier "long" is very likely to have a strong domain relevance for digital camera reviews and is also very likely to exhibit a strong association with head nouns such as "battery" (e.g., "long battery life") or "shutter" (e.g., "long shutter lag time"). A generic filter most likely keeps the pre-modifier and produces invalid product aspects.

We propose to leverage user-provided pros/cons summaries, which typically accompany a customer review. We hypothesize that valid pre-modifiers (e.g., "digital" in "digital camera" or "wireless" in "wireless internet") occur similarly distributed with their head noun in both, lists of pros and lists of cons. For invalid pre-modifiers, that is, target-specific sentiment words, we assume that they occur (jointly with their head noun) either more often in lists of pros or lists of cons. In the following we design a *likelihood ratio test* to operationalize our hypothesis. We consider the probabilities

$$\begin{aligned} p_1 &= Pr(pm|head; pros) \\ p_2 &= Pr(pm|head; cons), \end{aligned}$$

where p_1 denotes the probability in a corpus of pros lists that pm occurs as pre-modifier, given that the head noun $head$ previously occurred, and p_2 refers to the same probability, but in a corpus of cons lists. To design a hypothesis test, we assume as null hypothesis H_0 that $p_1 = p = p_2$. In words, we assume that, independent of whether we consider the pros or cons corpus, the probability that pm co-occurs with $head$ is the same. The alternative hypothesis is that $p_1 \neq p_2$, that is, pm co-occurs with $head$ with higher probability either in the pros or cons.

Take note that we explicitly design a test that considers the strength of association between modifier and head instead of relying on the occurrence frequency of the whole phrase. During preliminary studies we designed a test that examined the difference in the probabilities of occurrence in either pros or cons documents. This test produced an unacceptable high rate of false positives. The reason is that some correct terms exhibit an unequally high probability of occurrence in either the pros or cons. The mention of such a term is inherently associated with a positive or negative assessment. For example, the term "chromatic aberration", which refers to a type of distortion, is generally expressed in a negative context and has thus a significantly higher probability to occur in a cons document. The test would erroneously identify "chromatic" as a sentiment expressing modifier, thus removing it from the candidate. Another example is the term "panoramic mode". As this term refers to a generally desired feature, reviewers tend to comment on it (or just name it) more often in pros documents. Again, the modifier (here: "panoramic") would be incorrectly removed. By comparing the strength of association of modifier and head instead of the frequency of occurrence, we overcome this problem.

As in Eq. (7.4), we calculate the likelihood ratio λ and utilize the value $-2 * log\lambda$ to reject H_0 at a desired confidence level (Appendix C.1.2 explains how we derive estimates for p_1, p_2, p and how we calculate λ). In case we can reject H_0, we remove the pre-modifier pm from the term candidate since we are confident that the combination occurs with higher probability either in the pros or cons. Otherwise, we keep the pre-modifier.

7.6. Experimental Setup and Evaluation Metrics

Aspect Detection Algorithms

The previous section described a terminology extraction pipeline that allows to automatically generate a lexicon of product aspects. However, *obtaining* a product aspect lexicon is only one part. For extrinsic evaluation, we also need to consider how to *apply* the dictionary — that is, how to to detect mentions of product aspects in natural language text. For our experiments we examined two different approaches, which differ mostly in the way linguistic information is incorporated. We do not go into more details here as we set focus on approaches to automatic lexicon construction, rather than examining different methods for lexicon application. We refer to Appendix C, which describes the algorithms in detail and reports results of a comparative experiment. In short, our findings are that the linguistically more informed Algorithm C.2 shows generally better results for aspect extraction than Algorithm C.1, which basically performs simple string matching.

Baseline Approach

To measure the effectiveness of the different components and approaches, we establish a baseline for comparison. We implement the "feature term extraction" component of the "Sentiment Analyzer" system as described by Yi et al. [452]. We focus on the specific configuration for which they reported the best results[21]. In particular their best feature term extraction configuration is as follows:

1. Acquire candidate terms with the **BNP1** part-of-speech tag filter and application of the **bBNP** heuristic.

[21] Yi et al. [452] perform an intrinsic evaluation and only report results for the precision of the different configurations.

2. The original system does not apply any filtering, but we apply product name filtering.
3. The original system does not apply any variant aggregation.
4. Use **open counting**[22] to aggregate frequency statistics in C_+ and C_-.
5. Rank candidates with the **LRT-score**.
6. Select the candidates with a minimum score of 3.84[23].
7. Apply Algorithm C.1 (the less linguistically informed approach) for aspect extraction from natural language text.

Jakob [182] provides a detailed comparison of the frequent itemset mining approach by Hu and Liu [177] and the feature term extraction by Yi et al. [452]. The basic findings are that Yi's method consistently outperforms the frequent itemset mining approach on different datasets. It is thus reasonable to choose Yi's method as a baseline for our experiments. We report detailed results for the baseline method in Appendix C.3.

Definition of Fixed Parameter Values

Most of the extraction pipeline's components are parameterized with different threshold values. We fix the subset of parameters shown in Table 7.4, as they are not subject to further experimentation. To find reasonable values for these parameters, we optimize them on a development set that consists of customer reviews on mp3 players[24].

parameter	value	component	description
δ_{lower}	1.5	GlossEx filter	lower threshold for relative frequency ratio
δ_{upper}	3.0	GlossEx filter	upper threshold for relative frequency ratio
α_{lower}	0.005	GlossEx filter	lower threshold for pre-modifier/head cohesion
α_{upper}	0.3	GlossEx filter	upper threshold for pre-modifier/head cohesion
$\alpha_{intermediate}$	0.02	GlossEx filter	intermediate threshold for pre-modifier/head cohesion
γ	3	misspelling aggregation	minimum length of candidate in characters
ϵ	8	misspelling aggregation	characters per unit of edit cost
σ	5	compositional var. agg.	minimum absolute support
ρ	0.1	GDC-score	reduction factor for single word terms

Table 7.4.: Definition of parameter values that are not subject to variation.

Intrinsic Evaluation (Accuracy of the Generated Lexicons)

With regard to evaluating NLP components, such as our terminology extraction process, we can basically distinguish *intrinsic* and *extrinsic evaluation*. Intrinsic evaluation refers to the task of assessing the component in isolation, whereas extrinsic evaluation (or *evaluation in use*) assesses the component as part of a more complex system. In our case, intrinsic evaluation refers to assessing the quality of

[22] At this point the description in [452] is inconclusive. It is unclear whether filtered or open counting is applied to aggregate statistics. We assume that they use open counting.
[23] 3.84 is the critical value of the χ^2-distribution for one degree of freedom at a confidence level of 95%.
[24] We intentionally choose a domain differing from our test sets (hotel and digital camera reviews) to increase the generalizability of our experimental results.

the generated lexicon of product aspects. We assess the quality by examining the *precision* of the process. Traditionally the precision is calculated as the fraction of the results that are correctly returned by a system. But, since the extraction process generates a ranked list of aspects, it is also reasonable to evaluate the results with a metric known as *precision@n*[25] (P@n). Given a maximum (cut-off) rank n, P@n describes the precision value calculated over the first n results returned by the system. We decide on the correctness of the results by manually inspecting the generated lexicon. For each entry, a human supervisor decides whether it represents a valid product aspect in the target domain or not. An entry is marked as a *true positive* if it satisfies the rules we have described in the annotation guidelines for sentiment targets and product aspect mentions in Appendix A.3. Otherwise it is marked as *false positive*.

Extrinsic Evaluation (Accuracy of Aspect Extraction)

For extrinsic evaluation, we apply the generated lexicon to extract product aspects from the free text part customer reviews. We assess the aspect extraction task by using the expression level annotations of our hotel and digital camera datasets as a gold standard. Extrinsic evaluation is fully automated. We apply and report the standard evaluation metrics used to assess information extraction systems, namely *precision*, *recall*, and the *f-measure*[26] (the harmonic mean of precision and recall). Depending on the evaluation goal, we consider four different evaluation scenarios[27]:

- **Scenario A** — Extraction of product aspects. In this scenario, we evaluate how well a lexicon-based approach performs in identifying product aspects in customer reviews. That is, we do not consider the presence of sentiment expressions at all. Recall that our expression level annotation scheme allows to differentiate between sentiment targets and uncommented mentions of product aspects. For this task, we define the union of sentiment target and aspect mention annotations as gold standard. Thus, any extraction that matches either a sentiment target annotation or an aspect mention annotation is considered a true positive.

- **Scenario B1** — Extraction of sentiment targets when polar sentences are identified with perfect accuracy. In this scenario, we restrict valid matches to sentiment targets only. The extraction algorithm has access to the gold standard to determine whether a sentence is polar or not. The algorithm uses the expression level annotations and regards a sentence as polar, if at least one sentiment expression annotation is present. Only lexicon matches that occur in a polar sentence are extracted.

- **Scenario B2** — Extraction of sentiment targets when individual sentiment expressions are identified with perfect accuracy (without perfect target association). Again, the extraction algorithm has access to the gold standard sentiment expression annotations. For each sentiment expression *SE*, the algorithm extracts the lexicon match in the same sentence which is closest (fewest number of interjacent tokens) to *SE*. In case of a draw, the match with the higher score is extracted.

- **Scenario B3** — Extraction of sentiment targets when individual sentiment expressions are identified with perfect accuracy (with perfect target association). The algorithm can perfectly identify sentiment expressions and corresponding targets. It extracts each lexicon match that overlaps a sentiment target.

Scenario *A* examines lexicon-based product aspect extraction independent from further sentiment analysis. Thus, the scenario is comparable to a general terminology extraction task. Scenarios *B1*, *B2*,

[25] cf., Buckley and Voorhees [58]
[26] cf., Manning et al. [250, chap. 8]
[27] Jakob [182] examines and motivates similar evaluation scenarios.

and *B3* examine how well a lexicon-based approach performs with regard to identifying sentiment targets.

In this chapter, we primarily aim at evaluating the accuracy of aspect lexicon generation and lexicon-based extraction. For now, it is not our goal to assess the accuracy of a sentiment expression detection and association task. We therefore provide the extraction algorithm in scenarios *B1-B3* with perfect (gold standard) knowledge on the presence of sentiment expressions and, in case of *B3*, also with perfect information on the association of sentiment expressions and targets.

These *synthetic scenarios* serve as upper bounds for real-world lexicon-based sentiment target extraction. Scenario *B1* resembles a task where a sentence has been priorly classified as subjective. Scenario *B2* resembles a task where individual sentiment expressions have been identified (e.g., with a sentiment lexicon) and the association between targets is heuristically determined. We examine scenario *B3* to measure the influence of the algorithm used to associate sentiment expressions to targets and to provide an upper bound for such algorithms.

Type of product aspect mention

Our expression level annotation scheme distinguishes four different types of product aspect mentions (*nominal*, *named*, *pronominal*, and *implicit*). As our terminology extraction process is designed for finding nominal mentions of product aspects, we restrict the earlier mentioned scenarios accordingly. That is, for aspect mention annotations the
"isProductName"-flag must not be true and for sentiment target annotations none of the flags "isImplicit", "isProductName", or "isExophoric" must be true, nor must the attribute "pronounReferenceID" be set. Evaluation of the baseline method compares results for both, only-nominal and all mention types. For the other experiments, we concentrate on nominal extractions only.

Strict or lenient metric

We distinguish whether an extraction either exactly or partially matches the span of an annotation in the gold standard. The *strict metric* only regards exact matches as true positives and considers all other extractions as false positives. The *lenient metric* also allows for partial matches. In particular, a partial match is accepted as true positive, if the extraction e and the gold standard annotation g have at least one token in common. More formally, let the subscripts *begin* and *end* define the span boundaries of an annotation in a document. If the predicate $(e_{begin} \leq g_{end} \ \& \ e_{end} \geq g_{begin})$ holds, e is considered a true positive, otherwise it is a false positive. For example, consider that in the gold standard we annotate the aspect "intelligent auto mode". The strict metric requires to match exactly this phrase. The lenient metric also allows "sense-preserving" matches, such as "auto mode" or "mode". However, also less meaningful matches, such as "intelligent auto" or "auto", are considered correct with the lenient metric.

Statistical Significance

As part of the evaluation, we compare different configurations against each other. If applicable, we report the statistical significance of resulting differences in the evaluation measures. If not otherwise stated, we employ a *paired two-tailed t-test* at a significance level of 99% (p-value < 0.01). Statistically significant differences are indicated by using the common **-notation.

Corpora

- **Development set**: To determine the parameters listed in Table 7.4, we use a collection of 53,183 customer reviews on mp3 players (8,211,641 tokens) extracted from the websites Amazon.com,

Epinions.com, and Buzzillions.com.

- **Foreground corpora**: We evaluate our system on the hotel and digital camera review datasets that we described in Chapter 5. In particular, we use the web crawls of 417,170 hotel reviews and 180,911 digital camera reviews. From these, we randomly extract subsets of 10, 20, ..., 100, 200, ..., 1,000, 1,500, 2,000, ..., 5,000, 7,500, 10,000, ..., and 50,000 documents, where each larger set is a superset of the smaller sets. For the baseline setting, we choose the set of 20,000 hotel (digital camera) review documents, containing 212,183 (180,846) sentences and 3,792,588 (3,260,502) tokens.
- **Background corpus**: For approaches that require a background corpus, we use a small subset of the freely available "ukWaC corpus" [30] (2 billion words extracted from English language websites). From the original 2.7 million documents in the corpus, we compiled a subset of 100,000 randomly selected documents, which contain 3,284,560 sentences and 83,778,171 tokens. We choose this particular corpus based on the intuition that a general web corpus is a better contrast to specific customer reviews than for example the "British National Corpus" [46], which contains primarily more formal documents, such as newspaper articles, journals, or books.
- **Pros/cons corpora**: For parts of the experiments, we require weakly labeled data in form of pros and cons documents. For the hotel domain, we use the pros and cons parts of customer reviews which we extracted from the website Priceline.com. Out of a collection more than 500,000 reviews, we sample a subset of 100,000 reviews where pros and cons parts are both not empty. With regard to the digital camera domain, we collected a set of 139,417 customer reviews from the websites Buzzillions.com and Reevoo.com. Here, we sample a (smaller) subset of 50,000 documents for pros and cons each.
- **Test corpora**: The final results of the extrinsic evaluation are reported based on the gold standard annotations of the expression level hotel and digital camera corpora.

Unless otherwise stated, the experiments in Section 7.7 are based on the 20,000 document subsets of the foreground corpora, as well as on the complete 100,000 document background corpus.

7.7. Experiments and Results

In this section, we discuss the main results that we obtained with our terminology extraction approach to constructing a product aspect lexicon. In particular, we examine the influence of different *candidate filtering* techniques (Section 7.7.1), the effects of different *variant aggregation* approaches (Section 7.7.2), the influence of varying *candidate acquisition* methods (Section 7.7.3), and the effectiveness of the proposed *ranking metrics* (Section 7.7.4). We further experiment with varying corpus and lexicon sizes (Sections 7.7.5 and 7.7.6) and evaluate our indirect crowdsourcing method for more effective pre-modifier filtering (Section 7.7.7).

7.7.1. Influence of Candidate Filtering Techniques

We measure the effectiveness of the different filtering techniques in isolation and in combination. In particular, we report intrinsic evaluation results for the *review stop word filter*, the *sentiment pre-modifier filter*, the *measuring unit filter*, the *GlossEx pre-modifier filter*, as well as for the combination of all filters. With regard to extrinsic evaluation, we only report results for the joint application of all filters.

As reference, we use the baseline system in combination with aspect detection Algorithm C.2. By adding a particular filter to the baseline system, we can measure its influence in isolation. To determine the contrastive relevance of a pre-modifier pm in the *GlossEx filter*, we use the unigram statistics of pm in the complete background corpus (i.e., 100,000 documents). We do not lemmatize the

pre-modifier, but count its inflectional form. Further, we only count unigrams that match the part-of-speech of the pre-modifier (for shallow word sense disambiguation). To compute the association between pre-modifier and head noun, we use the unigram and bigram statistics of the foreground corpus. The statistics with regard to the head noun are based on the lemmatized form of the head.

Intrinsic Evaluation Figure 7.2 shows the results for intrinsic evaluation of both datasets. When applying all filters jointly, the resulting lexicons consist of 975 entries for the hotel dataset (baseline: 1182 entries) and 767 terms with respect to the digital camera dataset (baseline: 953 entries). Compared to the baseline, the precision of the complete lexicon is around 10 percentage points higher with regard to the hotel corpus (all filters: 0.707, baseline: 0.612) and approximately 14 percentage points when considering the camera corpus (all filters: 0.817, baseline: 0.676). Precision@40 increases from 0.625 to 0.675 (hotel) and 0.800 to 0.875 (camera). The figure shows that each filter has a positive effect on the precision and that for both datasets, the *GlossEx filter* has the greatest influence. The other filters' effects are more dependent on the application domain. For the digital camera dataset, we find that review stop word filtering and measurement unit filtering are nearly equally important. For the hotel corpus, measurement unit filtering only leads to a marginal improvement as numerical pre-modifiers are less common in this corpus.

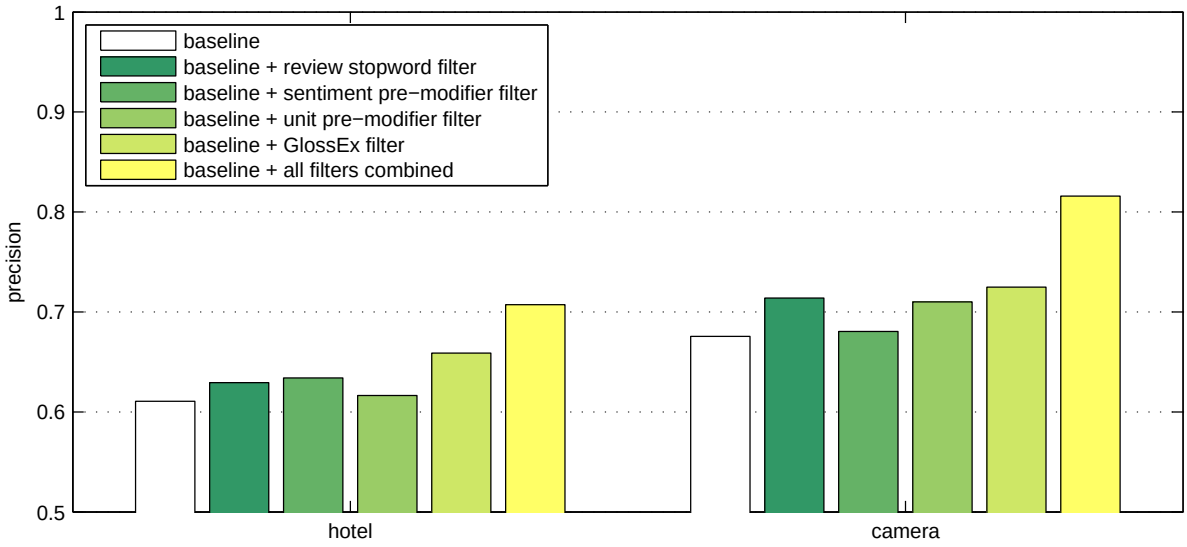


Figure 7.2.: Intrinsic evaluation of the candidate filter approaches. The bar chart shows the proportion of correctly extracted terms (precision).

Extrinsic Evaluation In Table 7.5 we present the results of the extrinsic evaluation for both datasets. We find that the higher precision of the generated lexicons consistently leads to better results for the product aspect and sentiment target detection tasks. All reported improvements are statistically significant (comparison to Table C.2). The maximal gain in f-measure is around 4 percentage points and is achieved in scenario B2 for both datasets. The minimal improvement with regard to the camera dataset is around 3 percentage points compared to 2 percentage points for the hotel corpus. The table also shows that not only a higher precision is responsible for the improved f-measure, but also higher recall values: Except for the review stop word filter, all other filters prune false pre-modifiers of different kinds in term candidates. Partial matches caused by such false pre-modifiers are reduced, which results in less false negatives. Indeed, a mistake analysis of false positives and false negatives in scenarios A and B3 reveals that after filtering only very few errors (< 0.5%) can be accounted to the inclusion of false pre-modifiers of the types detectable by the filters.

		nominal mentions		
dataset	**scenario**	precision	recall	f-measure
Hotel	A	0.569 (0.018**)	0.752 (0.022**)	0.648 (0.020**)
Hotel	B1	0.461 (0.027**)	0.751 (0.039**)	0.571 (0.032**)
Hotel	B2	0.707 (0.048**)	0.672 (0.042**)	0.689 (0.045**)
Hotel	B3	0.857 (0.044**)	0.751 (0.039**)	0.800 (0.041**)
Camera	A	0.692 (0.042**)	0.743 (0.018**)	0.717 (0.031**)
Camera	B1	0.470 (0.027**)	0.722 (0.023**)	0.569 (0.027**)
Camera	B2	0.680 (0.042**)	0.649 (0.032**)	0.664 (0.037**)
Camera	B3	0.793 (0.025**)	0.722 (0.023**)	0.756 (0.024**)

Table 7.5.: Results for product aspect and sentiment target extraction when jointly applying all filter techniques. Reported results are based on the strict evaluation measure.

7.7.2. Influence of Variant Aggregation Techniques

In this section, we examine the influence of the different variant aggregation techniques. We only evaluate extrinsically as we primarily expect the lexicons to achieve a better coverage due to the added variants. As a baseline, we use the results from the previous section (all filters activated) and compare it to a system that additionally incorporates all variant aggregation techniques into the lexicon extraction pipeline. The misspelling variant detection as well as the compositional variant detection are configured with the parameters defined in Table 7.4.

Table 7.6 shows the results for the joint application of the *symbolic*, *compounding*, *misspelling*, and *compositional* variant aggregation approaches[28]. The obtained results do not suffice to confirm our initial hypothesis that variant aggregation improves lexicon coverage and thus would lead to measurable higher recall values. Although we can measure improved results with activated variant aggregation for the digital camera corpus, the differences are relatively marginal (at maximum 1.1 percentage points for the f-measure) and are not statistically significant at the chosen 99% confidence level. Regarding the hotel corpus, the influence of variant aggregation is even lower.

		nominal mentions		
dataset	**scenario**	precision	recall	f-measure
Hotel	A	0.567 (-0.002)	0.751 (-0.001)	0.646 (-0.002)
Hotel	B1	0.461 (0.000)	0.751 (0.000)	0.571 (0.000)
Hotel	B2	0.706 (-0.000)	0.674 (0.002)	0.690 (0.001)
Hotel	B3	0.855 (-0.002)	0.751 (0.000)	0.799 (-0.001)
Camera	A	0.698 (0.006)	0.748 (0.004)	0.722 (0.005)
Camera	B1	0.477 (0.007)	0.730 (0.008)	0.577 (0.008)
Camera	B2	0.686 (0.006)	0.654 (0.005)	0.669 (0.005)
Camera	B3	0.807 (0.014)	0.730 (0.008)	0.767 (0.011)

Table 7.6.: Results for product aspect and sentiment target extraction with all filters and all variant aggregation approaches. Reported results are based on the strict evaluation measure.

Mistake Analysis To understand why the variant aggregation steps do not fulfill the anticipated gain in recall, we perform a mistake analysis of the false negatives. In particular, we examine the

[28] Take note that we do not apply inflectional variant aggregation. When using part-of-speech pattern BNP1, we lemmatize all words during aspect detection, which in effect is another form of inflectional variant aggregation.

results produced in the setting with scenario B3 and strict evaluation. For this setting, we compare the false negatives with and without variant aggregation. Concerning the hotel corpus, we find 18 out of 251 false negatives (7.2%) which are candidates for variant aggregation — for instance, "conciergelounge" (compound aggregation), "continetal breakfast" (misspelling aggregation), or "layout of the room" (compositional aggregation). In the ideal case, that is when variant aggregation successfully adds all the candidates to the lexicon, a maximum gain of 1.8 percentage points for recall can be achieved (778 instead of 760 true positives). Concerning the digital camera corpus, we identify 21 out of 242 false negatives (8.7%) as candidates, constraining the maximum gain in recall to 2.4 percentage points (646 instead of 625 true positives). Thus, a first observation is that, even in the ideal case, the expectable increase of recall is relatively low (< 2.5 percentage points). But our results vary widely from the ideal case:

For the hotel corpus only one of the candidates ("conciergelounge") is successfully added by variant aggregation (gain of 0.1 percentage points). For the digital camera corpus only 8 of 21 candidates are additionally detected (gain of 0.9 percentage points). The major reason for these results is again rooted in *Zipf's law*. Our variant aggregation approaches could easily associate false negatives such as "continetal breakfast" or "dtorage capacity" to their correct canonical forms, but the misspellings simply do not occur at all in the 20,000 documents foreground corpora we use for generating the lexicons. As our algorithm can only consider misspellings which are seen in the foreground corpus, we cannot identify these variants. The same reason also prevents the detection of other variant types, e.g., compositional variants such as "layout of the room" (the canonical form "room layout" does not occur) or "feel of this camera".

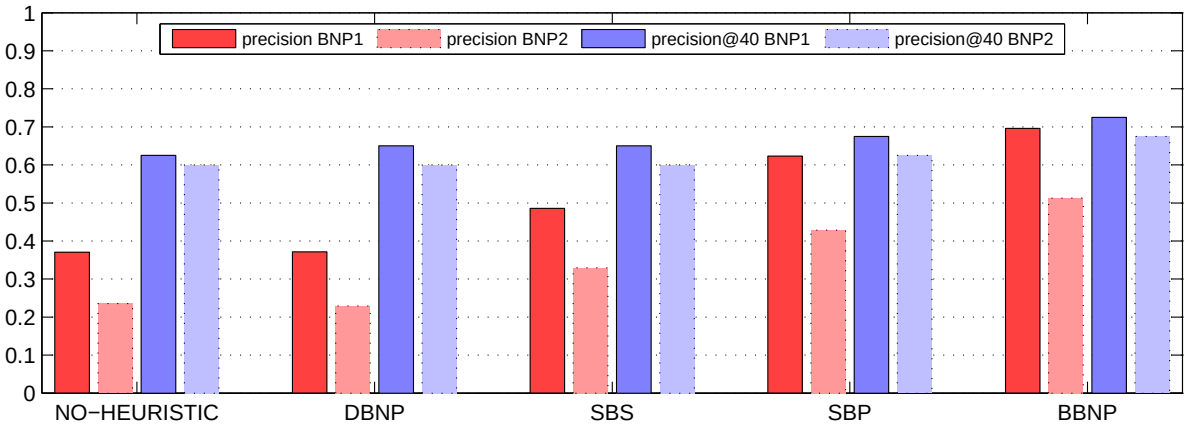
A further fact which affects the influence of the variant aggregation steps is the following: Some misspelling, compound, or symbolic variants occur by themselves so frequent in the foreground corpus, that the LRT-ranking method adds them as entries to the generated lexicon. That is, independent of whether variant aggregation is applied or not, some of the most frequent variants are part of the generated lexicon. Consequently, this further lowers the potential influence of the aggregation step on the achievable recall. However, when considering the manual post-processing of automatically extracted lexicons (refer to Section 7.7.8), variant aggregation helps to lower the manual effort significantly.

7.7.3. Influence of Candidate Acquisition Patterns and Heuristics

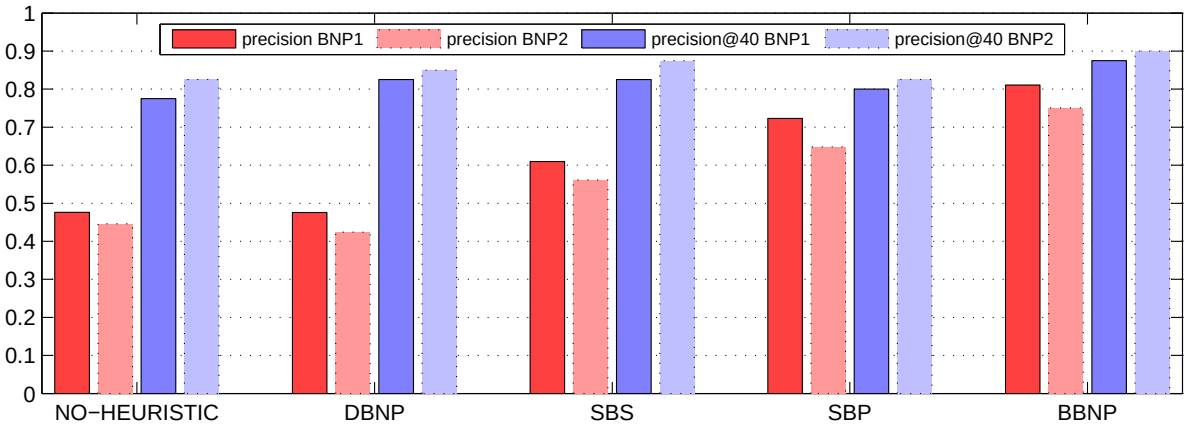
The goal of this section is to study the influence of the different part-of-speech tag filters and domain specific heuristics for initial candidate acquisition. We assess the different combinations of patterns and heuristics by means of intrinsic and extrinsic evaluation. In particular, we examine the two different part-of-speech tag filters (*BNP1* and *BNP2*) and the four different acquisition heuristics (*dBNP*, *bBNP*, *SBS*, and *SBP*). We additionally consider the case when no heuristic is applied, resulting in 10 different configurations for each of the two datasets. Reported results for each configuration refer to the application of the baseline method with all filters and variant aggregation approaches activated. For extrinsic evaluation we use Algorithm C.2.

Intrinsic Evaluation Figure 7.3 shows the results of the intrinsic evaluation for the hotel and digital camera datasets. We report the precision of the complete lexicons (red bars) and the precision@40 (blue bars). The line style of the bars indicates the part-of-speech pattern of a configuration, which is either BNP1 (solid line) or BNP2 (dotted line).

For both datasets, the overall precision with part-of-speech pattern BNP2 is significantly lower than with the pattern BNP1. This is expectable as the BNP2 pattern is less restrictive — it allows for the inclusion of arbitrary long candidates and also proper nouns. Especially regarding the hotel corpus, the inclusion of proper nouns is counterproductive in terms of lexicon precision. The lexicon generation process extracts many named entities such as locations (e.g., "New York City", "Union Square",



(a) hotel corpus



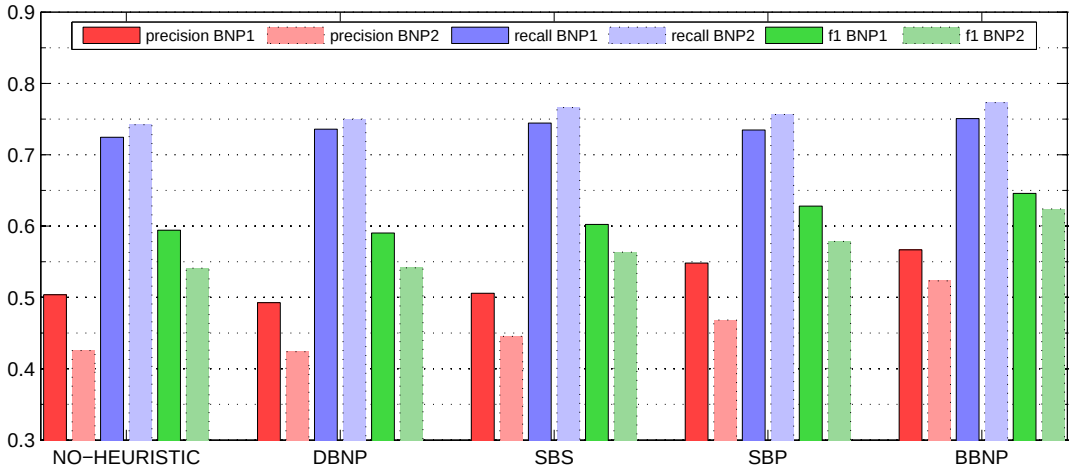
(b) digital camera corpus

Figure 7.3.: Intrinsic evaluation results with different acquisition patterns and heuristics.

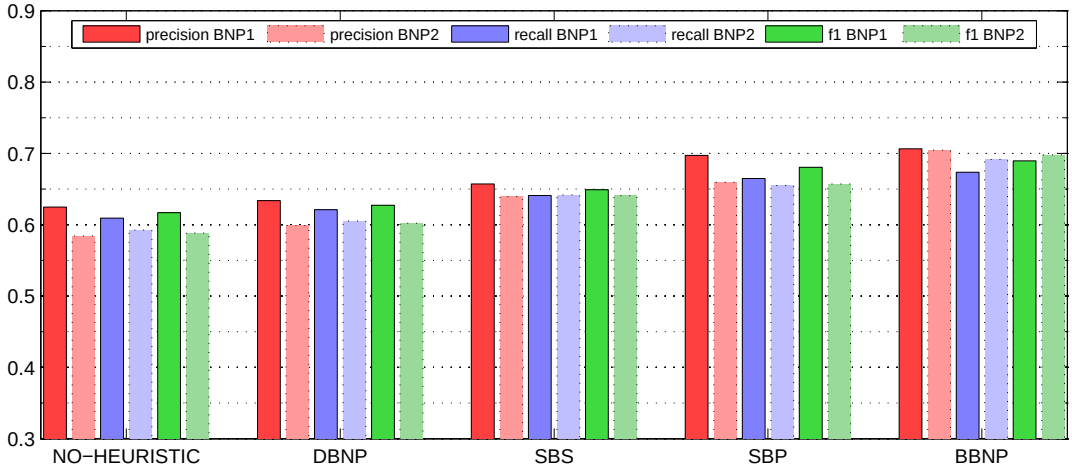
or "LA") or sights (e.g., "National Mall", "Rockefeller Center", or "Lincoln Memorial"), which naturally occur significantly more frequent in a corpus of hotel reviews than in a standard web corpus. Compared to the configurations with the BNP1 pattern, we achieve precision values which are consistently at least 15 percentage points lower in the hotel domain. The difference is less pronounced when considering the digital camera dataset. Here, we observe precision values which are about 5 percentage points lower. The problem with the inclusion of named entities is less severe in the camera domain.

Also, as expected, we find that the overall precision increases with the restrictiveness of the applied candidate acquisition heuristic. For the hotel dataset (and BNP1 pattern), the lexicon precision improves from 0.37 with no heuristic applied to 0.70 with the bBNP heuristic. With respect to the digital camera corpus, the precision increases from 0.48 to 0.81. The results show that the SBP heuristic does not provide any improvement in terms of lexicon precision over the bBNP heuristic for both datasets, but it is apparent that both heuristics achieve a consistently high precision of the complete lexicon: The measured results for the 40 highest ranked entries do not differ widely from the overall precision (at maximum 7.5 percentage points). For the less restrictive heuristics these values differ at minimum 16.5 percentage points (hotel: SBS) and at maximum 35 percentage points (camera: dBNP). We find that the precision of the highest ranked lexicon entries is relatively independent of the applied heuristic. For these entries the ranking algorithm alone provides reasonably good results.

Extrinsic Evaluation For extrinsic assessment we consider the evaluation scenarios A and B2[29]. Figure 7.4 shows the results obtained with the hotel dataset and Fig. 7.5 depicts the results for the digital camera corpus. The different colors of the bars indicate the specific evaluation measure: red bars refer to the precision, blue bars to the recall value, and green bars to the f-measure. Again, the line style indicates the applied part-of-speech pattern. Take note that the limit of the y-axes ranges from 0.3 to 0.9 in all figures.



(a) evaluation scenario A



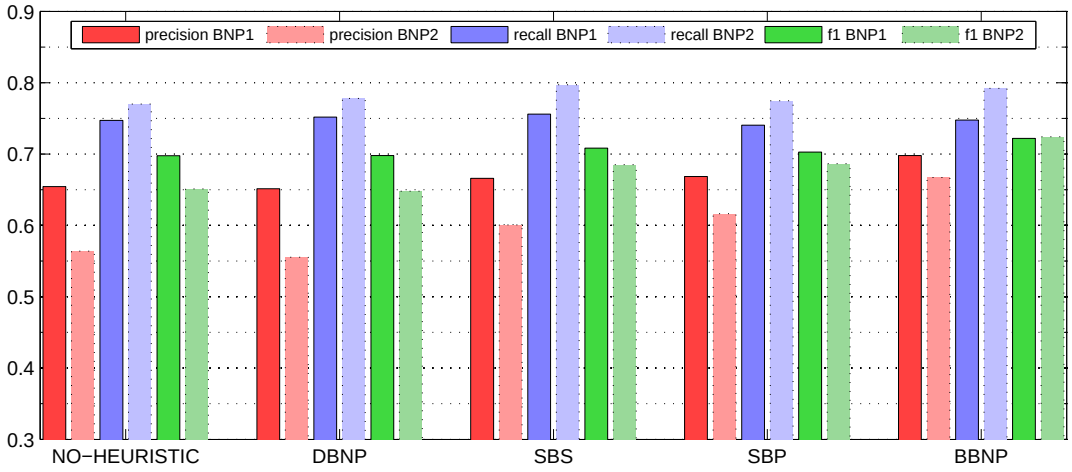
(b) evaluation scenario B2

Figure 7.4.: Hotel corpus: Extrinsic evaluation results with different acquisition patterns and heuristics for scenarios A and B2.

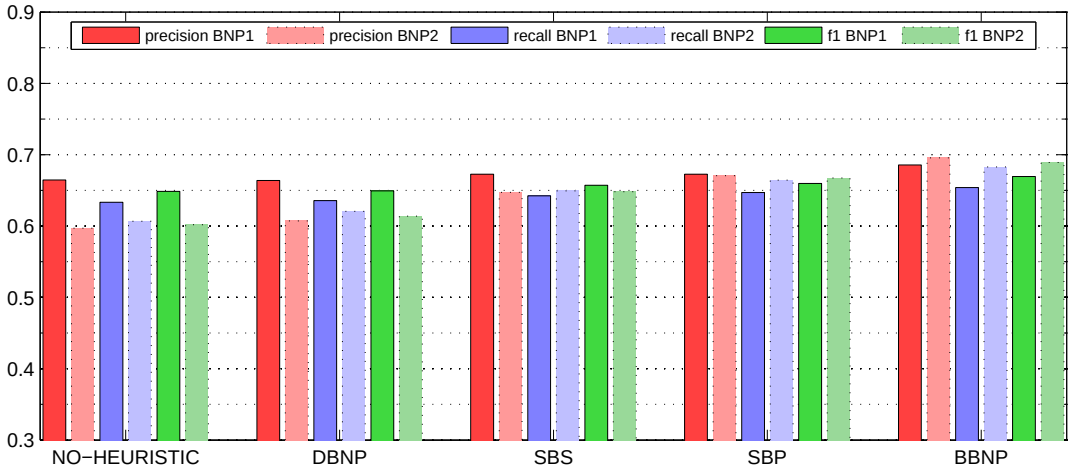
The main observation is that the influence of the applied acquisition patterns and heuristics is only moderately high. For evaluation scenario A we find a maximum difference in f-measure of 9.0 percentage points (hotel dataset, BNP1-bBNP vs. BNP2-dBNP) and for scenario B2 the maximum difference is 8.9 percentage points (hotel dataset, BNP2-bBNP vs. BNP2-no-heuristic). In both scenarios and datasets, the f-measure improves with the restrictiveness of the applied heuristics.

We further find that the bBNP heuristic consistently outperforms the SBP heuristic in all configurations and with regard to each evaluation measure (precision, recall, and f-measure). The intrinsic

[29] We do not report results for all evaluation scenarios. However, scenarios A and B2 cover the aspect extraction task and the sentiment extraction task. The tendency of the results for scenarios B1 and B3 are similar to B2 and do not provide further insight.



(a) evaluation scenario A



(b) evaluation scenario B2

Figure 7.5.: Camera corpus: Extrinsic evaluation results with different acquisition patterns and heuristics for scenarios A and B2.

evaluation already showed that the resulting lexicon with the SBP heuristic exhibits a lower precision than with the bBNP heuristic. This is despite the fact that the SBP heuristic is even more restrictive and generates a smaller lexicon (hotel-BNP1: 743 vs. 901 entries, camera-BNP1: 585 vs. 724 entries). The precision with the SBP heuristic is in average about 3 percentage points lower than with the bBNP heuristic.

Although the part-of-speech pattern BNP2 leads to a lower precision of the generated lexicons (compared to pattern BNP1), we achieve better results for both corpora when considering the best heuristic (bBNP) and the sentiment target detection task (scenario B2). Regarding the hotel corpus, the f-measure is 1.3 percentage points higher and for the camera dataset the increase is 2.2 percentage points. This increase is mainly due to the higher recall achieved with the less restrictive pattern (leading also to a slightly higher precision as some previously partial matches are matched correctly). In most other configurations we also observe a higher recall with pattern BNP2, but typically a significantly lower precision. From the intrinsic and extrinsic evaluation we can conclude that the bBNP heuristic consistently achieves the best results. Depending on the goal (high lexicon precision or high f-measure for sentiment target detection) either pattern BNP1 or pattern BNP2 is preferable. Considering that manual revision of automatically generated lexicons guarantees high lexicon accuracy, in

that case, applying pattern BNP2 is indicated as it generally promises higher recall values.

7.7.4. Comparison of Ranking Measures

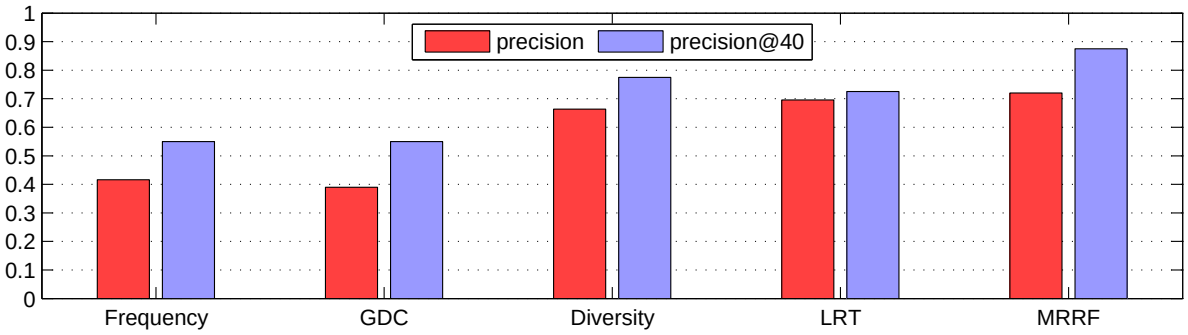
In this section, we evaluate the influence of the different ranking algorithms introduced in Section 7.4.6. In particular, we consider the *relative frequency ratio* (MRRF), the *likelihood ratio test* (LRT), the *generalized Dice coefficient* (GDC), and the *diversity value* (Diversity). We further report results for rankings based on the *raw frequency* alone, which we regard as a baseline for the following experiments. We examine the results for each algorithm in isolation and additionally consider reasonable combinations of the algorithms. All combinations are based on the weighted rank approach. To rule out the influence of varying lexicon sizes generated by the different algorithms, we choose a fixed size for each dataset. We set the maximum lexicon size to the size of the lexicon generated by the LRT-ranking with a minimum LRT-score of 3.84. For larger lexicons we prune the entries with the lowest scores. In each configuration we apply all filter and variant aggregation approaches and for extrinsic evaluation we utilize the Algorithm C.2.

Intrinsic Evaluation Figure 7.6 presents the results of intrinsic evaluation when the different algorithms are applied in isolation. As before, we report results for the overall precision of the generated lexicons (red bars) and for the precision of the top 40 entries (blue bars). For both datasets we find that the algorithms Diversity, LRT, and MRRF outperform the baseline by far. Regarding the hotel corpus, the raw frequency ranking achieves a precision of only 0.42, whereas for the three named algorithms the precision values are 0.66, 0.70, and 0.72. For the camera dataset the precision values are 0.77, 0.81, and 0.81 compared to 0.45. The precision of the lexicon generated by the GDC-ranking is even lower than the baseline with 0.39 (hotel) and 0.43 (camera).

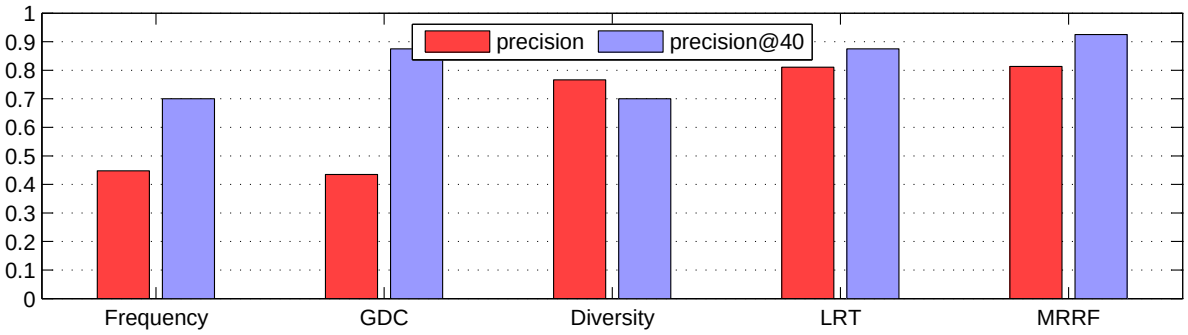
Comparing both contrastive term relevance measures, we observe that the MRRF-approach consistently achieves slightly better results than the LRT-approach. on both datasets The precision@40 improves to 0.88 (hotel) and 0.93 (camera), respectively. Lexicons generated by the Diversity algorithm exhibit an overall precision that is about 5 percentage points lower than the best results for both corpora. But nevertheless, they are significantly higher than the baseline and the GDC-approach, which also do not incorporate a background corpus.

As a next step, we examine whether the combination of different ranking algorithms leads to improved results. The hypothesis is that combining algorithms based on different definitions of term relevance is beneficial. We study combinations of contrastive and term cohesion measures (LRT+GDC, MRRF+GDC), contrastive and intra domain measures (LRT+Diversity, MRRF+Diversity), both contrastive measures (MRRF+LRT), combinations of all three types of measures (LRT+GDC+Diversity), the three measures with highest precision in isolation (MRRF+LRT+Diversity), as well as all five approaches in combination. In addition, we combine the MRRF-approach with the ranking based on raw frequency. We hypothesize that this combination is reasonable, as the MRRF-score alone does not account very well for frequency: two terms that exhibit the same relative frequency ratios get the same score, albeit one term may occur thousand times more often than the other. In this particular combination, we give a weight of $\omega_1 = {}^2/_3$ to the MRRF-ranking and a weight of $\omega_2 = {}^1/_3$ to the raw frequency ranking. Also in the combinations MRRF+GDC and LRT+GDC, we give the contrastive relevance rankings a higher weight (${}^2/_3$). For all other combinations, the weights are equally distributed.

Figure 7.7 shows the results of jointly ranking candidates with multiple algorithms. The main observation is that no combination achieves a significantly higher overall precision than the best algorithm in isolation (MRRF). The best combination (MRRF+Diversity) only marginally improves precision (hotel: +0.1 percentage points, camera: +0.5 percentage points), but is not statistically significant. Although achieving relatively high values for precision@40 (hotel: 0.8, camera: 0.98), the



(a) hotel corpus



(b) digital camera corpus

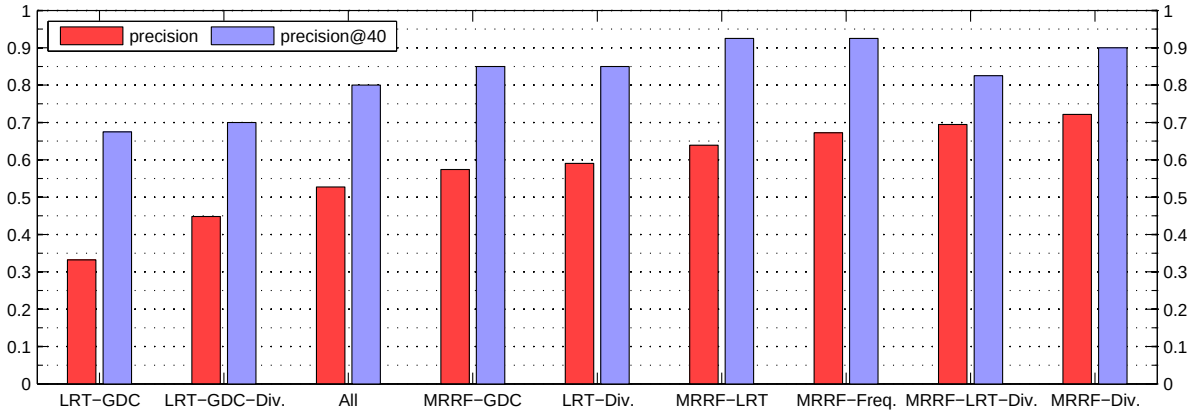
Figure 7.6.: Intrinsic evaluation results with different ranking algorithms.

combination of all rankings leads to only mediocre results in terms of overall precision of the generated lexicons. It can also be observed that a combination (LRT+GDC) may lead to even worse results than one of the rankings (GDC) produces in isolation.

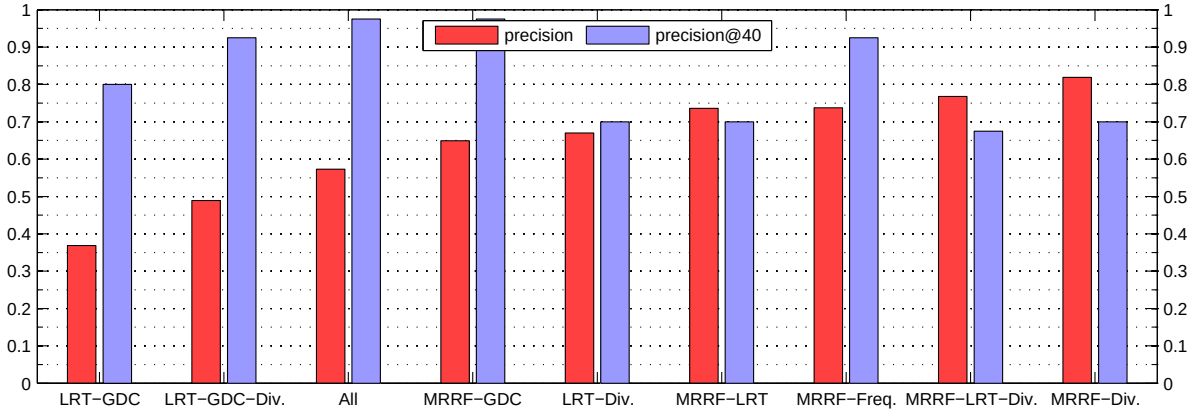
Extrinsic Evaluation For extrinsic evaluation, we consider the same combinations as introduced previously, except that, for reasons of clarity, we do not show results for the combinations LRT+GDC and LRT+GDC+Diversity (they perform worse than the baseline with raw frequency). As in the preceding section, we only report extrinsic evaluation results for scenarios A and B2. The results of the extrinsic evaluation on the hotel dataset are depicted in Fig. 7.8, whereas Fig. 7.9 shows the results regarding the digital camera dataset.

Also with respect to extrinsic evaluation, the main observation is that no algorithm or combination clearly outperforms all other approaches. The best results in terms of f-measure are achieved by the LRT and MRRF-approaches, as well as the combination MRRF+Diversity. We can conclude that contrastive measures generally lead to better results than the term cohesion or intra domain measures on our evaluation datasets. Both contrastive approaches exhibit a similarly high f-measure. Whereas, the MRRF-approach shows a slightly higher precision, the LRT-approach achieves a minimally higher recall.

Naturally, the differences in the results are more pronounced in scenario A as no gold standard knowledge is incorporated (in contrast to the synthetic scenarios B1-B3). In this scenario, each algorithm or combination improves on the baseline with raw frequency. The maximum difference in f-measure is 7.0 percentage points for the hotel corpus and 11.5 percentage points for the camera corpus. We find that the increased f-measure is solely due to a higher precision of the advanced approaches. Expectedly, the baseline exhibits a high recall value. However, its precision is relatively low. For evaluation scenario A, the precision is 14.9 (hotel) and 22.3 (camera) percentage points lower than the best results. The differences in scenario B2 are 4.6 (hotel) and 6.9 (camera) percentage points.



(a) hotel corpus



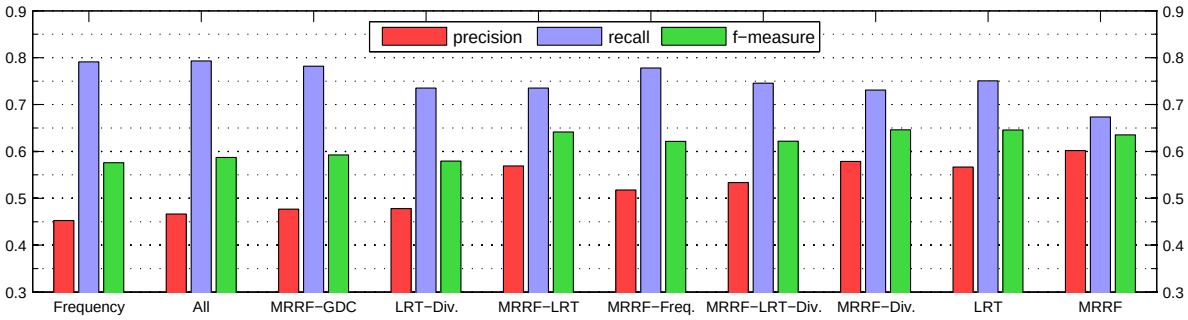
(b) digital camera corpus

Figure 7.7.: Intrinsic evaluation results with different combinations of ranking algorithms.

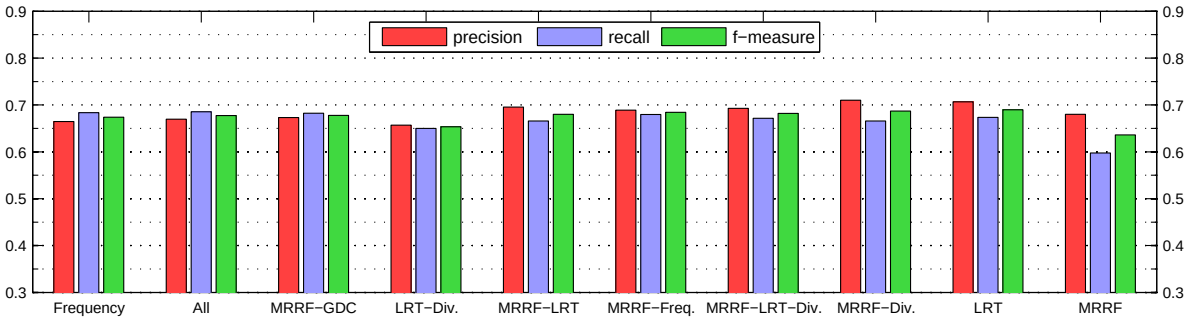
With regard to scenario B2 we cannot observe major differences in f-measure between the baseline and the other approaches. For the hotel dataset the baseline achieves an f-measure of 0.67, whereas the best approach exhibits an f-measure of 0.69. Concerning the camera dataset, the difference is 4 percentage points (0.63 compared to 0.67). In scenarios B1 and B3 (not shown in the figures), we achieve maximum differences of 6.9 and 0.8 percentage points, respectively. Considering these results, we can mainly confirm the following quite obvious assertions: The more accurate the information on sentiment expressions, the less important is the concrete algorithm to detect product aspects. For example, the low precision of the raw frequency ranking is compensated by the (here synthetically) high accuracy of sentiment expression detection. However, this means vice versa that the less accurate the information on sentiment expression, the more important is the applied aspect detection approach. Without any sentiment information (scenario A), the differences are quite high (max. 11.8 percentage points).

7.7.5. Varying Foreground Corpus Sizes

In this section we are interested in studying the influence of the foreground corpus size. Recall that our evaluation datasets stem from very popular domains (hotel and camera reviews) for which it is easy to crawl tens of thousands and even millions of review documents from the Web. For other, less popular product types, the available number of customer reviews is expectedly less. Also if the task is to find aspects of a very specific product (e.g., Canon EOS 60D), the amount of customer reviews that



(a) evaluation scenario A



(b) evaluation scenario B2

Figure 7.8.: Hotel corpus: Extrinsic evaluation results with different ranking algorithms for scenarios A/B2.

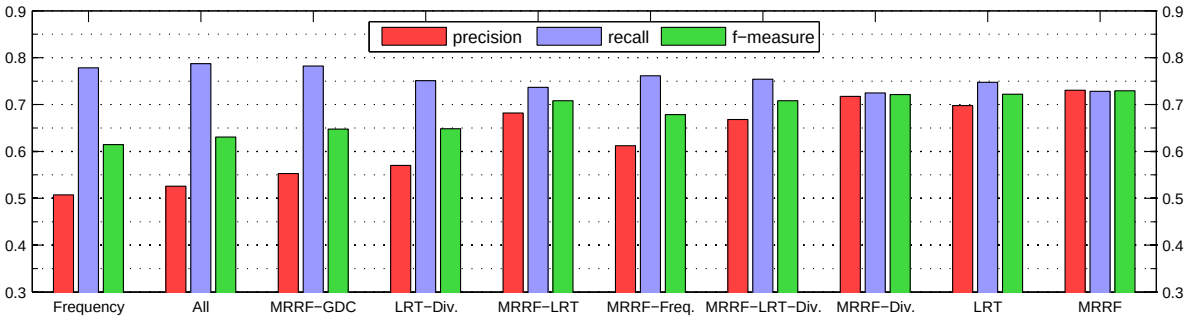
may serve as a foreground corpus is generally less compared to a whole product class. We therefore examine how well the lexicon extraction performs with changing foreground sizes. In particular, we vary corpus sizes from 10 to 50,000 review documents. Each larger corpus contains the reviews of the smaller corpora. We apply the BNP1 part-of-speech pattern, utilize the bBNP candidate acquisition heuristic and activate all candidate filters. No variant aggregation is performed in this evaluation setting. We use the LRT-approach for scoring and ranking candidates. Again, the minimum score for selecting a candidate is set to 3.84.

The results of this study are presented in Figs. 7.10 and 7.11. The figures refer to the two different datasets and each depicts the results of extrinsic evaluation for scenarios A, B2, and B3. Whereas the colors indicate the different evaluation measures (red: precision, blue: recall, green: f-measure), the different marker types determine the examined scenarios (cross: scenario A, circle: scenario B2, square: scenario B3). Since we vary the corpus size within a large range of values, we scale the x-axis logarithmically.

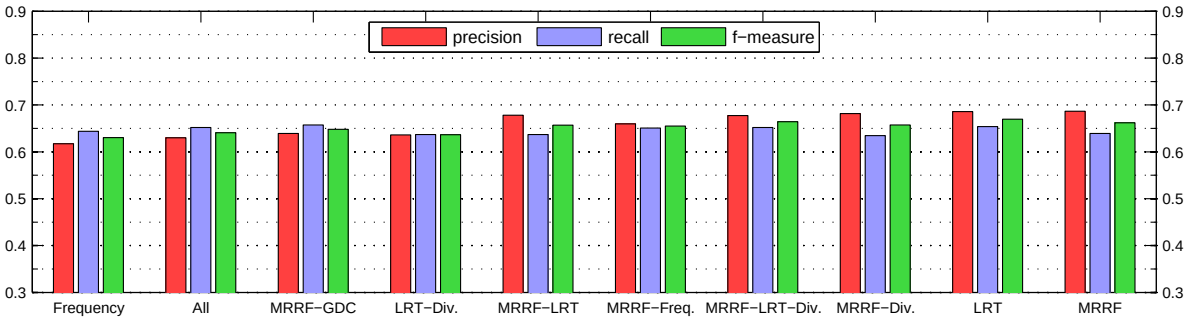
As is to be expected, the recall of the generated lexicons increases with larger foreground corpora sizes. For all three evaluation scenarios we can observe a roughly (despite a few outliers) monotonic increase in recall. The more data we see, the higher the likeliness that a previously missed product aspect is found with significantly high frequency in the foreground corpus. Naturally, the gradient of this increase lowers with larger corpus sizes.

When considering precision, evaluation scenario A is most informative as no gold standard information is incorporated. We observe a steady drop in precision for both datasets. This is also expectable with a lexicon based approach to aspect detection. The larger the generated lexicon[30], the higher is the absolute number of false entries (even at the same level of intrinsic precision), so that more false positives are extracted, which lower the extrinsic precision.

[30] For the foreground corpus sizes of 10, 100, 1000, 10000, and 50000, the resulting number of lexicon entries for the hotel/camera datasets are 4/2, 46/46, 257/213, 693/569, and 1502/1236 entries.



(a) evaluation scenario A



(b) evaluation scenario B2

Figure 7.9.: Camera corpus: Extrinsic evaluation results with different ranking algorithms for scenarios A/B2.

For the synthetic sentiment target extraction scenarios B2 and B3 we cannot observe the same effect of decreasing precision. After an initial drop in precision, the value steadily increases with corpus size. In this case, the larger lexicon sizes favor the precision of the extraction task: While the sentiment information prevents false lexicon entries to result in false extractions, the larger number of entries reduces the amount of false positives due to partial matches. Thus, the higher lexicon coverage leads to both, increased recall and precision. However, for the hotel dataset, we find that beyond a certain size (around 5,000 documents), the precision drops again for both evaluation scenarios. This effect is not observed for the camera dataset. Here, the precision increases monotonically. Comparing the types of false positives encountered at sizes of 5,000 and 50,000 for the hotel dataset, we find that the decrease in precision can be mainly attributed to a higher number partial matches counted as false positives.

Concerning the f-measure, the results for the sentiment target extraction scenarios are slightly diverse within the two different datasets. Whereas for the digital camera datasets the results show a steady increase in f-measure, for the hotel dataset we find that the decrease in precision beyond corpus sizes of 5,000 is higher than the increase in recall (which consequently leads to a lower f-measure for large corpus sizes). Also regarding evaluation scenario A, the highest f-measure is achieved at corpus sizes of around 5,000 documents for the hotel corpus.

From the discussed results, we can generally conclude that even with relatively small corpus sizes of around 1,000 documents, comparably high results for f-measure can be achieved. Regarding the hotel corpus and scenario A, the results for a size of 1,000 documents are even better than for 50,000 documents. Concerning both datasets and the other evaluation scenarios, the maximum increase in f-measure between sizes 1,000 and 50,000 is 7.2 percentage points (camera, scenario B3). On the other hand, since the recall increases steadily with larger corpus sizes, it is obvious that the larger the corpus the better the expectable results. That is in particular true when considering manual post-processing of extracted lexicons. The largest extracted lexicon in our setting exhibits a size of 1502 entries only.

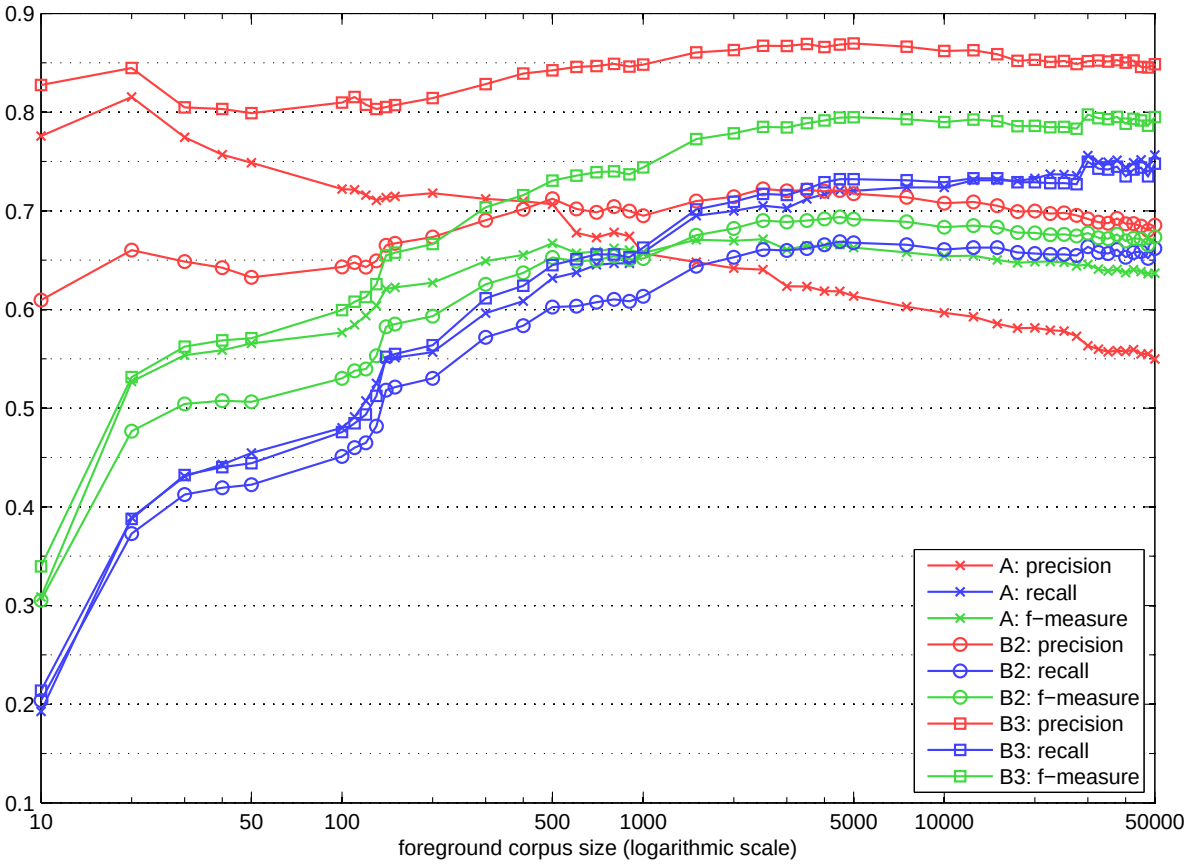


Figure 7.10.: Hotel corpus: Extrinsic evaluation results with different sizes of the foreground corpus.

With comparably small effort false entries can be removed, thus preventing the described effect of lower precision with larger lexicon sizes.

7.7.6. Varying Lexicon Sizes

In the previous experiments with the LRT-approach we used a fixed threshold value of 3.84 for selecting candidates. Although we can associate this threshold with a confidence level for the likelihood-ratio statistical test (here: 99%), it is unclear whether this choice is reasonable and valid for different domains. We therefore experiment with different lexicon sizes and examine the corresponding LRT-scores. In particular, we apply the top-k selection strategy while varying k from 10 to 2,000 with different step sizes. We choose a maximum k of 2,000 for the simple reason that the relatively restrictive acquisition heuristic bBNP does not find more candidates in our 20,000 document foreground corpora.

The results presented in Fig. 7.12 are based on an extraction process with all basic filter and variant aggregation approaches activated. As before, the green color represents the f-measure, whereas blue and red refer to recall and precision, respectively. The different marker types indicate the different test corpora (cross: hotel, circle: camera).

As expectable, precision decreases with larger lexicon sizes, whereas recall values improve. By definition, the recall must increase monotonically in this evaluation scenario, which we can observe in the figure: Adding lexicon entries may only reduce the number of false negatives, and never increase it. On the other hand, the addition of entries may increase *or* decrease the precision value. In fact, we

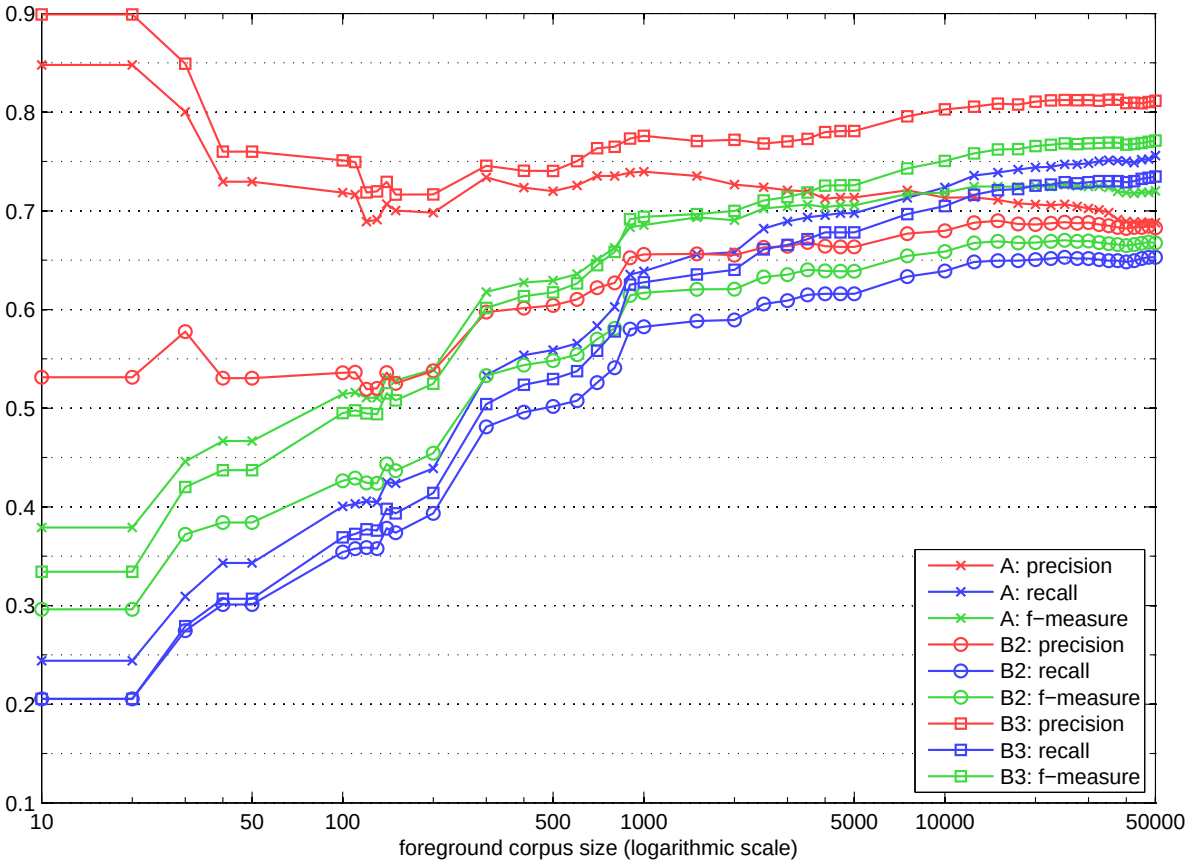


Figure 7.11.: Camera corpus: Extrinsic evaluation results with different sizes of the foreground corpus.

find that the precision is relatively constant for medium sized lexicons and then drops significantly[31] at a certain lexicon size. For the hotel dataset this point is at a size of about 1,100 entries and for the camera dataset at roughly 900 entries. Translating these sizes to the corresponding threshold value for the LRT-score, we find that in both datasets the threshold is about 1.5. Due to the drop in precision, f-measure begins to decrease with lexicon sizes beyond the mentioned ones — in other words, with the addition of entries exhibiting a score less than 1.5. Observing this threshold in two independent datasets, we may conclude that it is generally reasonable to use a fixed threshold of 1.5 for the LRT-approach.

We further examine the results, we would achieve when calculating the threshold value dynamically with the outlier detection approach proposed by Jakob et al. [185] (see also Section 7.4.6). For both datasets, the mean value of the scores is relatively high with 289.1 (hotel) and 237.4 (camera). The 40 highest scored term candidates all have scores of at least 1,000 and the highest scored candidates "hotel" and "camera" show values of over 60,000 and nearly 80,000, respectively. This also explains the high standard deviation of ∼2,200 (hotel) and ∼2,450 (camera). Applying the outlier detection selection strategy, we calculate very high thresholds, so that in both datasets only about the 25 highest ranked candidates are selected. Using these small lexicons, the f-measure achieved for both test corpora is consistently about 10 percentage points lower than the optimal value. For our setting, we thus cannot confirm the findings by Jakob et al. [185], who observe an improvement with dynamically selected threshold values for small foreground corpora (hundreds of review documents instead of several ten thousands).

[31] Take note that the logarithmic scale of the x-axis further stresses this effect.

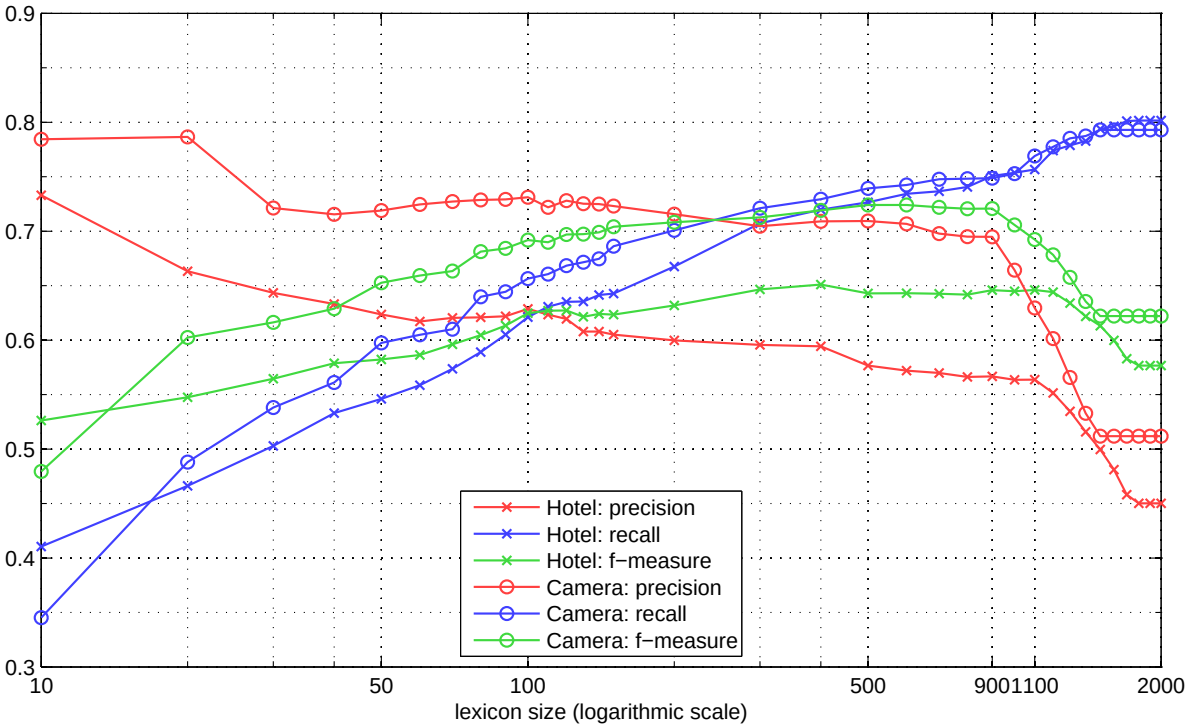


Figure 7.12.: Extrinsic evaluation results with different lexicon sizes.

7.7.7. Effectiveness of Indirect Crowdsourcing Approach

In this section we examine the effectiveness of incorporating the weakly labeled data from pros and cons parts of customer reviews. We consider the pre-modifier filter discussed in Section 7.5, which we use to remove target-specific, sentiment related modifiers from candidate terms (e.g., "long" in "long battery life" or "large" in "large room").

Recall that we designed a statistical test that compares the strengths of association between modifiers and heads in pros and cons documents. To generate the necessary statistics for the test, we need to count the frequency of the heads in isolation and the co-occurrence frequency of heads with their modifiers. We define co-occurrence by means of a window of 3 tokens around the head: A modifier co-occurs with a head h, if it precedes the first token of h with a distance of at most 3 tokens (i.e., at most 2 interjacent tokens) or succeeds the last token of h, also with a distance of at most 3 tokens. This less restrictive definition of co-occurrence allows us to capture different types of relevant modifier-head combinations. For example for the modifier-head combination of "large" and "room", we can find occurrences in such diverse phrases as "room is very large", "large hotel room", "liked the large, comfortable room", or simply "large room".

As pros/cons corpora we use the document collections as described in Section 7.6. That is for the hotel domain, we make use of 100,000 documents for pros and cons each. For the digital camera domain, the set of pros and cons has a cardinality of 50,000 documents each. We found that results are improved when applying a relatively high threshold for rejecting the null hypothesis of the statistical test. In this particular experiment we set the threshold to 10.83 which corresponds to a confidence level of 99.9%. Table 7.7 shows the results of applying the pros/cons pre-modifier filter. Take note that the filter is applied in addition to the other, "basic" filters which we examined earlier. The indicated improvements refer to the setting when only the basic filters are applied. No variant aggregation is performed.

We find that incorporating weakly labeled data by means of the pre-modifier filter results in a sig-

		nominal mentions		
dataset	**scenario**	precision	recall	f-measure
Hotel	A	0.580 (0.011**)	0.763 (0.011)	0.659 (0.011**)
Hotel	B1	0.479 (0.018**)	0.774 (0.024**)	0.592 (0.021**)
Hotel	B2	0.737 (0.030**)	0.698 (0.027**)	0.717 (0.028**)
Hotel	B3	0.889 (0.032**)	0.774 (0.024**)	0.828 (0.027**)
Camera	A	0.717 (0.025**)	0.762 (0.019)	0.739 (0.022**)
Camera	B1	0.496 (0.027**)	0.751 (0.029**)	0.598 (0.028**)
Camera	B2	0.714 (0.034**)	0.677 (0.028**)	0.695 (0.031**)
Camera	B3	0.834 (0.040**)	0.751 (0.029**)	0.790 (0.034**)

Table 7.7.: Results for product aspect and sentiment target detection when activating all filters and additionally using the pros/cons pre-modifier filter. Improvements are reported with respect to the same setting without the pros/cons pre-modifier filter.

nificant improvement of f-measure. The maximum increase of f-measure is 3.4 percentage points and is observed for the camera dataset in evaluation scenario B3. The minimum gain in f-measure is 1.1 percentage points in scenario A for the hotel corpus, which is still significant with regard to the 99% confidence level. We further observe that the improvement is due to both, higher precision and higher recall, which is expectable: The pre-modifier filter is to remove erroneous modifiers from candidate terms and thus reduces the amount of partial matches. This lower number of partial matches leads to less false positives and false negatives (i.e., to higher precision and higher recall).

For the hotel dataset, the filter removes 36 false pre-modifiers, such as in "free breakfast", "helpful staff", "comfortable bed", "spacious room", or "huge bathroom". It further removes pre-modifiers from completely false candidates, such as "down side" or "only regret". In effect such terms are completely removed from the generated lexicon: Scoring with a background corpus results in too low domain relevance for the terms "side" or "regret". The size of the generated lexicon is reduced to 940 entries (was 975)[32]. With regard to the digital camera dataset, we find that 33 false pre-modifiers are detected, effectively leading to a lexicon with a size of 738 entries (was 767). Exemplary candidates with detected false modifiers are "low price", "big screen", "proprietary battery", or "slow shutter speed".

The precision of the generated lexicons increases to 0.734 (+2.7 percentage points) concerning the hotel dataset and to 0.846 (+2.9 percentage points) for the camera corpus. Despite this small increase, and although relatively few lexicon entries are altered by means of the filter, we observe the mentioned (significant) gain in f-measure of 3.2 percentage points. For both datasets this is mainly because the affected lexicon entries exhibit a high frequency of occurrence in the evaluation datasets (e.g., "large room" or "low price").

7.7.8. Manual Revision of Generated Lexicons

In the following, we manually revise the priorly extracted lexicons and examine achievable results with the post-processed lexicons. By definition, intrinsic evaluation of the lexicons exhibits a perfect accuracy of 100% (using our annotation guidelines as reference). As a basis for the manual revision, we use the lexicons generated with the BNP2 part-of-speech tag pattern, the bBNP acquisition heuristic and all filter (including the pros/cons pre-modifier filter) as well as variant aggregation techniques activated. We opt for the BNP2 pattern as it generally allows for higher recall — the lower precision is compensated by the manual validation process. For ranking we use the LRT-approach.

[32] By removing false pre-modifiers, the observed frequency of occurrence for the head noun may be increased. These higher counts may lead to a higher LRT-score, potentially promoting a candidate to the lexicon. In our case, a single term is promoted to the lexicon.

		nominal mentions		
dataset	**scenario**	precision	recall	f-measure
Hotel	A	0.793 (0.164**)	0.792 (0.173**)	0.793 (0.169**)
Hotel	B1	0.598 (0.109**)	0.818 (0.183**)	0.691 (0.138**)
Hotel	B2	0.804 (0.158**)	0.760 (0.176**)	0.781 (0.168**)
Hotel	B3	0.914 (0.115**)	0.818 (0.183**)	0.863 (0.156**)
Camera	A	0.776 (0.138**)	0.837 (0.089**)	0.805 (0.116**)
Camera	B1	0.539 (0.111**)	0.830 (0.115**)	0.654 (0.118**)
Camera	B2	0.772 (0.140**)	0.760 (0.130**)	0.766 (0.135**)
Camera	B3	0.873 (0.144**)	0.830 (0.115**)	0.851 (0.129**)

(a) strict evaluation metric

		nominal mentions		
dataset	**scenario**	precision	recall	f-measure
Hotel	A	0.869 (0.131**)	0.867 (0.142**)	0.868 (0.137**)
Hotel	B1	0.649 (0.068**)	0.887 (0.134**)	0.749 (0.094**)
Hotel	B2	0.869 (0.112**)	0.821 (0.137**)	0.844 (0.126**)
Hotel	B3	0.991 (0.043**)	0.887 (0.134**)	0.936 (0.097**)
Camera	A	0.869 (0.097**)	0.937 (0.032**)	0.902 (0.068**)
Camera	B1	0.608 (0.074**)	0.937 (0.045**)	0.738 (0.070**)
Camera	B2	0.854 (0.080**)	0.841 (0.070**)	0.847 (0.075**)
Camera	B3	0.984 (0.076**)	0.937 (0.045**)	0.960 (0.060**)

(b) lenient evaluation metric

Table 7.8.: Results for product aspect and sentiment target detection with manually revised lexicons. Improvements are reported with respect to the results of the previous section.

The manual revision is performed by a single human annotator. As the precision of the automatically extracted lexicons is already sufficiently high, the revision process can be performed in approximately 3 hours for each lexicon. In addition to using the results from the LRT-ranking, we manually select the ten most frequent terms that represent a valid product aspect, but are not included in the automatically derived lexicons. We add these terms to the revised dictionaries. For the hotel domain, this includes for example the terms "internet", "area", or "value". As can be seen, these are all terms that are so frequent in common language that the contrastive LRT-score is lower than the fixed threshold of 3.84. After the whole process, the lexicons consist of 1,040 (hotel) and 1,271 (camera) entries.

We regard these reworked lexicons as a reference to results achievable with the fully automatic approaches presented earlier: Table 7.8 presents extrinsic evaluation results with the manually revised lexicons in comparison to the best reported results so far (Table 7.7). Whereas Table 7.8a refers to results interpreted with the strict evaluation metric, Table 7.8b shows results according to the lenient metric.

First, we note that for both metrics and in all evaluation scenarios, the f-measure increases significantly (at minimum 6.0 percentage points and at maximum 16.9 percentage points). With regard to strict evaluation, we even observe a minimum increase in f-measure of more than 11 percentage points in each scenario. In particular, for the aspect detection evaluation scenario A, we achieve an f-measure of 79.3 percent for the hotel dataset and a value of 80.5 percent for the camera corpus. In the synthetic scenario B3, the results are even higher with 86.3% (hotel) and 85.1% (camera). The considerably improved f-measure in all scenarios is based on both, higher recall and higher precision values. Whereas the higher precision is mainly due to sorting out false lexicon entries (e.g., "day",

"city", or "street" in the hotel domain), the increased recall stems from the application of the less restrictive BNP2 acquisition pattern and the manual addition of the ten most frequent aspects, which were missing in the original lexicons. With regard to recall, we observe very high gains in the hotel dataset with improvements between 17.3 and 18.3 percentage points. Concerning the camera corpus, the gains in recall are between 8.9 and 13.0 percentage points. In absolute numbers we achieve recall values of about 80 percent and more in both datasets and in the different evaluation scenarios.

When considering the lexical variability discussed in Chapter 6, we can conclude that the semi-automatic approach examined in this section, is capable of incorporating roughly the 25% most frequent nominal mentions in the test corpora. From corpus analysis we also know that nearly 70% of all distinct nominal mentions occur only once in the evaluation corpora. As discussed before, these rare mention types are hard or impossible to detect with frequency based methods such as the LRT-approach. Due to the roughly Zipfian distribution of distinct mention types, even with our relatively large foreground corpora of 20,000 documents, the majority of mention types occurs only once in the corpus. However, for both corpora the 125 most frequent mention types (25% of around 500 distinct types) already account for nearly 80% of all occurrences. The semi-automatic process is able to detect the majority of these terms and thus achieves recall values of around 80 percent. Looking at the results evaluated with the lenient metric, we even observe recall values of over 90% for the camera dataset and nearly 90% for the hotel corpus. A major share of the rare mention types occurring only once are derivations of more frequent (lexicon included) terms — for instance, the rare term "rooftop swimming pool" refers to a specific type of "swimming pool", which is frequent in the corpus. With the lenient metric such occurrences are evaluated as correct extraction. We further know from the mistake analysis conducted in Appendix C.3, that in most cases ($\sim$75%) the sense of the extraction remains unaltered with such partial matches. Thus, the lenient metric only slightly overestimates the "true" f-measure (i.e., when counting sense-preserving partial matches as correct).

With respect to precision, the most significant gains are achieved in scenarios A and B2. We observe improvements of over 14 percentage points and more in both datasets (strict metric). But also in the synthetic scenario with the highest accuracy of sentiment information (B3), gains of 11.5 (hotel) and 14.4 (camera) percentage points are measured. With lenient evaluation the gains are less. Although lexicons with perfect accuracy let expect perfect precision in the synthetic scenario B3 under lenient evaluation, we observe precision values of slightly less than 100%. The single source of false positives in this configuration is the case when compositional variants, such as "design of the camera", are not included in the lexicon, but the involved terms (here: "design" and "camera") are. In this case we count a single false positive (the second term, here: "camera"). Reduced precision in synthetic scenarios B1 and B2 is also partly due to this issue, but mainly due to the insufficient information regarding sentiment expressions. False positives in scenario A are primarily caused by the missing context awareness of the lexicon approach, as well as the lexical ambiguity of lexicon entries. We find that, for the hotel review domain, the observed precision values are generally higher than for the digital camera domain, whereas the recall values are generally lower. We attribute this finding to the different lexicon sizes (the hotel lexicon has a smaller size) and to the slightly different lexical variability of the two domains.

When comparing the two different datasets in each evaluation scenario, we find that the results for the f-measure are very similar when utilizing the manually revised lexicons (maximum difference less than 4 percentage points). Especially in scenario A, which does not rely on gold standard information, we observe nearly identical results for the f-measure (about 80 percent). This observation indicates that results for other domains may not differ significantly.

7.8. Summary and Conclusions

In this chapter, our goal was to provide a detailed study of lexicon-based approaches to the extraction of fine-grained product aspects in customer review datasets. We stressed the relevance of this task by highlighting its importance as subtask of an aspect-oriented sentiment analysis system. In particular, we set focus on unsupervised, corpus-based techniques for the automatic creation of the needed knowledge bases.

In the introductory part, we elaborated on the basic **characteristics of such knowledge bases**. We pointed out that they are mainly characterized by their degree of supervision during acquisition (manual, semi-automatic, or fully automatic acquisition), their immanent definition of relevance (product or product-class centric), their degree of structuring (only terms or higher level semantic information), and by their coverage.

Related work in Section 7.2 gave an overview of other studies that are most relevant in our context. We basically distinguished unsupervised and supervised approaches to lexicon creation, while concentrating on works that explicitly target the area of sentiment analysis. Our goal was to distill the most substantial ideas and methods examined in the literature, which we summarized in Table 7.1.

Section 7.3 introduced the basic concepts of **terminology extraction**. We reviewed the typical pipeline architecture of such a system and elaborated more closely on measures for defining term relevance. In particular, we identified measures based on contrastive domain relevance, intra domain relevance, as well as term cohesion. In Section 7.4, we cast the task of finding product aspects as an instance of a terminology extraction problem. We discussed our concrete approaches to each step of the extraction pipeline in detail. A specific approach to incorporate weakly labeled data was proposed in Section 7.5.

Section 7.7 covered our experiments. Our goal was to examine the general applicability of lexicon-based approaches, as well as to study the influence of the various parameters and different configurations of our system. We evaluated the system intrinsically and extrinsically. For extrinsic evaluation, we compared lexicon-based extractions with the gold standard annotations of our customer review corpora. We examined a scenario for product aspect detection and three different scenarios for sentiment target extraction. To view the lexicon-based target detection in isolation, we explicitly did not rely on automatic sentiment discovery techniques, but provided synthetic sentiment information from the gold standard annotations. Since the system is tailored towards the detection of nominal mentions of product aspects, results were mainly reported with regard this mention type. In the following we summarize the major findings of our experiments:

- Lexicon-based approaches to product aspect detection show consistently good results in different settings. With relatively effortless manual revision, f-measures of 80% are measured for two different datasets. In the best fully automatic setting f-measures are only about 10 percentage points lower.

- Product aspect extraction with a lexicon mainly suffers from the following problems: False positives are produced by the missing context awareness, lexical ambiguity of terms, as well as an inaccurate lexicon acquisition process. False negatives are mainly due to low frequency terms (Zipf's law) and too common terms that are not included with a contrastive term relevance measure. Unrecognized part-of-speech patterns of aspects only play a minor role. Instead of extending the acquisition patterns (e.g., to include also verbs and prepositions as for "easy/JJ to/TO use/VB"), the small set of affected aspects is preferably manually added to the lexicon.

- Regarding the accuracy of extracted lexicons, we identified primarily four types of errors. Frequency based methods for lexicon acquisition erroneously tend to extract non-aspect terms that are generally either closely related to (1) the product type (e.g., "subway") or (2) the domain of

reviews (e.g., "problem"). Third, the acquisition heuristics and ranking measures do not prevent the extraction of false pre-modifiers and also tend to find overly specific terms (e.g., "4gb memory card"). Both failure types often lead to partial matches.

- The presented candidate filtering techniques significantly increase the accuracy of the extracted lexicons. For both datasets, accuracy is more than 10 percentage points higher. All filters contribute to the improved results, while the GlossEx method is responsible for the major share. The higher lexicon quality lets observe f-measures that are up to 4 percentage points higher. Leveraging the weakly labeled data from pros and cons lists to remove unidentified false pre-modifiers, additionally improves f-measure up to 3.4 percentage points.

- The application of variant aggregation techniques promises only minor improvements in recall (< 2.5 percentage points in the best case). Closer analysis showed that the major share of variants is not detected due to their rarity, that is, their non-occurrence in the foreground corpus. Instead or in addition to aggregating variants during lexicon acquisition, the step may also be performed during lexicon application. Using fuzzy/approximate string matching algorithms[33] during lexicon application may improve results.

- With regard to acquisition heuristics we found that the bBNP heuristic consistently outperforms all other methods. Incorporating sentiment information in the acquisition process (as with the SBP heuristic) did not improve results. The less restrictive BNP2 part-of-speech pattern expectedly exhibits a lower lexicon accuracy, but allows for improved recall values. When applying a manual revision process, which compensates the lower precision, it is indicated to use the BNP2 pattern.

- Experiments with the five different ranking algorithms showed that contrastive relevance measures such as the LRT-score or MRRF-score perform best. Using raw frequency only leads to very low lexicon accuracy. Combining different measures in a weighted-rank scheme did not reveal any improvements. For sentiment target detection, the results further show that differences between the various methods become less significant the more accurate sentiment information is available (this is generally true). By implication, this means that accuracy and coverage of the lexicons becomes more important the less accurate the sentiment information.

- The presented unsupervised terminology extraction approach benefits from larger foreground corpora sizes. Comparison to results reported on the Hu/Liu corpus showed major improvements. For the fully automatic setting, we achieved the best f-measure with foreground corpus sizes between 1000 and 5000 documents. Larger corpora lowered the precision. However, in a semi-automatic setting (i.e., with a manual post-processing step), increasing the foreground corpus is generally better.

- Our experiments with varying lexicon sizes revealed that a fixed threshold of 1.5 for the LRT-score achieved the best results on both datasets. At least when using large foreground corpora, we could not confirm that dynamically computed thresholds, as proposed by Jakob et al. [185], lead to good results. These thresholds only selected the 25 highest ranked candidate terms.

[33] Gusfield [156] provides a good overview of approximate string matching techniques.

When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind: it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the stage of science.

Lord Kelvin

8. Detection of Product Aspect Mentions at the Sentence Level

In this chapter, we examine methods for implementing the topic dimension of the discourse oriented model for customer reviews. In particular, our goal is to automatically attribute each discourse segment with one or more coarse-grained aspects from a set of predefined topics. We cast the task of attributing discourse segments with topics as an instance of a *multi-label text categorization* problem and experiment with lexicon-based and supervised machine learning approaches. We further propose an indirect crowdsourcing method that exploits the correlation between section headings and topics in reviews. As predetermined by our annotation scheme, we implement the discourse oriented model at the sentence level. We refer to Appendix D which explains how we automatically derived the set of predefined aspects with a *probabilistic topic modeling* approach.

The remainder of this chapter is organized as follows: Section 8.1 formalizes the concrete problem setting as a multi-label classification task. In Section 8.2, we cover our lexicon-based approach for topic classification and Section 8.3 describes the supervised methods that we experiment with. In Section 8.4, we propose a method for acquiring weakly labeled data that can be exploited for learning a topic classification model. We summarize and conclude our findings in Section 8.5. Related work will be discussed within the individual sections.

8.1. Problem Description

As already indicated in Section 6.1.2, the task of attributing topics to discourse segments (that is sentences) is actually a **multi-label** text categorization problem (around 15% of on-topic sentences are associated with multiple topics). In contrast to standard, single-label classification, a multi-label classifier may assign multiple labels to a single instance. For example, consider the sentence "This camera is so easy to use and takes great pictures in low-light conditions.". The sentence addresses two relevant topics and an ideal multi-label classifier would assign the labels "ease of use" as well as "low-light performance". To put it more formally:

Definition 8.1 (Multi-label Classification). *Let $\mathcal{L} = \lambda_1, \lambda_2, \ldots, \lambda_k$ be a finite set of k distinct labels and $\mathcal{X} = x_1, x_2, \ldots, x_n$ the set of instances that are to be classified. Then, a multi-label classifier is a function $c : \mathcal{X} \rightarrow \mathcal{P}(\mathcal{L})$, with $\mathcal{P}(\mathcal{L})$ being the power set of $\mathcal{L}$. In other words, a multi-label classifier maps an instance x_i either to the empty set $\emptyset$ or to any subset of $\mathcal{L}$.*

Besides taking into account the multi-label characteristic of our problem setting, we need to consider a further aspect: With the product type taxonomy (refer to Section 4.1), we introduced a **hierarchical organization** of the predefined set of categories (i.e., coarse-grained aspects). For example, the broader category "sleep quality" may subsume the narrower concepts "bed" and "noise". For a more precise understanding of this aspect, we formalize the three main properties with regard to our hierarchy[1]:

[1] We partly adapt the notation used by Esuli et al. [122].

Definition 8.2 (Properties of the Hierarchical Label Organization). *Let $\lambda \in \mathcal{L}$ be one of the predefined labels and let $\Downarrow(\lambda)$ / $\Uparrow(\lambda)$ denote the set of all descendant/ancestor categories of λ (excluding the root node of the hierarchy). Further, let $\mathcal{X}_\lambda \subseteq \mathcal{X}$ be the subset of instances which are labeled with λ. Then our hierarchy fulfills the following three properties:*

1. *Each instance $x \in \mathcal{X}_\lambda$ that is labeled as category λ, implicitly belongs to all ancestor categories $\lambda_a \in \Uparrow(\lambda)$. That is, $x \in \mathcal{X}_\lambda \Rightarrow x \in \mathcal{X}_{\lambda_a} | \forall \lambda_a \in \Uparrow(\lambda)$.*
2. *An instance $x \in \mathcal{X}_\lambda$ that is explicitly labeled as λ does not belong to the descendant categories $\lambda_d \in \Downarrow(\lambda)$. In other words, a parent category may be broader than the union of all its descendant categories. In fact, we allow that $\bigcup_{\lambda_d \in \Downarrow(\lambda)} \mathcal{X}_{\lambda_d}$ may be a proper subset of $\mathcal{X}_\lambda$.*
3. *The labels of a multi-labeled instance x may stem from arbitrary subtrees of the hierarchy. More precisely: Let λ_1 and λ_2 be the labels attached to instance x. Then $\Uparrow(\lambda_1) \cap \Uparrow(\lambda_2)$ may in fact be the empty set $\emptyset$.*

Whereas for traditional binary or multi-class classification, it is straightforward to calculate standard evaluation metrics such as *accuracy*, *precision*, *recall*, or *f-measure*, it is not directly obvious for multi-label, hierarchical classification tasks. Appendix E describes the metrics we use to evaluate our approaches.

8.2. Unsupervised, Lexicon-Based Approach

In this section we discuss a lexicon-based approach to discovering coarse-grained product aspects in customer reviews. As introduced earlier, the concrete goal is multi-label classification of sentences, where the labels represent the coarse-grained aspects (topics) predominantly discussed within a sentence. The basic idea with a lexicon-based approach is simple:

Each lexicon entry is associated with one of the predefined topics. For example, the terms "water pressure", "restroom", "toiletries", and "bathroom amenity" are all linked to the same topic "bathroom". During application, a review sentence is parsed and all token sequences that match a lexicon entry are extracted. For instance, in the sentence "The restroom is pretty good, with fresh smelling portico toiletries.", the terms "restroom" and "toiletries" are extracted. By looking up the terms' associated topics, we determine the relevant labels for a sentence. In this case, we would find that both terms refer to the aspect "bathroom" and thus would attach the label "bathroom" to the sentence. With respect to the labeling task, lexicon-based approaches exhibit the following properties :

- Accuracy: Matching token sequences can be compared to a very simple rule-based system. Each match triggers a rule that outputs the appropriate label. As these "rules" are manually crafted, it can generally be assumed that they show a relatively high precision.
- Sensitivity: The achievable recall with a lexicon-based approach mainly depends on the size and coverage of the employed knowledge base. Typically, a knowledge base for customer review mining is limited to cover nominal (e.g., "camera size" or "startup time") and possibly named mentions of product aspects, but does not address implicit mentions. For example, phrases such as "fits into pocket" (referring to aspect "dimensions") or "starts up rapidly" (referring to aspect "speed") are not included. Such a limitation generally lowers the recall.
- Lack of context-awareness: A negative impact on precision is due to missing context-awareness. Each match is evaluated separately and no context information is considered. For example, due to a missing word sense disambiguation, a sentence such as "The desk in our room was much

too small." may be falsely attributed to the topic "service" as the word "desk" is included in the lexicon as short version of the term "front desk".

- Flexible and interpretable: In contrast to models obtained with machine learning approaches such as SVMs or logistic regression, a knowledge base has the advantage of being directly interpretable. This eases the ability to fine-tune and configure the approach to specific application needs. For instance, one can easily add missing terms to improve the coverage, adjust the lexicon to reflect changes of the predefined topic set, or modify the desired depth of hierarchy.

- Manual effort: Naturally, the manual construction of high-quality knowledge bases induces a considerable amount of effort and involves several iterations of fine tuning.

8.2.1. Implementation

Our knowledge base implements the *product type taxonomy* we introduced in Section 4.1. That is, we represent the set of coarse-grained aspects in a hierarchical manner and associate each fine-grained aspect to exactly one of the nodes in the hierarchy. We restrict the depth of this concept level hierarchy to only two levels (not counting the root). Each "cluster" of fine-grained aspects is also hierarchically organized. Furthermore, our knowledge base represents the semantic relations between the different aspects on both, the concept and mention level[2]. It is constructed in a manual process, which is composed of the following steps:

1. **Initial terminology extraction**: We apply the terminology extraction approach that we discussed in the previous chapter to obtain a flat, unstructured list of fine-grained product aspects for the product type under consideration. In particular, we use the configuration described in Section 7.7.8 — that is, we use the BNP2 extraction pattern and the bBNP acquisition heuristic together with all filters and variant aggregation techniques. For ranking, we employ the LRT-score and a threshold of 3.84. Extraction is performed on corpora of 20,000 review documents each. After manual revision, the resulting lists cover 1,040 (hotel) and 1,271 (camera) product aspects (associated variants not counted).

2. **Manual clustering**: Given a flat list of fine-grained product aspects, we manually cluster all terms according to their association with one of the predefined topics (i.e., the number and shape of clusters equals the number and shape of the topics). For the major share of terms, finding the correct cluster is straightforward for a human. However, for some terms, especially single word terms, the correct association may be unclear or ambiguous. For instance, it may be difficult to decide on the related topic for the term "room service". We may either link it with the topic "service" or with the concept "dining". In such cases we consult the results of the LDA topic modeling process (see Appendix D). We look up the strength of association of the term with the topics in question: we determine the shares of sentences that contain the term and are linked to a relevant topic. In the exemplary case of "room service", we find that the term significantly more often occurs in the context of the concept "dining" (highlighting the fact that reviewers typically evaluate the room service in terms of food quality). Terms that we cannot reliably associate with one of the predefined topics are simply discarded. To provide some numbers, the largest cluster in the hotel domain — "service" — is represented by 90 unique fine-grained aspects, whereas the smallest topic — "security" — is related to only five aspects.

3. **Hierarchical organization and semantic relations**: Having clustered the list of fine-grained product aspects, the next step is to hierarchically organize each cluster along the four semantic

[2]Considering the categorization of Buitelaar and Magnini [60], our knowledge base can be considered as adhering to the fourth or fifth level of complexity.

relations "part-of", "feature-of", "type-of", and "synonym-of". This step is also performed manually and straightforward for a human. Take note that for the experiments conducted in this chapter, we do not consider the mention level hierarchy, but consider only the concept level hierarchy and the clustering. The semantic relations will be exploited in the context of sentiment lexicon extraction, which we will describe in Section 9.3.

The whole process of clustering and organizing the individual terms took about 12 person-hours of work for each of the two knowledge bases (hotel and camera). We implement the product type taxonomy by means of an XML Schema — that is, the actual data, the hierarchy, and the relation information are stored as an XML document. Listing 8.1 shows an excerpt of such a document. Take

```
<?xml version="1.0" encoding="UTF-8"?>
<producttype parent-relation="root" label="hotel">
  ...
  <aspect label="dining" parent-relation="feature-of" defines-subtopic="true">
    <aspect label="restaurant" parent-relation="feature-of">
      <aspect label="hotel restaurant" parent-relation="type-of"/>
      <aspect label="restaurant staff" parent-relation="feature-of"/>
      ...
    </aspect>
    ...
    <aspect label="breakfast" parent-relation="feature-of" defines-subtopic="true">
      <aspect label="hotel breakfast" parent-relation="type-of"/>
      <aspect label="breakfast offering" parent-relation="feature-of">
        <aspect label="breakfast option" parent-relation="synonym-of"/>
        <aspect label="breakfast choice" parent-relation="synonym-of"/>
        <aspect label="breakfast selection" parent-relation="synonym-of"/>
        <aspect label="breakfast spread" parent-relation="synonym-of"/>
      </aspect>
      ...
    </aspect>
    ...
  </aspect>
  ...
</producttype>
```

Listing 8.1: Excerpt from an XML document representing the product type taxonomy for the hotel domain. The attribute "parent-relation" defines the relation type between the aspect and its parent. By setting the attribute "defines-subtopic" to "true" an aspect is marked as constituting one of the predefined topics.

note that the knowledge base does not provide any confidence scores. We neither quantify our confidence in including an aspect within the dictionary, nor do we provide a value that indicates our confidence in the topic association. Furthermore, the lexicon does not encode any information about correlations between the different topics or individual terms. Each term is associated with exactly one topic. Figures 8.1 and 8.2 illustrate the concept level hierarchy of the resulting knowledge bases. It

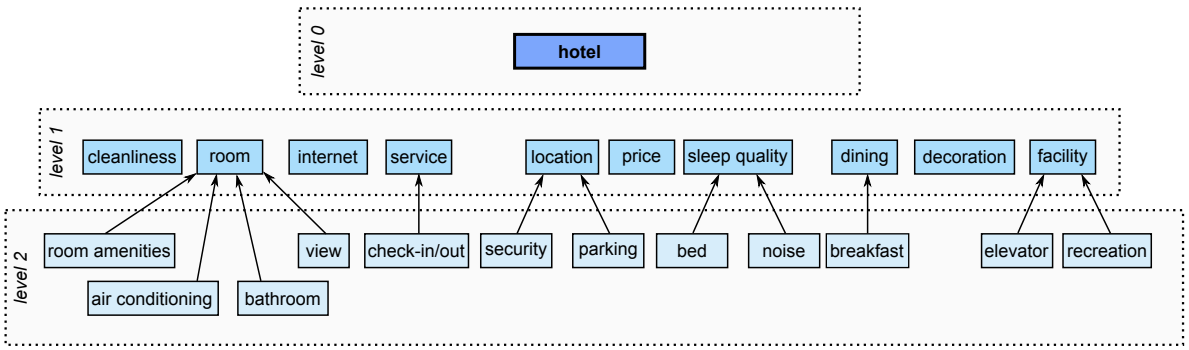


Figure 8.1.: Product type taxonomy for the domain of hotel reviews (restricted to the concept level).

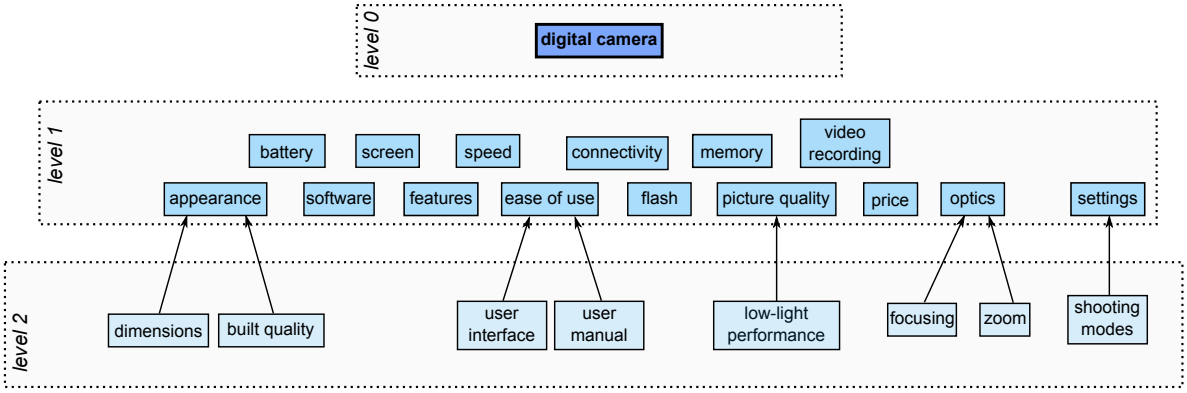


Figure 8.2.: Product type taxonomy for the domain of digital camera reviews (restricted to the concept level).

is important to note that the presented concepts form a subset of the topics we initially gathered by means of the topic modeling approach. Whereas we annotated our evaluation corpora with the complete set of topics, for the experimental setups in the following sections, we subsume or discard some "rare" topics: Naturally, the real support (i.e., within the manually annotated datasets) differs from the support estimated by the topic modeling approach. We found that for some topics that exhibited a low, albeit sufficient estimate, the true support in the evaluation corpus was too low. Each of these topics is either subsumed within another topic or is discarded (i.e., considered as "off-topic"). For the hotel domain, all topics exhibited a sufficient support, thus no topic was subsumed or discarded. With regard to the digital camera domain, we subsumed the topics "image stabilization", "underwater capability", and "face detection" within the main topic "features", the topics "macro mode" and "manual mode" within the main topic "settings", and "lens" within "optics". The topic "customer service" was discarded.

For the actual task of attributing coarse-grained aspects to sentences, we utilize the knowledge base as described in the introductory part of this section. In particular, for matching candidates, we apply the linguistically informed aspect detection procedure defined by Algorithm C.2. Further, we make use of the variants attached[3] to each entry of the knowledge base so that, for instance, spelling mistakes or compound variants can be detected.

8.2.2. Experiments and Results

Experimental Setup

We conduct our experiments on the sentence level annotated corpora of hotel and digital camera reviews. We do not perform any optimization of the lexicons nor any other fine-tuning (i.e., there is no need to set aside a separate *development set*). We use the complete corpora as presented in Chapter 5. However, we distinguish two different experimental settings. In the first setting, we evaluate the lexicon-based approach on all sentences, irrespective of their status regarding the expression of sentiment. In the second setting, we only take into account polar sentences. For this distinction, we utilize gold standard knowledge: Polar sentences are all sentences that have been attributed with a "sentiment polarity" annotation (i.e., including also polar facts). Taking the label-based perspective (see Appendix E), the total number of instances in the first setting (all sentences) is 3861 for the hotel

[3] Variants are collected in a separate file. The final (binary encoded) lexicon that is used for the actual extraction and labeling, is merged from the XML document and the variants file.

	all sentences				polar sentences			
aspect/label	**instances**	**precision**	**recall**	**f-measure**	**instances**	**precision**	**recall**	**f-measure**
recreation	52	0.909	0.962	0.935	38	0.925	0.974	0.949
parking	68	0.967	0.853	0.906	50	0.980	0.960	0.970
view	73	0.940	0.863	0.900	58	0.964	0.931	0.947
internet	65	0.918	0.862	0.889	61	0.964	0.869	0.914
breakfast	117	0.861	0.846	0.853	88	0.923	0.955	0.939
bathroom	120	0.800	0.867	0.832	115	0.879	0.887	0.883
bed	116	0.724	0.948	0.821	107	0.779	0.953	0.857
service	581	0.833	0.754	0.791	506	0.917	0.761	0.832
dining	271	0.706	0.790	0.746	195	0.740	0.846	0.789
room	628	0.629	0.912	0.745	530	0.719	0.908	0.802
elevator	29	0.583	0.966	0.727	24	0.605	0.958	0.742
check in-out	131	0.832	0.641	0.724	86	0.917	0.640	0.753
facility	160	0.609	0.787	0.687	130	0.656	0.777	0.711
sleep quality	262	0.778	0.603	0.680	235	0.817	0.609	0.698
decoration	60	0.968	0.500	0.659	57	0.967	0.509	0.667
room amenities	126	0.760	0.579	0.658	91	0.754	0.571	0.650
price	159	0.794	0.535	0.639	130	0.816	0.546	0.654
air conditioning	37	0.826	0.514	0.633	36	0.818	0.500	0.621
location	470	0.829	0.506	0.629	368	0.877	0.579	0.697
noise	146	0.978	0.301	0.461	128	0.974	0.297	0.455
security	27	0.800	0.148	0.250	23	1.000	0.174	0.296
cleanliness	163	0.567	0.104	0.176	159	0.708	0.107	0.186
micro-average	3861	0.750	0.692	0.720	3215	0.814	0.708	0.757
macro-average	3861	0.766	0.645	0.667	3215	0.813	0.666	0.696

Table 8.1.: Hotel corpus: Results for the lexicon-based detection of coarse-grained product aspects. The table distinguishes between results for all sentences and results for the subset of polar sentences.

dataset and 3329 for the digital camera corpus[4]. Restricting the evaluation corpus to polar sentences only (the second setting), we observe 3215 (hotel) and 2525 (digital camera) instances in total.

Regarding the hierarchy, we evaluate by using the relabeling approach described in Appendix E.2. For the hotel review dataset, our product type taxonomy defines 10 main topics and 12 subtopics. Considering only main topics, the maximum number of instances is attributed to the aspect "room" (628) and the minimum amount to the topic "decoration" (60). For subtopics the maximum number of instances is 146 for the aspect "noise" and the minimum number is 27 for the topic "security". Regarding the digital camera dataset, we have 15 main topics and 8 subtopics. Here, the maximum/minimum number of instances are 483 ("picture quality") and 55 ("software"), as well as 160 ("dimensions") and 48 ("low-light performance").

Results and Mistake Analysis

We summarize our results of the lexicon-based classification in Tables 8.1 and 8.2. For each label/class we report precision, recall, and f-measure for both evaluation settings. Topics are ordered descendingly by the f-measure achieved within in the "all sentences" setting.

The main observation is that we achieve quite good results, even with the very simple lexicon-based approach (and without performing any fine-tuning). In particular, we find micro-averaged f-measures of 72.0% (hotel) and 70.0% (digital camera), respectively. Macro-averaged results are lower with 66.7% and 66.5%[5], indicating that "large" classes (i.e., covering many instances) exhibit a slightly higher f-measure in average. When considering only polar sentences, we consistently observe better

[4] Observe that due to the attribution of multiple labels to a single sentences and our label-based perspective, the actual number of instances is higher than the amount of sentences in the test corpus.

[5] Take note that with macro-averaged computation, the f-measure does not need to exhibit a value between the precision and recall scores.

	all sentences				polar sentences			
aspect/label	**instances**	**precision**	**recall**	**f-measure**	**instances**	**precision**	**recall**	**f-measure**
software	55	0.877	0.909	0.893	40	0.925	0.925	0.925
battery	232	0.921	0.849	0.883	142	0.960	0.845	0.899
zoom	112	0.862	0.893	0.877	90	0.898	0.878	0.888
video recording	128	0.831	0.922	0.874	96	0.885	0.958	0.920
optics	219	0.784	0.913	0.844	166	0.843	0.904	0.872
focusing	54	0.810	0.870	0.839	40	0.872	0.850	0.861
screen	113	0.720	0.956	0.821	85	0.792	0.941	0.860
memory	95	0.741	0.905	0.815	43	0.745	0.884	0.809
price	161	0.854	0.727	0.785	116	0.897	0.750	0.817
user manual	49	0.682	0.918	0.783	41	0.771	0.902	0.831
flash	70	0.619	1.000	0.765	45	0.652	1.000	0.789
shooting modes	80	0.747	0.775	0.761	55	0.792	0.764	0.778
ease of use	326	0.712	0.752	0.731	276	0.806	0.754	0.779
features	150	0.674	0.773	0.721	108	0.693	0.815	0.749
user interface	113	0.647	0.779	0.707	85	0.734	0.812	0.771
settings	182	0.672	0.731	0.700	101	0.667	0.772	0.716
picture quality	483	0.434	0.899	0.586	439	0.584	0.895	0.707
connectivity	68	0.702	0.485	0.574	37	0.783	0.486	0.600
speed	87	0.745	0.402	0.522	70	0.867	0.371	0.520
dimensions	160	0.744	0.381	0.504	124	0.780	0.371	0.503
appearance	269	0.676	0.357	0.467	220	0.745	0.345	0.472
low-light performance	48	1.000	0.188	0.316	42	1.000	0.143	0.250
built quality	75	0.500	0.120	0.194	64	0.583	0.109	0.184
micro-average	3329	0.664	0.739	0.700	2525	0.744	0.735	0.740
macro-average	3329	0.706	0.688	0.665	2525	0.761	0.686	0.687

Table 8.2.: Camera corpus: Results for the lexicon-based detection of coarse-grained product aspects.

results. With regard to both datasets, the micro-averaged f-measure is approximately four percentage points higher (75.7% and 74.0%) in this setting. We find that this increase is mainly due to a significantly higher precision (+6.4 and +8.0 percentage points). As the majority of false positives stems from a lack of context awareness (see next paragraph), we conclude from the increased precision with polar only sentences, that in this setting the correlation between a product aspect mention and the associated topic is stronger.

We now take a closer look at the main causes for false positives. To do so, we randomly sample a set of 300 false positives for each experimental setting ("all sentences" and "polar sentences") and analyze these representative subsets. The samples stem from a combined set of false positives from both application domains (hotel and camera). Naturally, we find similar reasons as already discussed in Section 7.6. The great majority of false positives originates from the **lack of context awareness** of the lexicon-based approach. With regard to the experimental setting "all sentences", 84.9% of false positives can be attributed to this issue. As examples, consider the sentences "The lamp next to the bed had a hole in the base." or "The front desk was like being waited on at a fast food restaurant.". In the first sentence the lexicon-based approach matches the keyword "bed" and in the second sentence it is the word "restaurant". The former sentence is erroneously labeled as topic "bed", the latter as topic "dining". A second cause for false positives is the **ambiguity of lexicon entries**. An entry may be polysemous by itself (e.g., "sheet" mostly referring to "bed sheet" in our context) or its association to a topic may be ambiguous (e.g., "room service"). In experimental setting 1, such mistakes account for 9.6% of false positives. A third cause that can be distinguished is **partial matches** as for instance in "If you want to sleep quietly, make sure you get a courtyard or pool room.". Here, the keyword "pool" is extracted instead of matching the correct term "pool room". The sentence is erroneously associated with the topic "recreation". These kind of errors sum up to 5.5%. With regard to experimental setting 2 ("polar sentences"), we observe the same tendency. However, the share of errors due to context awareness is slightly lower with 81.5%. The other numbers are 15.2% for false positives

due to ambiguity and 3.3% for partial matches.

We perform a similar analysis for false negatives. We randomly sample a set of 300 instances which stem from a combined set of false negatives from both application domains. In this case, we only consider the setting with polar sentences (recall between both experimental settings does not deviate much). We can distinguish four main types of errors: The major share of false negatives is due to **implicit mentions** of the topic, mostly by **paraphrasing** it. For instance, by uttering "I agree, it is nearly impossible to break this camera.", the reviewer implies a positive evaluation of the aspect "built quality". Or with the expression "Great camera to put in pocket and have at all times.", he addresses the topic "dimensions". We observe that 57.5% of all false negatives stem from this type of implicit mention. Another form of implicit mention, which is not covered by our lexicon, is the use of **adjectives and verbs** that indicate a product aspect mention (e.g., "The camera is fast." –> "speed" or "Zooming more than about 3x or 4x yields nothing but blurry pictures" –> "zoom"). We find that this type accounts for 23.8% of errors. In total, implicit mentions (either by paraphrases or adjectives/verbs) are responsible for 81.3% of all false negatives.

The third and fourth type of errors are more closely related to the actual coverage of the lexicon. We distinguish errors due to a **missing lexicon entry** (e.g., "The dynamic range mode is very effective in avoiding blown-out highlights.") or due to a **missing variant** (e.g., misspellings: "The panaroma mode is the one new to me and it is good."). The former error type accounts for 14.0%, whereas the latter sums up to 4.7%. From these results, we can basically conclude that to improve the recall, it is most important that a system must cope with implicit mentions of product aspects. Including the set of most significant and frequent phrases (e.g., "put in pocket", "easy to operate" or "is fast") in addition to nominal mentions, promises to increase the recall of a simple lexicon-based approach.

The results show further that there exist significant deviations with regard to the accuracy of the different classes. For the hotel dataset (polar sentences), the best results are obtained for the class/aspect "parking" with an f-measure of 97.0%, compared 18.6% for the aspect "cleanliness" with the worst results. Concerning the digital camera dataset, the deviation is similar with an f-measure of 92.5% ("software"), compared to 18.4% ("built quality"). It is obvious that some kind of topics are relatively easy to detect with a lexicon-based approach, whereas others raise difficulties. Taking a closer look, we generally find that topics which are related to a very concrete aspect (e.g., "software" or "parking") receive better results than very abstract topics such as "cleanliness" or "speed". Due to the restriction of the lexicon to mostly nominal product aspect mentions, this observation is not really astonishing. Abstract topics are more often referred to by paraphrases and other implicit mentions: The low f-measure obtained for these classes is in nearly all cases due to a very low recall. For instance the classes "security", "noise", or "low-light performance" exhibit a perfect or nearly perfect precision, but a very low sensitivity between 14.3% and 29.7%. Taking a look at the false negatives produced by these classes also shows that such abstract topics are mostly referred to in an implicit manner. In consequence, we draw the same conclusion as indicated in the previous paragraph.

On a meta level, when comparing the results obtained for both corpora (hotel vs. camera), a further implication is that our main results and conclusions are consistent over different product domains (strengthening the validity of the results). The overall macro and micro-averaged results differ only slightly ($\leq$ 2% for micro and macro-averaged f-measure) and also other tendencies observed in the data behave similarly.

8.2.3. Related Work

Bloom et al. [45] examine the problem of extracting "appraisal expressions" and linking these to a set of predefined target entities. They **manually create taxonomies** of relevant target types for two domains, namely movie reviews and (generic) product reviews. The complexity of both taxonomies is relatively low. Basically, lists of terms are associated with generic concepts such as "product part",

"experience", "company", "marketing", or "support". The authors do not motivate their choice of concepts and do not provide any information regarding the manual process of creating the lexical resources. It is unclear which types of terms are covered, neither is the size of the resulting lexicons known. Evaluation is performed by manually inspecting a small random sample (160 tuples for each domain) of the output of their system. Despite the small evaluation set, no statement about the recall of their system can be made.

Cadilhac et al. [61] design an ontology for the domain of restaurant reviews (in French). With regard to the different levels of complexity (see Buitelaar and Magnini [60]), their knowledge base can be ascribed to the highest level, modeling a concept hierarchy and relations between them. They apply the restaurant ontology in an expression level extraction task and argue that the semantic information can be used to improve the tasks of aspect detection and of linking aspects to sentiment expressions. They further propose to use the ontology for summarizing the results of the review mining process. Their actual ontology is composed of 239 hierarchically organized concepts, which are associated with 646 instances (i.e., terms representing the concepts) and 36 different relations. Evaluation is performed on a very small hand-annotated dataset of French restaurant reviews (4000 words only) by comparing their **manually crafted ontology** with two automatic approaches which extract lists of aspects [177, 304]. Their experiments show slightly higher precision and recall values for their handcrafted knowledge base. They argue that the improved recall is due to the fact that relations are used to identify implicit mentions of product aspects. However, they do not provide any numbers regarding the significance of their findings, which (concerning the small evaluation set) would have been of importance. They neither provide any information about the employed annotation scheme and annotation process, which eventually renders the validity of their results questionable.

Streibel and Mochol [366] propose an **ontology for mining trends** in the context of market research. Their ontology evolves around abstract categories such as "product quality", "service", or "image", which are most relevant in market research. Each category is implemented by a set of keywords that may be organized in synonym groups. Their generic ontology is implemented (in German) with the advice from domain experts in market research. Unfortunately, they neither provide information about the size of their knowledge base, nor do they evaluate the proposed system.

Cheng and Xu [75] use ontologies for topic extraction and polarity analysis in reviews of the automotive domain. They describe an approach to **merge existing ontologies to a larger knowledge base**, but do not explicitly evaluate the performance or effects of this step. A manually created ontology, which basically represents a concept hierarchy, is automatically extended by means of heuristically extracting related terms from a domain-specific lexicon. Whereas in the original ontology 363 concepts are covered by 1,233 terms, the enrichment process adds 9,033 lexical representations. Evaluation is performed by measuring the effectiveness of their ontology for the task of topic extraction. Within an evaluation corpus of 1,000 sentences (extracted from customer reviews), a gold standard of 2,038 terms is manually annotated. Again, no information about the annotation process is provided. It is also unclear how they define a correct match (i.e., a true positive) — it may be a correctly identified topic or a lexical match. Comparison of their approach to a generic terminology extraction system as well as an opinion mining system [304], shows significantly better results for both, precision (94.44%) and recall (89.35%). They find that the enrichment process achieves a major boost in recall (+70 percentage points) and at the same time increases the precision (+6 percentage points). However, due to the small dataset, the unclear annotation process, and the imprecise definition of correct matches, it is difficult to assess the validity of the results.

Similar to Cheng and Xu [75], also Carenini et al. [64] present an approach to automatically enrich an existing knowledge base. Given an existing concept hierarchy, they apply term similarity heuristics to map automatically extracted product aspects to one of the concepts of the hierarchy. Employed similarity measures[6] are exclusively based on the semantic distance of words in WordNet [263] and

[6] A subset of the measures proposed in [59] is applied.

the method for aspect extraction equals the approach presented in [177]. The final, enriched **concept hierarchy** is obtained after manual refinement. As in [75], the process of enrichment only adds new lexical items, but no new concepts or relations are detected. Evaluation is performed only for the mapping algorithm, but not for the actual task of extracting product aspects from text. Results retrieved by the automatic method are compared with gold standard mappings in two different domains (digital camera and DVD reviews). Both gold standards are rather small, containing not more than 116 mapped terms.

Further works which study the applicability of pre-built lexicons or semi-automatic approaches to lexicon creation are for example found in [136, 464, 468]

8.3. Supervised, Machine Learning Approach

In this section, we examine the use of supervised machine learning approaches for the discovery of coarse-grained product aspects in customer reviews. We basically cast the task as an instance of a text categorization problem and train *maximum entropy classifiers* on our sentence level corpora. In this context, we are primarily interested in answering the following three questions:

- Our first goal is to generally compare the results achievable with supervised classifier methods to the relatively simple, purely lexicon-based approach we discussed in the previous section. As supervised methods require a tedious and costly manual annotation of training data, the question is whether the induced effort leads to significantly better results.
- When learning feature-based classifiers, such as maximum entropy models, it is not directly clear which set of features serves best for the task at hand. Our aim is to answer the question which feature set fits the specific task of attributing sentences to coarse-grained product aspects. In this context we also examine a hybrid approach, combining supervised classifiers with a lexicon-based method. We incorporate the lexicon information as feature to the maximum entropy model.
- As already mentioned, creating training data for supervised methods is typically very costly. We are therefore interested in the question of whether we can successfully exploit weakly labeled data to reduce these costs in our specific scenario. In particular, we identified that section headings provided in customer reviews often correspond very well to our coarse-grained concepts of product aspects. We employ high-precision heuristics to extract related sentences that we can use as training samples for supervised classification. We cover this approach in the separate Section 8.4.

8.3.1. Implementation

In this section we describe our method of implementing the hierarchical multi-label classification task with supervised machine learning techniques. First, we transform the original problem so that we can make use of standard classification algorithms. We briefly discuss this transformation process and explain why we follow this path, instead of applying dedicated, more complex algorithms to multi-label classification. Second, we describe the different sets of features we experiment with and explain how we incorporate information from our lexicon as features to the maximum entropy classifiers.

Problem Transformation

Figure 8.3 gives an overview of the complete transformation process. First, we flatten the concept hierarchy and relabel the original corpus. In a second step, we transform the original multi-label problem to multiple binary classification problems and create corresponding training corpora. Based

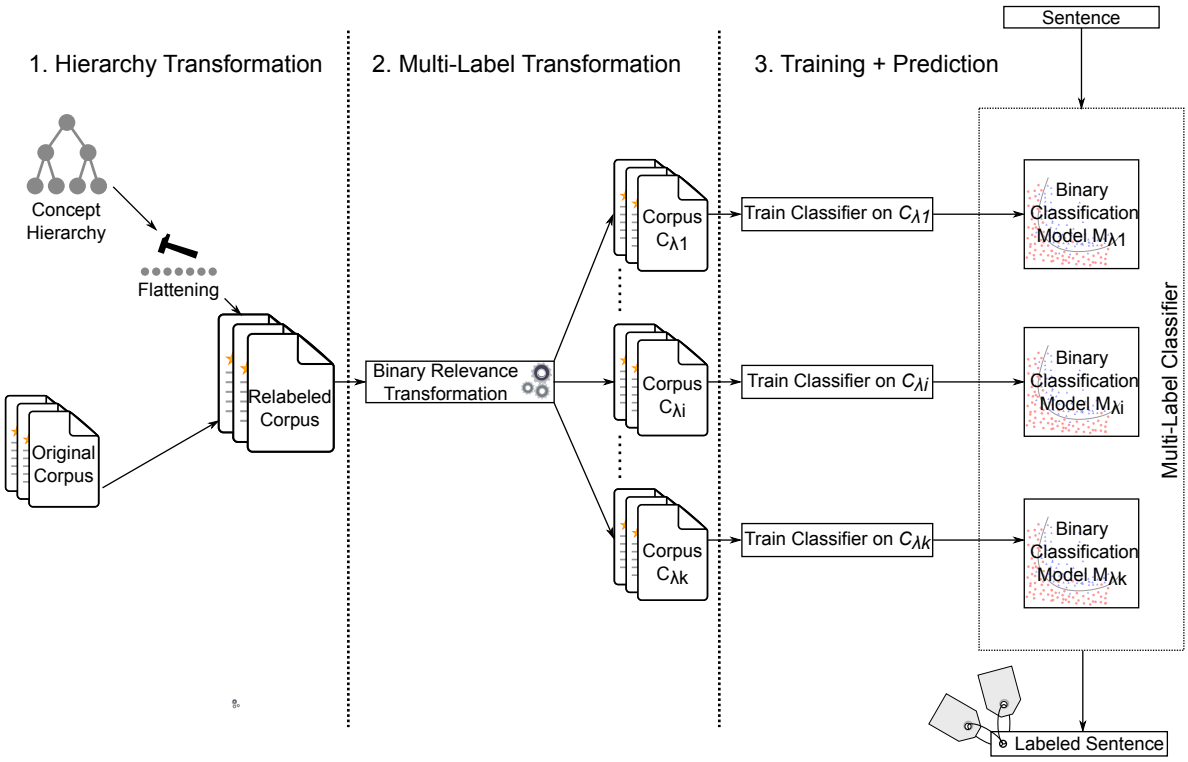


Figure 8.3.: Implementing hierarchical multi-label classification by an ensemble of multiple binary classifiers.

on these corpora, we train a set of binary classifiers that (in combination) allow to label sentences in accordance to our model.

Hierarchy Flattening Although incorporating structural information, such as provided by our concept hierarchy, may improve the performance of classification algorithms [62, 106, 122, 217, 253, 326, 404], we do not intend to make use of it in our classification models. We simply flatten the hierarchy as described in Appendix E.2 and relabel the corpus accordingly. We then use the relabeled corpus to learn the desired classification models. It is important to remark that, although we withhold the hierarchical information (i.e., dependencies between categories) from the actual learning algorithm, the information is still available within an application (e.g., for visualization purposes). In summary, we are aware that the overall classification performance may deteriorate when disregarding the hierarchy information. But as we will see, also this simpler approach achieves very good results and suffices to study the questions defined earlier.

Binary Relevance Transformation Many supervised machine learning algorithms are not directly applicable for multi-label classification tasks. To cope with this problem, typically one of the following approaches is utilized: Either the original multi-label problem is transformed to one or more single-label problems (solvable with traditional learning algorithms), or a specific learning algorithm is adapted to directly deal with multi-label instances. For our experiments, we opt for the relatively simple problem transformation approach as it suffices for the purposes defined earlier. Further, the approach has the major advantage of being independent of any specific machine learning algorithm. After transformation, any of the multitude of well-known single-label learning procedures for text categorization can be applied.

An overview of different problem transformation methods is provided by Tsoumakas and Katakis

[388]. The methods basically differ with respect to the loss of information coming along with the transformation process. Most naively, and with the greatest loss, we could either simply discard all multi-label instances or randomly select single labels from all multi-label instances — clearly both approaches are not desired. The most commonly used techniques are *label power set* and *binary relevance* transformation. The label *power set method*[7] creates a new, artificial label for each distinct subset of labels found in the data. For example, if we find instances which are multi-labeled as {A, B, C} in the data, we create an artificial label "A+B+C". The advantage of such a transformation is that it preserves all information on the correlations and interdependences between labels. However, the major drawback is the exponential increase of the label set's cardinality and consequently a data sparsity problem. As the cardinality of our label sets for both domains is relatively high compared to the size of our datasets, the binary relevance transformation better fits our needs:

Let $k = |\mathcal{L}|$ be the number of distinct labels associated with a given corpus C. Then for each label $\lambda \in \mathcal{L}$, we construct a "binary" corpus C_λ containing all the instances of the original corpus, with each instance of the newly created corpus being either labeled as λ or $\neg\lambda$. We label an instance as λ if the original label set contains λ, otherwise we set the label to $\neg\lambda$. This transformation results in k different binary corpora $C_{\lambda_1}, \ldots, C_{\lambda_k}$ and on each corpus C_λ, we train a binary classification model $M_\lambda : \mathcal{X} \rightarrow \{\lambda, \neg\lambda\}$. In other words, each of the k classification models is able to predict the presence or absence of one particular label λ. For multi-label classification, we use the set of learned classifiers as an ensemble. To (multi-label) classify a new instance $x \in \mathcal{X}$, we apply all classifiers $M_{\lambda_1} \ldots M_{\lambda_k}$ of the ensemble and output as a label set the union of all positive labels that are output by the individual classifiers. If all classifiers output the negative label $\neg\lambda$, we consider the instance as belonging to a class "other" or, in our specific case, as being "off-topic". For example, with regard to the hotel review domain, we may train (amongst others) two distinct binary classifiers that can distinguish between aspects "service" and "not service", as well as "location" and "not location", respectively. If both classifiers predict positively, we would label a sentence as being related to both aspects, "service" and "location". If both classifiers predict negatively, we would consider the sentence as being off topic.

The advantage of the described binary relevance approach is (besides its simplicity) that it avoids the data sparsity problem discussed earlier. On the other hand, as the transformation reduces the available information to the presence or absence of a particular label, while disregarding any other, classifiers are unable to learn correlations and interdependencies between different classes (e.g., that the aspect "cleanliness" may be related to the aspect "bathroom"). Also the computational overhead is increased. For each prediction all k classifiers need to be consulted. We are aware of these disadvantages. We may achieve better results with more complex approaches to multi-label classification, such as [74, 81, 115, 147, 254, 333, 461, 462]. However, to answer the questions we have formulated in the introductory part, it is irrelevant whether the absolute results might be improved by, for example, 2 percentage points in f-measure.

Feature Engineering

Since we reduced the task of identifying coarse-grained product aspects to a traditional text categorization problem, it is intuitive to employ the standard set of features as proposed in the relevant literature. However, in contrast to the classic setting (e.g., the Reuters RCV1 evaluation corpus [227]) where the task is to classify whole documents, we apply text categorization at the **sentence level**. Due to the comparably small number of words within a sentence, categorization at the sentence level may differ substantially from document-level classification. Bag of words feature vectors will be even "sparser" than in document categorization, potentially lowering the classification performance. It it thus not directly clear whether previous results with regard to feature selection are transferable to our specific setting: Although not particularly consistent, the tendency in text categorization literature is

[7] in analogy to the power set construction method known from automata theory

mostly that other features, apart from the simple "bag of words" approach, do not lead to significant improvements or even deteriorate performance [31, 65, 107, 228, 267, 338, 378].

For our experiments, we distinguish two classes of feature types: "Basic features" and "knowledge-rich features". With basic features we refer to features derived from the lexical information (i.e., the words/tokens) and shallow linguistic preprocessing (e.g., lemmatization, part-of-speech tagging). These feature types have in common that they can be derived in an automated manner and do not involve any manual intervention. In contrast, knowledge-rich features make use of a manually curated knowledge base[8]. In our case, these types of features are all based on the information encoded in the product type taxonomy lexicons we presented earlier.

We consider the following preprocessing steps and basic feature types for representing a sentence:

- **Bag of words** (BOW): We tokenize the sentence (with the Stanford CoreNLP tokenizer) and create a single feature for each unique token. We use a standard stop word list for filtering irrelevant tokens and do not case fold tokens. The feature value represents the term frequency of a token within the sentence. We use the BOW representation as our baseline.
- **Downcase** (DOW): As BOW, but we lowercase each token.
- **Lemmatization** (LEM): As BOW, but we reduce each token to its lemma (e.g., "eating", "ate" –> "eat"). Take note that with lemmatization we also lowercase each token. The intuition with this representation is to lower the overall number of different features, while preserving the topic relevant semantics of a token.
- **Part-of-speech tags** (POS): As a further shallow linguistic feature, we consider the part-of-speech of a token (tagging is based on the Stanford CoreNLP tagger). To do so, we simply concatenate token and POS tag (e.g., "eat_VB"). We only use the first two letters of the *PennTreebank encoding* [251] — for instance, we do not distinguish between the tags "VBN" (verb: past tense) and "VBG" (verb: present participle), but simply encode the information that the token is a verb ("VB"). In general, part-of-speech tags can be helpful for a very simple word sense disambiguation.
- **Token n-grams** (NGR): Instead of creating features from single tokens (unigrams) only, we may derive features from token n-grams (i.e., combinations of adjoining tokens). The intuition is that capturing longer "phrases" may encode more precise semantic information. For example, the bigram "room service" may be a better indicator for the topic "dining" than the two unigrams "room" and "service" considered separately. If we use bigrams, we abbreviate with NGR-2; if we combine unigrams and bigrams, the notation is NGR-1+2.

The following enumeration summarizes the knowledge-rich feature types we examine:

- **Lexicon term match** (LEX-MAT): We create a separate feature for each term/phrase of a sentence that matches a lexicon entry. The features encode the canonical form of the matching token sequence and the fact that a lexicon match occurred. For example, if the sequence "air-conditioning unit" matches, we create a feature "LEX-MAT:air conditioning unit". As with token n-grams, the idea is that longer phrases may be semantically richer than isolated words. Further, such a feature represents the information that a term has been identified by a human as being important for a specific topic.
- **Lexicon sentence label** (LEX-LAB): We use the lexicon-based classifier to label the sentence — that is, for each lexicon match, we look up the corresponding topic. For each unique label discovered, we create a feature of the form "LEX-LAB:<label>" (e.g., "LEX-LAB:room amenities").

[8] Take note that only the step of creating the knowledge bases is manual, the actual extraction of knowledge-rich features is of course fully automated.

The feature value reflects the number of matches that correspond to the particular label (within the sentence). The idea is to add the vote/prediction of the lexicon-based classifier to the system.

- **Lexicon context labels** (LEX-CTX1): Instead of considering the lexicon-based predictions for the current sentence only, we may include the predictions for sentences in the immediate context. In particular, we add the labels of the immediately preceding and the immediately following sentence. Our intuition is that topics addressed in adjacent sentences are correlated. Knowing the topic of the previous and following sentences may help for classifying the current sentence.
- **Lexicon context term matches** (LEX-CTX2): We look at the preceding and following sentence and add each lexicon match in these adjacent sentences as a feature for the current sentence. As for the LEX-MAT feature, we use the canonical form of the matching lexicon entries.

8.3.2. Experiments and Results

Experimental Setup

For our experiments, we use the multinomial logistic regression classifier (maximum entropy classifier) of the LingPipe text processing tool kit[9]. For regularization/smoothing we use a Gaussian prior distribution with a fixed variance. To estimate the regression coefficients, the implementation uses a specific form of *stochastic gradient descent* [66]. The following table summarizes the parameter settings for the MaxEnt classifier[10]:

parameter	value	description
prior variance (regularization)	8.0	the fixed variance of the Gaussian prior
simulated annealing type (optimization)	exponential	the type of annealing schedule which defines the learning rate for numerical optimization
initialLearningRate (optimization)	0.002	minimum relative improvement in error during an epoch to stop search
base (optimization)	0.9975	base of the exponential decay
minImprovement (optimization)	0.000001	minimum relative improvement in error during an epoch to stop search
minEpochs (optimization)	100	minimum number of search epochs
maxEpochs (optimization)	1000	maximum number of search epochs

Table 8.3.: Parameter settings for the MaxEnt classifiers.

We do not perform any optimization of the hyperparameters for classification — for example, the Gaussian prior is fixed to a prior variance of 8.0 for each classifier we train. Thus, there is no need for a separate development dataset. We use the complete corpus for validation. To prevent the overestimation of accuracy and to guarantee generalizability of the reported results to independent datasets, we use 10-fold cross validation. In particular, all reported results are based on 10 rounds of 10-fold cross validation. 10-fold cross validation randomly splits the original corpus into 10 distinct folds. The classifier is trained on 9 of the 10 folds and is tested on the single remaining fold. The folds are "rotated" 10 times so that every observation is used for both, training and testing. The 10 results obtained with every rotation are averaged to produce a single estimate. To further cope with the variability of results and to compare different settings (e.g., different feature sets) with statistical significance, we repeat 10-fold cross validation ten times for each setting. Naturally, each of the ten rounds is based on a unique partitioning of the original corpus. To make results obtained for different settings comparable, the partitioning is based on a pseudo-random permutation of the corpus with a manually

[9] http://alias-i.com/lingpipe/demos/tutorial/logistic-regression/read-me.html
[10] Some of the descriptions haven been adopted from the LingPipe documentation.

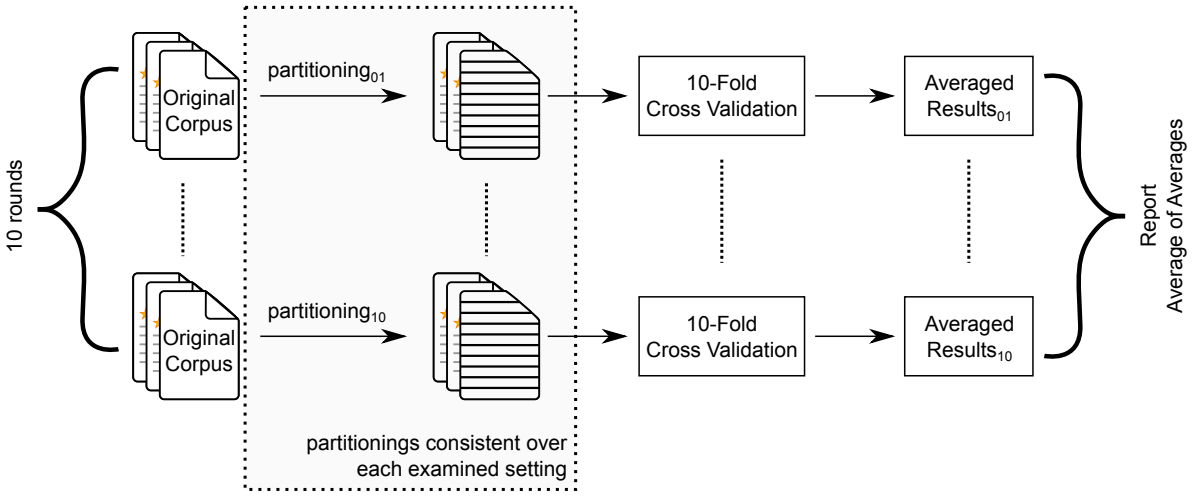


Figure 8.4.: Repeated 10-fold cross validation. To guarantee comparability of different classifier configurations, the partitioning is consistent for all experiments.

defined seed. That is, the ten distinct partitionings for each round of 10-fold cross validation are the same for each considered setting. Figure 8.4 illustrates this experimental setup.

Feature Engineering

We first discuss our results with regard to the comparison of different feature sets. Table 8.4 shows the results for the basic feature types. We report the micro and macro-averaged f-measure for relevant combinations of feature sets, while distinguishing the experimental settings "whole corpus" vs. "polar sentences only". All differences (shown in brackets) are computed in reference to the baseline feature set (bag-of-words). Combinations of feature types are indicated by a "+" symbol. The tables are ordered ascendingly by micro-averaged f-measure as observed for the "all-sentence" corpus (baseline is fixed).

There exist two main results we can consistently read from the data: First, considering the absolute numbers, the maximum increase in micro/macro-averaged f-measure that we achieve by extending the baseline feature set is moderate at three to four percentage points. Second, besides case folding and lemmatization, other linguistic preprocessing or token n-gram representations do not improve results. Differences between feature sets with and without adding part-of-speech tags or token 1+2-grams are marginal and, when comparing pairwise, mostly not significant (pairwise t-test at 99% confidence level, measures not shown here). Lowercasing the tokens achieves an improvement of around 1.5 to 2.0 percentage points compared to the baseline. With lemmatization we observe a further increase of around 1.5 to 2.0 percentage points. All results are consistent over domains (hotel vs. camera), corpus subsets (all vs. polar), and different approaches to averaging (micro vs. macro). In summary, we find that our results for sentence level topic categorization confirm the earlier mentioned tendency we found in the literature on document-level classification: Incorporating more sophisticated linguistic preprocessing and longer phrases does not improve results. We may hypothesize (albeit, without further examination) that the improvements by lowercasing and lemmatization actually stem from relative data sparseness. Reducing the number of features by mapping tokens to a "canonical" form, observably helps in our case. But in general (given enough training data), capitalization and different inflected forms may in fact be identified as helpful cues for classifying text. However, as in our case, creating "enough" training data is very costly. Also, the drastic deterioration in performance when using lemma bigrams only (around -25 percentage points) is likely to be related to data sparseness (the majority of bigram features occurs only once in the training data).

	all		polar	
features	**micro-f**	**macro-f**	**micro-f**	**macro-f**
BOW (baseline)	0.754	0.714	0.789	0.744
NGR-2+LEM	0.540 (-0.214**)	0.464 (-0.250**)	0.553 (-0.236**)	0.466 (-0.279**)
BOW+DOW	0.768 (+0.015**)	0.736 (+0.022**)	0.804 (+0.014**)	0.768 (+0.024**)
NGR-1+2+LEM+POS	0.783 (+0.029**)	0.750 (+0.036**)	0.817 (+0.028**)	0.778 (+0.034**)
NGR-1+2+LEM	0.783 (+0.029**)	0.752 (+0.038**)	0.819 (+0.030**)	0.777 (+0.033**)
BOW+LEM+POS	0.785 (+0.031**)	0.754 (+0.039**)	0.817 (+0.028**)	0.778 (+0.034**)
BOW+LEM	0.785 (+0.032**)	0.756 (+0.041**)	0.817 (+0.028**)	0.778 (+0.034**)

(a) hotel corpus

	all		polar	
features	**micro-f**	**macro-f**	**micro-f**	**macro-f**
BOW (baseline)	0.715	0.698	0.734	0.705
NGR-2+LEM	0.517 (-0.198**)	0.475 (-0.222**)	0.525 (-0.209**)	0.465 (-0.241**)
BOW+DOW	0.729 (+0.015**)	0.714 (+0.016**)	0.753 (+0.019**)	0.725 (+0.019**)
BOW+LEM+POS	0.734 (+0.019**)	0.723 (+0.026**)	0.759 (+0.024**)	0.737 (+0.032**)
NGR-1+2+LEM+POS	0.736 (+0.021**)	0.722 (+0.024**)	0.763 (+0.029**)	0.734 (+0.029**)
BOW+LEM	0.738 (+0.024**)	0.729 (+0.031**)	0.765 (+0.031**)	0.746 (+0.040**)
NGR-1+2+LEM	0.746 (+0.031**)	0.729 (+0.032**)	0.769 (+0.035**)	0.742 (+0.037**)

(b) digital camera corpus

Table 8.4.: Comparison of results for different sets of basic features. Differences to the baseline (bag-of-words) are shown in brackets. All differences are significant at the 99% confidence level.

We now consider the results obtained when introducing knowledge-rich features from the lexicon-based classifier. Table 8.5 summarizes our findings in this regard. As a baseline we choose the best basic feature set (BOW+LEM) and, again, all differences are calculated in reference to the baseline. We first experimented with all lexicon-based features (MAT, CTX-1/2, and LAB) in isolation. We find that including the information about the context does not lead to any improvement. In fact, we consistently observe that the performance slightly decreases. In case of the lexicon-based labeling, we must consider that provided context knowledge is not perfect — the lexicon-based labeling has an average precision of around 70%-75%. To eliminate the influence of partly incorrect information, we also experimented with including (synthetically perfect) context information from the gold standard (GOLD-CTX). But also when allowing the classifier access to gold standard annotations, the context features lead to inferior results. We thus conclude that our initial hypothesis on knowing about the topic of adjacent sentences may help in labeling the current sentence, cannot be confirmed. On the other hand, features based on lexicon matches (MAT) and the actual lexicon labeling (LAB) do help. Including information about matching terms increases the averaged f-measure around 1.0 to 1.5 percentage points. Including the labels has a stronger effect, with improvements between 2.5 and 5.0 percentage points. Combining both feature types (LEX-MAT+LAB) shows only very marginal gains compared to using the label only (LEX-LAB). In fact, the differences measured for the hotel corpus are not significant at a 99% confidence level (for the camera corpus they are). We also tested a combination of all lexicon feature types, but no significant deviation (neither positive, nor negative) is observed. In summary, we find that following a hybrid approach, by including lexicon information as features to a supervised classifier, is superior to using the supervised classifier alone. We observe improvements of up to 5.5 percentage points in macro-averaged f-measure.

	all		polar	
features	**micro-f**	**macro-f**	**micro-f**	**macro-f**
BOW+LEM (baseline)	0.785	0.756	0.817	0.778
baseline+LEX-CTX2	0.781 (-0.004**)	0.751 (-0.005**)	0.816 (-0.001)	0.775 (-0.003)
baseline+LEX-CTX1	0.782 (-0.003**)	0.755 (-0.001)	0.815 (-0.002**)	0.776 (-0.002)
baseline+GOLD-CTX	0.784 (-0.001)	0.743 (-0.013**)	0.815 (-0.002**)	0.773 (-0.006**)
baseline+LEX-MAT	0.792 (+0.007**)	0.765 (+0.010**)	0.825 (+0.008**)	0.794 (+0.016**)
baseline+LEX-LAB	0.809 (+0.024**)	0.801 (+0.045**)	0.843 (+0.026**)	0.829 (+0.050**)
baseline+LEX-MAT+LAB	0.810 (+0.025**)	0.801 (+0.045**)	0.846 (+0.028**)	0.831 (+0.052**)
baseline+LEX-ALL	0.812 (+0.027**)	0.801 (+0.045**)	0.846 (+0.028**)	0.829 (+0.050**)

(a) hotel corpus

	all		polar	
features	**micro-f**	**macro-f**	**micro-f**	**macro-f**
BOW+LEM (baseline)	0.738	0.729	0.765	0.746
baseline+LEX-CTX2	0.734 (-0.005**)	0.722 (-0.007**)	0.760 (-0.005**)	0.738 (-0.008**)
baseline+LEX-CTX1	0.735 (-0.003**)	0.722 (-0.007**)	0.759 (-0.007**)	0.734 (-0.011**)
baseline+GOLD-CTX	0.736 (-0.002)	0.729 (-0.000)	0.767 (+0.002)	0.742 (-0.003)
baseline+LEX-MAT	0.752 (+0.014**)	0.741 (+0.012**)	0.778 (+0.013**)	0.759 (+0.013**)
baseline+LEX-LAB	0.772 (+0.033**)	0.770 (+0.040**)	0.803 (+0.038**)	0.796 (+0.050**)
baseline+LEX-ALL	0.777 (+0.038**)	0.772 (+0.043**)	0.807 (+0.042**)	0.799 (+0.053**)
baseline+LEX-MAT+LAB	0.778 (+0.039**)	0.774 (+0.045**)	0.809 (+0.044**)	0.801 (+0.055**)

(b) digital camera corpus

Table 8.5.: Comparison of results for different sets of knowledge-rich features. Differences to the baseline (bag-of-words + lemma + POS tags) are shown in brackets. All differences are significant at the 99% confidence level.

Comparison to Purely Lexicon-Based Approach

Also without further optimization (no parameter tuning, no *feature selection*[11], and no dedicated classification algorithm for hierarchical multi-label data), we observe that the approach based on supervised classification shows significant improvements in comparison to the purely lexicon-based method. Tables 8.6a and 8.6b illustrate this comparison in numbers. The tables report the f-measure achieved with the best performing maximum entropy classifier (BOW+LEM+LEX-MAT+LAB) for each coarse-grained product aspect. The numbers in brackets represent the difference to the lexicon classifier (cf., Table 8.2).

For the hotel corpus we find that the supervised approach increases the f-measure by around 9 percentage points (micro-averaged) and around 13.5 percentage points with respect to macro-averaged results. We further observe that this increase is consistent for the settings with all and only polar sentences, raising the validity of the results. For the digital camera dataset, we have similar results. Here, the increase is slightly lower with 7-8 percentage points in micro-averaged f-measure and around 11 percentage points in macro-averaged f-measure. In general, we find that improvements in terms of macro-averages are higher compared to looking at the micro-averages. This is the case when small categories (i.e., with fewer instances) benefit disproportionately in comparison to large categories. In fact, for the hotel dataset we observe major improvements of over 30 percentage points for the relatively small categories "cleanliness", "security", and "noise". For the camera corpus such topics are for instance "built quality", "low-light performance", and "dimensions".

Recall that the lexicon-based approach particularly exhibits inferior performance with regard to more "abstract" topics (e.g., "sleep quality", or "noise"), compared to topics which cover many concrete

[11] See for example Manning et al. [250, chap. 13.5]

aspect/label	f-measure (all)	f-measure (polar)
recreation	0.939 (+0.004)	0.941 (-0.008)
view	0.905 (+0.005)	0.941 (-0.006)
internet	0.893 (+0.004)	0.915 (+0.001)
parking	0.886 (-0.020)	0.959 (-0.011)
breakfast	0.881 (+0.028)	0.938 (-0.000)
bathroom	0.863 (+0.031)	0.917 (+0.034)
cleanliness	0.852 (+0.676)	0.876 (+0.690)
location	0.842 (+0.214)	0.887 (+0.190)
bed	0.826 (+0.005)	0.850 (-0.007)
elevator	0.817 (+0.090)	0.849 (+0.107)
service	0.816 (+0.025)	0.879 (+0.047)
noise	0.809 (+0.348)	0.840 (+0.384)
room	0.807 (+0.063)	0.836 (+0.034)
sleep quality	0.805 (+0.125)	0.838 (+0.140)
dining	0.794 (+0.049)	0.824 (+0.035)
check in-out	0.777 (+0.053)	0.791 (+0.038)
facility	0.727 (+0.041)	0.749 (+0.038)
price	0.719 (+0.080)	0.729 (+0.074)
decoration	0.719 (+0.059)	0.710 (+0.043)
room amenities	0.710 (+0.052)	0.668 (+0.018)
air conditioning	0.677 (+0.044)	0.700 (+0.079)
security	0.563 (+0.313)	0.641 (+0.345)
micro-average	0.810 (+0.091)	0.846 (+0.089)
macro-average	0.801 (+0.134)	0.831 (+0.135)

(a) hotel corpus

aspect/label	f-measure (all)	f-measure (polar)
software	0.905 (+0.012)	0.929 (+0.004)
battery	0.898 (+0.015)	0.907 (+0.008)
video recording	0.866 (-0.008)	0.920 (-0.000)
focusing	0.866 (+0.027)	0.887 (+0.026)
zoom	0.854 (-0.023)	0.885 (-0.003)
price	0.850 (+0.064)	0.876 (+0.059)
optics	0.849 (+0.006)	0.875 (+0.003)
memory	0.816 (+0.001)	0.779 (-0.029)
screen	0.812 (-0.010)	0.841 (-0.019)
dimensions	0.794 (+0.290)	0.828 (+0.326)
ease of use	0.783 (+0.052)	0.826 (+0.047)
flash	0.779 (+0.014)	0.821 (+0.032)
shooting modes	0.777 (+0.017)	0.780 (+0.002)
user manual	0.773 (-0.009)	0.827 (-0.004)
settings	0.769 (+0.069)	0.751 (+0.035)
user interface	0.745 (+0.038)	0.793 (+0.022)
picture quality	0.735 (+0.150)	0.800 (+0.093)
features	0.725 (+0.005)	0.775 (+0.026)
appearance	0.683 (+0.215)	0.719 (+0.247)
speed	0.678 (+0.156)	0.673 (+0.153)
low-light performance	0.672 (+0.356)	0.669 (+0.419)
connectivity	0.670 (+0.097)	0.738 (+0.138)
built quality	0.510 (+0.317)	0.516 (+0.331)
micro-average	0.778 (+0.078)	0.809 (+0.069)
macro-average	0.774 (+0.109)	0.801 (+0.113)

(b) digital camera corpus

Table 8.6.: Comparison of results for maximum entropy versus lexicon-based classification. The MaxEnt classifier is based on the best performing feature set (including knowledge-rich features), but without any further optimization of hyperparameters.

entities (e.g., topic "bathroom"). In the former case, the sensitivity of the lexicon-based method is typically very low. We find that especially this type of topics benefits from the supervised approach. For example, we see improvements of nearly 70 percentage points for the topic "cleanliness" or over 40 percentage points for the aspect "low-light performance". The recall (not shown in the table) for these categories increases drastically. In Table 8.7 we take a closer look at the most important features for three of these "difficult" topics. In particular, the tables show excerpts of the estimated maximum entropy models[12] for the aspects "hotel:noise", "hotel:cleanliness", and "camera:built quality". Our intention to present this exemplary data is to point out the importance of some tokens/terms that are very hard to integrate in a lexicon-based approach. For instance, with regard to the topic "noise", besides observing very obvious features, such as "quiet", "noisy", "noise", or "hear", we also find tokens such as "sleep", "night", "quite", "wall", or "thin". We can use this example to show that a supervised classifier exhibits superior performance (compared to a lexicon-based approach) mainly for three reasons: First, a lexicon that is based on mostly nominal mention types does not include important terms such as "quiet", "noisy", or "hear". Second, even for nominal terms, it may be very hard for a human to correctly associate them to a topic. Nouns such as "sleep", "night", or "wall" cannot directly be identified as being relevant for the topic "noise". Third, the classifier is able to learn weights for each term and to calculate a probability based on the interplay of occurring terms. With regard to the lexicon-based classifier, the decision is simply based on the presence or absence of indicator terms. To point out some more interesting examples: With respect to the aspect "cleanliness" we find (at a first sight) surprising features such as "well", "would", or "pet friendly". The token "well" seems to be a good indicator because it is often used within the phrase "well kept" (see corresponding bigram feature), whereas the word "would" is apparently used in phrases such as "would need some cleaning" or "cleaning would help". With regard to "pet friendly", it seems that reviewers correlate the term to the "cleanliness" of a hotel (presumably with a negative connotation). For the topic "built

[12] For didactic reasons, we choose models based on the *NGR-1+2+LEM+LEX-MAT+LAB* feature set (i.e., including bigrams).

#	feature symbol	coefficient
1	quiet	3.750
2	noisy	2.612
3	noise	2.054
4	loud	1.943
5	hear	1.925
6	LEX-LAB:noise	1.877
7	sleep	1.314
8	LEX-MAT:noise	1.083
9	LEX-LAB:sleep quality	1.073
10	night	0.984
11	quite	0.890
12	wall	0.875
13	thin	0.860
14	problem	0.836
15	LEX-LAB:air conditioning	0.794
16	door	0.701
17	outside	0.665
18	could hear	0.659
19	all night	0.655
20	floor	0.653
21	street noise	0.620
22	siren	0.616
23	earplug	0.615
24	next	0.614
25	elevator	0.577

(a) hotel: "noise"

#	feature symbol	coefficient
1	clean	5.992
2	LEX-LAB:cleanliness	2.016
3	smell	1.184
4	mold	0.954
5	stain	0.925
6	LEX-MAT:cleanliness	0.794
7	cleanliness	0.794
8	dirty	0.793
9	very clean	0.625
10	room clean	0.614
11	well	0.578
12	sheet	0.527
13	would	0.515
14	bathroom	0.509
15	very well	0.508
16	hotel clean	0.504
17	old	0.439
18	black mold	0.416
19	smell like	0.411
20	housekeeping	0.381
21	well keep	0.376
22	maid	0.363
23	problem housekeeping	0.352
24	maid service	0.348
25	pet friendly	0.342

(b) hotel: "cleanliness"

#	feature symbol	coefficient
1	feel	1.458
2	build	1.190
3	durable	1.153
4	scratch	1.149
5	break	1.051
6	solid	0.971
7	drop	0.805
8	LEX-LAB:built quality	0.780
9	make	0.775
10	quality great	0.770
11	LEX-LAB:appearance	0.757
12	water	0.711
13	quality	0.660
14	already break	0.648
15	seem	0.634
16	still work	0.629
17	plastic	0.623
18	after	0.621
19	over	0.572
20	already	0.545
21	kid	0.533
22	case	0.529
23	cheaply make	0.527
24	cheaply	0.527
25	damage scratch	0.485

(c) aspect "built quality"

Table 8.7.: Top 25 features for three different MaxEnt classifiers. The coefficients represent the estimates of the maximum entropy model for the positive outcome in a binary classification task.

quality", we observe interesting indicators such as "water", "plastic", or "kid" (which we do not further comment on).

Mistake Analysis

In this section we examine the most common types of mistakes encountered with the supervised classification approach. Our primary intention is to point out the most promising paths for improvement. The concrete approach to mistake analysis is as follows: We pick out exemplary categories/topics for which we achieved either low precision or low recall and then analyze the main reasons for false positives and false negatives. We consider the results produced by the best performing MaxEnt classifiers with the BOW+LEM+LEX-MAT+LAB feature set. Representatives for comparably low precision are for instance the topics "flash" and "picture quality". On the "polar sentences" sub corpus we observe (nonetheless good) precision values[13] of 73.2% (76.7%) for these particular aspects (macro-averaged precision for the related corpora are at 81.3% and 87.3%, respectively). As representatives for low recall, we choose the topics "built quality" (recall: 48.5%) and "security" (recall: 56.1%).

We first take a closer look at correctly identified sentences and compare with false positives. A very basic observation is that true positives typically contain strong indicators (i.e., features with a high weight coefficient) which correctly identify the overarching topic. Further, in most cases multiple strong indicators exist in the context, while indicators for other categories are absent. For instance the sentence "Only complaint is that the flash coverage begins to weaken beyond about 10 feet.", contains the three strong indicators "flash", "coverage", and "weak", but no strong indicators for other topics. Such a type of sentence is nearly always correctly identified by the classifier. On the other hand, a common pattern for false positives is when a single strong indicator exists for the false class, while only weak indicators provide clues for the true class. For example, consider a sentence like "100+ photos, most with flash, and the camera was still going strong.". The binary classifier for the topic "flash" produces a false positive — the true class is "battery". We have the single strong feature "flash", but only very weak indicators for the true topic ("going" + "strong"). Without any further information

[13] not shown in the tables

from the context (e.g., "battery life" is mentioned in the previous sentence), the classifier consequently misconceives the occurrence of the strong indicator. So in such cases, the MaxEnt classifier is no better than the lexicon-based classifier which totally neglects context information. Here, the context is simply not indicative enough and admittedly such a sentence is also hard to categorize for a human. We may summarize this type of failure as "**weak context information**".

Another common case when lack of information leads to false positives, is when sentences are simply too short. For instance the sentence "Style and color is great." contains only the three tokens "style", "color", and "great" after stop word filtering. The true class for this sentence is "appearance", but the classifier falsely identifies it as being related to the topic "picture quality". The classifier learned that the feature "color" (in terms of color reproduction in pictures) is much more closely related to the topic "picture quality" than to "appearance". And in fact, it is a markedly stronger indicator for the positive class "picture quality" than the feature "style" is for the negative class "not picture quality". In the absence of further information ("great" also indicates "picture quality"), the classifier cannot take an accurate decision.

The previous example also points towards another characteristic that is the cause of reduced accuracy. We have seen that one reason for misclassification was the ambiguous token "color". It has a stronger correlation with the topic "picture quality" as it is more likely to appear in "picture quality sentences" than in "appearance sentences". But a second reason is also the considerably higher prior probability for the topic "picture quality" compared to the topic "appearance". Corpus analysis has already shown that the majority of comments within a customer review is related to only a few topics. In our case, reviewers much more often comment on the aspect "picture quality" than on the aspect "appearance". In consequence, the classifier is confronted with highly **imbalanced data**. Learning accurate models in the presence of such imbalanced data is a research topic on its own [72, 118, 162, 188]. We observe that in cases when information is lacking (e.g., short sentences) or context information is weak, this produces false positives for the larger class and false negatives for the smaller class.

A further (foreseeable) issue we encounter is related to the relatively simple approach of transforming a multi-label problem to multiple binary classification tasks. We already mentioned that the **binary relevance transformation** approach basically neglects the information on interdependencies between categories. In fact, we observe that sentences which are truly multi-labeled in the original data, relatively more often lead to misclassification. For example, consider the sentence "The location was central and close to everything, however resulting in a constant noise level during nights.", which refers to the two topics "location" and "noise". A binary classifier that learns to discriminate "location" sentences from "non location" is likely to miss the relation between location and noise. Out of all positive examples for the class "location", only a few include also the mention of topic "noise". And, as the concept "noise" is more often expressed without referring to the location, the number of negative examples mentioning "noise" is higher than the number of positive examples. The binary classifier is likely to learn that noise features are more indicative for the "non location" class. Imbalanced data further pronounces this issue.

The major reason for reduced recall is **data sparsity**. Natural language allows to refer to a topic in nearly arbitrary many ways. While the classifier is good in detecting very common indicator phrases, it falls off when reviewers paraphrase a topic by means of very specific formulations. There simply exist too few training data to successfully discover such cases. For instance, consider a sentence like "The sealing rubbers dry out after a while, so inform clients to use silicon gel.", which is labeled as topic "built quality". Weakly indicative features such as "sealing", "rubber", or "client" appear only once in the context of the "built quality" topic. The same is true for a sentence such as "I walked to the walgreens in the morning and saw drug paraphernalia on the ground.". While referring to the topic "security", the actual category is only implied. A human knows that discovering drug paraphernalia on the ground is typically a sign for a more dangerous area. However, the classifier has too few data to learn this correlation. Data sparsity is a general problem in supervised machine learning (cf., Bishop [39, chap. 1.4]). In our case and going along with the issue of imbalanced data, we observe that

especially the small categories "security", "decoration", "air conditioning", "built quality", or "low-light performance" suffer from low recall.

In summary, we identify the following four predominant causes for failures:

- **Weak or missing context information**: Classifying sentences (especially short ones) is, in comparison to document classification, prone to this type of error. The difficulty is inherent to our specific task. We tried to alleviate the issue by providing the classifier with information from the adjacent sentences. However, we could not achieve any improvement by incorporating these type of features (see the previous section). We hypothesize (without further proof) that the amount and complexity of possible correlations between adjoining sentences is too high in relation to the size of our training data.

- **Imbalanced data**: This issue is also inherent to our task. It is a fact that a small number of topics is far more often referred to by reviewers. However, here is potential for improving the classifier performance. A good overview of learning in the presence of class imbalance is provided by He and Garcia [162]. Most common approaches to handle imbalanced datasets are sampling techniques [49, 119, 239], cost-sensitive learning [116, 246, 405], and the adaptation of kernel-based methods [134, 169, 444] such as support vector machines.

- **Binary relevance transformation**: Transformation of the multi-label problem to a binary classification task neglects inter-class correlations and leads to an increased misclassification rate. We already pointed out in Section 8.3.1 that more sophisticated methods have been proposed in the literature to handle multi-label data directly.

- **Data sparsity**: The most obvious approach to overcome the data sparsity problem is to increase the size of the training dataset. But also very obviously, this is the most laboriously approach. Semi-supervised learning [1, 71] and active learning [285, 343] represent algorithmic approaches to alleviate the issue at reduced costs. For our specific setting, we will examine an approach based on incorporating weakly labeled data in the next section of this chapter.

8.3.3. Related Work

Ganu et al. [138] examine the use of support vector classifiers for detecting the prevalent topic(s) of a sentence. Their manually labeled corpus of restaurant reviews covers four domain related topics and two miscellaneous topics. For each of these topics they train a separate, binary **SVM classifier**. As features they use stemmed tokens. Results are reported as precision and recall values, separately for each topic. For the four domain related topics a macro-averaged f-measure of 73.4 is achieved. Taking all topics into account, the macro-averaged f-measure decreases to 67.1. micro-averaged results are not reported. The authors do not further explain their choice of topics. For example Brody and Elhadad [55] who experiment with the same corpus, find a different, more fine-grained set of topics.

Also Blair-Goldensohn et al. [40] examine the application of supervised machine learning methods to classifying sentences with regard to the topic dimension. They as well consider service reviews, namely restaurant and hotel reviews. For each domain manually labeled corpora of roughly 1,500 sentences are created (the corpora are however not publicly available). They also distinguish only a very small number of topics, four in the restaurant domain and five in the hotel domain. As Ganu et al., they train binary classifiers for each topic, but they use **maximum entropy models** (multinomial logistics regression) instead of SVMs. No information is provided with regard to the feature set used for classification. Results for the restaurant dataset are comparable to the numbers reported by Ganu et al.. Blair-Goldensohn et al. report a macro-averaged f-measure of 71.1 compared to 73.4. Achieved results for the hotel domain are slightly better with a macro-averaged f-measure of 76.4.

8.4. Exploiting Weakly Labeled Data

In this section we examine the utility of weakly labeled data for extracting coarse-grained product aspects with a supervised classification approach. In this regard, we basically extend the previous section. We apply the same techniques (e.g., binary relevance transformation, MaxEnt classifiers), but additionally incorporate weakly labeled training data. Obviously, our motivation for considering the use of weakly labeled samples is based on the expectation to successfully reduce the costs of gathering training data for supervised learning. In the following we provide an overview of the basic idea. Our concrete implementation is described in Section 8.4.1.

While sifting through numerous customer reviews, we found that authors (especially of longer reviews) often structure their writings by providing section headings. In this context, we observed two common patterns: Section headings either introduce a new topic/aspect or they refer to one of the discourse functions we have presented earlier. This is natural, as the section headings reflect the actual flow of topics and discourse functions within a review on a more abstract level. To give a few examples: Section headings referring to the discourse function "advice" may for instance read "My advices", "A few advices", "Tip", or "Some hints". Section headings for the topic "built quality" may be "On built quality", "Durability", or "Robustness:". The majority of "discourse section headings" refers to the function "sentiment". They introduce sections which enumerate either the advantages (pros) or disadvantages (cons) of the reviewed entity. Exemplary headings would be "Pros", "The good", "The positive", "Downsides", "What I dislike", or "Drawbacks:".

The basic idea of our approach is very simple: We make the intuitive assumption that there is a strong correlation between the section heading's semantic and the actual semantic of the headed content (in the following referred to as *headed segment*). Figure 8.5 illustrates this idea. The excerpt from a hotel review shows the three section headings "Room:", "Location:", and "Atmosphere and service:". Naturally, the topic implied by each of these headings is reflected by the related paragraphs. For instance, the first two sentences of the "location" segment "We loved staying in Pacific Heights, and we found it a really good alternative to staying in the city center. It's a lovely neighborhood [...]." clearly correlate with the topic. Interpreting the heading as label, these two sentences constitute perfect positive samples for training a "location" classifier. The same is true for the first two sentences of the "room" segment. The sentences "Our queen bedroom was quite big, and our window faced the tiny interior courtyard. Sure, the view towards the Golden Gate Bridge would have been nicer [...]." are without question related to the topic "room".

However, whereas on the level of paragraphs our assumption is valid at quasi a one hundred percent, this is not the case when referring to the sentence level. For example, consider the first sentence of the "atmosphere and service" segment. The sentence "These two go hand in hand in this hotel." is actually a comment on the the heading and does not relate to the topic on its own. Extracting such a sentence as positive example for the "service" topic would introduce noise into an heuristically generated training set. A human annotator would have labeled such a sentence as being "off-topic". The example thus shows that a heuristic which creates positive samples by extracting sentences from a headed segment and using the heading as a label, in fact may generate an inconsistent training set. In other words, such training data is only weakly labeled and is likely to contain more noise than traditionally curated training corpora. We are thus interested in examining whether this type of data can actually help in our context.

8.4.1. Implementation

We now discuss our concrete implementation of the basic idea to exploit section headings as labels. We first provide details on the extraction heuristics we use to automatically generate the labeled data.

[14]The illustration shows an excerpt of a Tripadvisor.com review by user "cantona7": http://www.tripadvisor.com/ShowUserReviews-g60713-d80983-r76549118

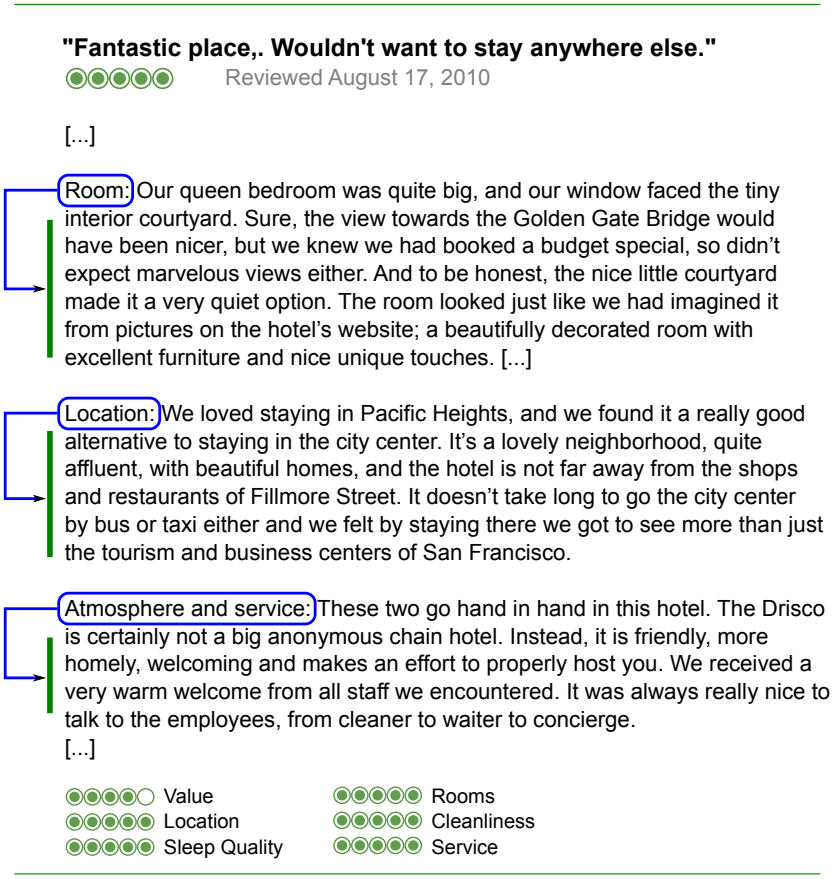


Figure 8.5.: An indirect crowdsourcing approach that exploits section headings as user signals[14].

Second, we demonstrate how we use the generated training data to support the task of detecting coarse-grained product aspects in customer reviews.

Extraction of Weakly Labeled Data

The illustration in Fig. 8.6 provides an overview of the extraction process. A first step is to use a web crawler to collect huge amounts of customer reviews for the desired product type. Of course, these reviews must be of a type that includes free text comments. In our case, we utilize the collections of hotel and digital camera reviews we have introduced in Chapter 5. Recall that both collections consist of several hundred thousand documents. The next step is to process each individual review:

Extract Paragraphs We automatically partition the free text part of each review into a set of non-overlapping paragraphs. This step is performed by a heuristic that simply splits the free text's string representation at boundaries defined by at least two consecutive newline characters. In printed material, it is a convention to start a paragraph at a new line and to indent it. But as indention is typically not directly available in HTML forms, reviewers most commonly use two or more newline characters to indicate a paragraph break. Thus, our very simple heuristic performs very well in detecting paragraph boundaries. Take note that not all customer reviews are structured by paragraphs — many authors do not use line breaks at all. So the result of this step may also be a "partitioning" into a single paragraph only.

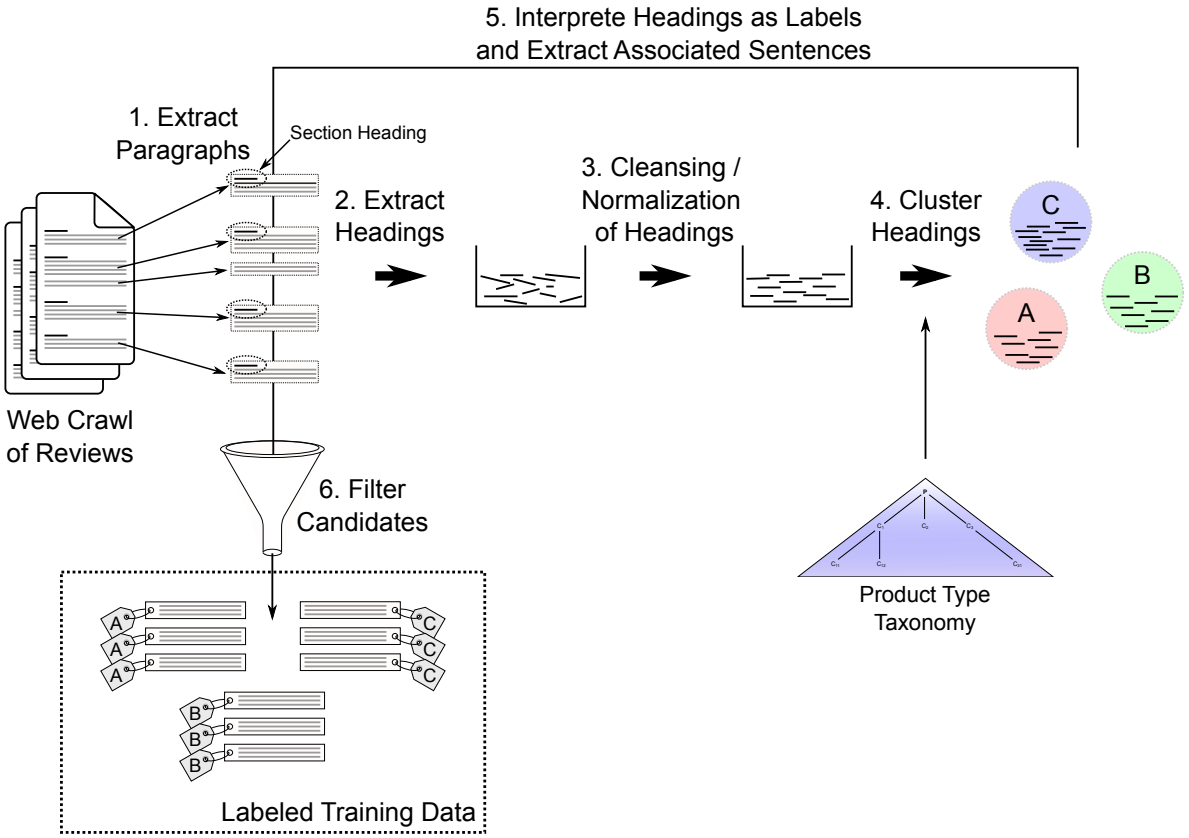


Figure 8.6.: Schematic overview of the process that generates a labeled training corpus from section headings.

Extract Headings Having segmented the review into a set of paragraphs, only two possibilities exist for the occurrence of a heading: A heading either occurs as its own (short) paragraph (i.e., it is separated from other text by at least two newline characters) or it appears at the beginning of a longer paragraph. Now the task is to distinguish headings from regular text. If the particular review site allows to use HTML tags for structuring, we can simply look for heading tags in the HTML source of the review. However, not all sites allow to use HTML and if available, not all reviewers make use of this option. Refraining from the HTML markup, we identified two intuitive patterns that are strong indicators for a heading. The first pattern applies to headings that occur in its own paragraph: If a paragraph consists of a single line that is not terminated by a punctuation mark (except for a colon) and is no longer than h_{max} characters, it is most likely a valid section heading. After some experimenting, we set h_{max} to 40 characters (including whitespace). The second heuristic applies to headings that appear at the beginning of a longer paragraph: Here, we basically use the colon symbol (':') as strong indicator. We consider the first h_{max} characters of the paragraph. If the string contains a colon character that is not followed by a digit (e.g., as in "11:00 am") or another punctuation mark (e.g., as in ":-)"), we regard the substring to the left of the colon as a heading. The heading candidate may consist of any alpha-numeric and whitespace characters, but must not contain a newline symbol. For both heuristics we set the minimum length of a heading h_{min} to 3 characters. We use regular expressions for encoding the constraints defined by the two patterns and for extracting the actual headings. Both heuristics provide a reasonably high precision and a good recall (the majority of section headings follows one of the two patterns). However, take note that only a few paragraphs are actually entitled by a section heading. The most common case is that reviewers do not provide headings at all. Further, a paragraph may also be subordinate to a preceding paragraph that is directly

headed. In fact there exists a 1:N relation between headings and paragraphs (a heading may set the topic for multiple paragraphs).

Cleansing and Normalization While the precision for the task of distinguishing headings from regular text is high, the extracted headings still contain a lot of "noise" that masks the actual "label" we are interested in. For instance, we may extract the headings "The service:", "On the service:", "1. Service:", "About the service:", or "Towards the service". Obviously all headings label subsequent paragraphs as referring to the topic "service". So the next step is cleansing and normalization of extracted headings. We would like to remove the clutter in form of different prepositions, determiners, or enumeration symbols and map the label to a canonical form (e.g., "service" instead of "hotel services"). To remove the clutter, we apply several cleansing operations, for example including:

- Delete preceding/trailing whitespace and dashes (e.g., " – Service – " –> "Service").
- Delete preceding enumeration markers such as "1.", "a)", "*", etc. (e.g., "* Service" –> "Service").
- Delete preceding determiners and possessive pronouns
  (e.g., "My final thoughts" –> "final thoughts").
- Delete "filler phrases" such as "Comment on the...", "As for the...", "About the...", etc. For this purpose we created a lexicon of the most common filler phrases used in section headings of customer reviews. We use it as a stop word list (e.g., "As for the service" –> "service").
- Delete review related terms/phrases such as "Problem with...", "...issues", "...highlights", etc. (e.g., "Service issues" –> "Service").

There are more heuristics which we do not list here. Our basic procedure to find a reasonable set of cleansing operations was iterative. We identified the most frequent "noise" pattern, generalized from it and implemented a cleansing step to remove this particular type of clutter. We ran our extraction procedure again, verified the results and identified the next common pattern until results were satisfiable. For normalization, we lowercase the whole heading and use the Stanford CoreNLP tool to lemmatize each token (e.g., "Hotel Services" –> "hotel service"). In addition, we remove any preceding reference to the name of the product type. For instance in the case of hotel reviews, we delete the occurrences of "hotel" in headings, such as "hotel room", "hotel location", or "hotel amenity".

Clustering/Grouping At this point the extracted headings are merely strings, not related to any of the predefined topics we are interested in. For instance, after cleansing and normalization we may find headings such as "neighborhood", "location", "surrounding area", or "immediate area" which all refer to the same topic "location". Our goal is thus to identify, group and associate all headings that are semantically similar to one of the predefined topics. For this task we can perfectly use the product type taxonomy which encodes exactly the required information. Just as the lexicon entries, relevant section headings are normally of the *nominal mention type*. For each processed heading, we look up whether a matching lexicon entry exists (including all variants). In case of a match, we use the most specific subtopic related to the matched term as the label we associate with the heading. For instance, the heading "parking fee" would match a lexicon entry that is categorized as a feature of the subtopic "parking". We thus associate the heading with the label "parking". We found that for a few (hotel: 2, camera: 3), but very frequent headings, no corresponding lexicon entry exists. We manually associate such headings with the correct group.

Associate Paragraphs to Section Headings Irrespective of whether we can associate a label with a section heading or not, we need to correctly identify the paragraphs that are related to the heading. We use a simple heuristic: We iterate through all immediately following paragraphs and attach them

to the heading until we reach a paragraph that is a heading on its own or starts with a heading. To prevent erroneous extractions, we link at most the first $dist_{max}$ paragraphs to a single heading. The intuition with such an upper bound is also that more distant paragraphs are more likely to be inconsistent with the actual topic indicated by the heading. A further parameter p_{max} constrains the maximum size of a single paragraph. We assume that valid paragraphs are bound in their length. In fact, this restriction prevents some errors which would have been introduced due to imperfect heuristics (here we trade recall against precision).

Extraction, Filtering and Labeling Sentences We now have a set of headed segments. Our heuristics partitioned the review into a set of paragraphs and when possible, associated a heading to an individual section. From the set of headed segments we extract those whose heading can be associated with one of the predefined topics. All other segments (headed or not) are simply discarded. As our final task is to generate a training set for classifying sentences, we need to extract sentences from the paragraphs. This is done by using a standard *sentence splitter* tool (we use Stanford CoreNLP). To remove sentences that are most likely not valid candidates for our training set, we apply several filter steps: First, we discard all sentences that consist of fewer than $tok_{min} = 4$ tokens after stop word removal. Second, we introduce a parameter $index_{max}$ that constrains the maximum index of a sentence within a paragraph. Our observation is that the first sentences within a paragraph exhibit a stronger correlation to the section's topic than following ones. Authors may get off the subject in longer paragraphs. Our observation also correlates with the concept of topic sentences[15]. We discard all sentences that occur at a position greater than $index_{max}$. Sentences that pass the filters are labeled with the topic induced by the heading and are added to the related training corpus.

Incorporating Weakly Labeled Data

We consider two alternatives for incorporating the automatically created training data:

- Train classifiers on weakly labeled data only: This approach has the obvious advantage that there is no need at all for the expensive step of manually creating a training corpus. As our heuristics for extracting training data are agnostic to the particular product domain, such an approach scales well over different domains. However, as the training data is imperfect, we assume that the overall accuracy does not reach the performance of a system that is provided with access to manually labeled data.
- Use weakly labeled data in addition to manually labeled corpus: In the previous section we have seen that most of the loss in recall is related to data sparsity. With the automatically extracted samples, we can add additional training data at virtually no cost. The hope is to relieve the problems with data sparsity to a certain extent and in consequence increase the overall recall of the system. However, the question is whether the noise introduced by using the weakly labeled data deteriorates the overall precision, thus overcompensating the expected value.

8.4.2. Experiments and Results

Basic Experimental Setup

Both, the heuristic process for gathering weakly labeled training data, as well as the maximum entropy classifier are parameterized. To guarantee comparability with previous results the MaxEnt classifier takes the same parameters as defined in Table 8.3. With regard to the heuristic extraction process, we use the parameter values given in Table 8.8. As the table shows, we do not fix the values

[15] Citing the Merriam-Webster Online Encyclopedia, a topic sentence is "a sentence that states the main thought of a paragraph or of a larger unit of discourse and is usually placed at or near the beginning" (http://www.merriam-webster.com/dictionary/topic%20sentence).

parameter	value	description
h_{max}	40	the maximum length (in characters) of a valid section heading
h_{min}	3	the minimum length (in characters) of a valid section heading
p_{max}	4000	the maximum length (in characters) of a valid paragraph
p_{min}	60	the minimum length (in characters) of a valid paragraph
tok_{min}	4	the minimum length (in tokens, after stop word removal) of a valid sentence
$dist_{max}$	experiment	the maximum distance of a paragraph to the section heading
$index_{max}$	experiment	the maximum index of a sentence within a paragraph

Table 8.8.: Parameter settings for the MaxEnt classifiers.

for the parameters $dist_{max}$ and $index_{max}$ as they are subject to experimenting. The parameter $dist_{max}$ takes either the value 1 or 2, that is we either consider the first paragraph only, or consider both, the first and second paragraph. We vary the maximum sentence index in 1, 2, 3, 4, or 10, considering either only the first, the first two, the first three, etc. sentences within a paragraph.

We run our extraction heuristic on the crawled customer review collections we introduced in Chapter 5. Prior to application, we filtered out reviews which free text part was shorter than 200 characters. For the hotel domain this resulted in a collection of 393,360 customer reviews and for the digital camera scenario we obtained 127,143 reviews.

Basic Statistics of Weakly Labeled Corpora

We will first present some basic statistics of the extracted training data and then take a closer look on its actual quality. The basic statistics are presented in Table 8.9. We read the table for example as

	hotel				digital camera			
corpus	#paragr.	#sent.	reviews/sent.	#chars	#paragr.	#sent.	reviews/sent.	#chars
w-1-1	9,266	9,172	42.89	809,429	4,623	4,576	27.78	436,614
w-1-2	″	17,376	22.64	1,519,060	″	8,073	15.75	769,873
w-1-3	″	23,546	16.71	2,059,176	″	10,628	11.96	1,018,249
w-1-4	″	27,894	14.10	2,431,007	″	12,411	10.24	1,193,163
w-1-10	″	36,898	10.66	3,190,582	″	15,988	7.95	1,534,321
w-2-1	12,157	12,020	32.73	1,089,434	6,711	6,639	19.15	662012
w-2-10	″	46,166	8.52	4,036,135	″	23,739	5.36	2,308,970

Table 8.9.: Basic statistics of the weakly labeled training corpora for topic classification. The corpus name indicates the parameter settings used for extraction. For instance, the notation "w-2-4" refers to a weakly labeled corpus that is created with $dist_{max} = 2$ and $index_{max} = 4$.

follows: Within the 393,360 hotel reviews our high-precision/low-recall heuristics were able to find 9,266 paragraphs that immediately follow a section heading and which could be correlated to one of the predefined topics. Considering only the first sentence within all these paragraphs, we obtain 9,172 valid sentences[16]. The ratio "reviews per sentence" expresses how many reviews we need to process in average to find a single (valid) weakly labeled sentence. Considering only the first paragraph and sentence, we need to parse 42.89 reviews in average. Observe that this number gives an estimate on how many reviews we need to crawl for a given training corpus size. Naturally, the ratio decreases with increasing tolerance of the heuristics. For instance, when setting $index_{max} = 10$, we need four times less reviews for obtaining a single labeled sentence. The table shows that the ratio is generally lower for the digital camera review collection. We hypothesize that this stems from the different review types. Tripadvisor.com (hotel reviews) allows reviewers to provide explicit ratings

[16]Some sentences are filtered out because they are too short ($< tok_{min}$).

for individual aspects. On the other hand, most camera reviews in our collection stem from Amazon.com, where reviewers do not have this opportunity. They are thus animated to provide more structure within the free text part. The table further shows that even in the most conservative setting (w-1-1), the automatically extracted hotel corpus is around three times larger than the manually compiled training set. For the digital camera domain this ratio is around 1.5:1.

Accuracy of Extraction Heuristics (Intrinsic Evaluation)

We now consider the accuracy of various corpora. More precisely, we measure the data quality in terms of the proportion of correctly labeled data. The correctness of an extraction is evaluated by manual inspection. In particular, we randomly sample 200 sentences from a specific corpus and manually mark them as being either correct or false. For this analysis we only consider the hotel corpus and assume that our results are transferable to other domains. We first compare the data quality of corpora when increasing the tolerance with respect to the sentence index (we fix $dist_{max} = 1$ and vary $index_{max}$).

Considering only the first sentence of a paragraph, we observe a precision of 0.91, that is 182 of the examined samples are correctly labeled by the heuristics. As expected the precision decreases with higher tolerance. Varying $index_{max}$ with values 2, 3, 4, and 10, we find precision values of 0.85, 0.82, 0.80, and 0.72. To more closely examine the influence of this parameter, we compare to a corpus that is based of sentences at index 10 only (i.e., only considers sentences at 10th position within a paragraph). In this case the precision drops to 0.51. These numbers confirm our hypothesis that the first sentences within a paragraph are more indicative for the topic induced by the section heading.

Next, we examine the influence of the $dist_{max}$ parameter. To this end we analyze two additional corpora. When extracting sentences ($index_{max} = 1$ is fixed) from both, the first and second paragraph, we observe a precision of 0.70. This is around 20 percentage points lower than when considering only the first paragraph. To further underpin the dramatically worse behavior of the heuristic with increasing paragraph distance, we examine a corpus that consists of sentences taken from the second paragraph only. Here, we find a precision of merely 0.24, indicating that is not reasonable at all to incorporate other paragraphs than the first one.

Classification Performance (Extrinsic Evaluation)

In the following we discuss our results with regard to the performance of classifiers trained on weakly labeled data. As we want to compare to results of the previous section, the particular setup for the underlying experiments is analogous to the one described before (see Section 8.3.2): Our experiments cover the two alternatives for incorporating weakly labeled data as discussed earlier in the implementation part. When training on weakly labeled data only, we train on the complete weak corpus and test on the complete gold standard corpus (no need for cross validation). When using the weakly labeled corpus as additional dataset, we use 10-fold cross validation with regard to the gold standard corpus. That is, for each fold we use the complete weak corpus and 9/10 of the gold standard corpus for training. The remaining tenth of the gold standard corpus is used for evaluation. To increase validity of results, we repeat this process 10 times with differently permuted gold corpora. To guarantee comparability, we use the same pseudo random numbers for permutation as in the previous section (all training and test folds are the same). Of course, we also use the same feature generators. In particular, we use unigrams with lemmatization and downcasing. As we are especially interested in reducing the manual effort involved with the task of detecting coarse-grained product aspects, we also neglect the use of the manually created knowledge-base. For the following experiments we do not use any of the knowledge-rich features. All tables report results for the polar subset of the gold standard corpus.

One obvious problem which we encounter when using the weakly labeled dataset is that this data

	all		collapsed	
features	**micro-f**	**macro-f**	**micro-f**	**macro-f**
baseline	0.817	0.778	0.824	0.805
weakly-1-1	0.804 (-0.013**)	0.771 (-0.007**)	0.824 (-0.000)	0.811 (+0.006**)
weakly-1-2	0.797 (-0.020**)	0.755 (-0.024**)	0.818 (-0.007**)	0.802 (-0.004**)
weakly-1-3	0.785 (-0.033**)	0.742 (-0.036**)	0.808 (-0.016**)	0.796 (-0.009**)
weakly-1-4	0.783 (-0.035**)	0.738 (-0.040**)	0.808 (-0.016**)	0.794 (-0.011**)
weakly-1-10	0.765 (-0.052**)	0.715 (-0.064**)	0.798 (-0.026**)	0.782 (-0.024**)

(a) hotel corpus

	all		collapsed	
features	**micro-f**	**macro-f**	**micro-f**	**macro-f**
baseline	0.765	0.746	0.780	0.782
weakly-1-1	0.756 (-0.010**)	0.746 (+0.000)	0.773 (-0.007**)	0.789 (+0.006**)
weakly-1-2	0.765 (-0.001)	0.742 (-0.004**)	0.781 (+0.001)	0.792 (+0.009**)
weakly-1-3	0.766 (+0.000)	0.740 (-0.006**)	0.774 (-0.007**)	0.792 (+0.009**)
weakly-1-4	0.752 (-0.013**)	0.725 (-0.020**)	0.770 (-0.010**)	0.782 (-0.000)
weakly-1-10	0.740 (-0.026**)	0.708 (-0.038**)	0.765 (-0.015**)	0.774 (-0.009**)

(b) digital camera corpus

Table 8.10.: Results for varying the parameter $index_{max}$ (weakly labeled data only). Differences to the baseline (gold standard corpus only) are shown in brackets. Double asterisks indicate a significance at the 99% confidence level.

is (in contrast to the manually crafted corpus) not multi-labeled. Our heuristic is not capable of detecting whether a sentence discusses multiple topics. We find that the majority of extracted sentences which actually should be multi-labeled, targets a single main topic and one or more of its subtopics. For example, the extracted sentence "Small sized room with TV, DVD player, small table and tiny bathroom (with tub and Aveno products though)." addresses the main topic "room" and the subtopics "room amenities" as well as "bathroom". With our *binary relevance transformation* approach to multi-label classification such a sentence would count as a negative example for the subtopic categories. In order to relieve this issue, we also look at results when collapsing subtopics, that is only considering the main topics for evaluation. In the following tables the standard scenario with all topics (main and sub) is denoted as "all", whereas the simplified scenario is named "collapsed".

We first consider the results when using weakly labeled data only. Table 8.10 summarizes the achieved micro and macro-averaged f-measures for varying parameter $index_{max}$[17]. As reference to training with weakly labeled data only, we use our results when training with the gold standard only (see Section 8.3.2). The obtained results are quite encouraging: The best setting with training on weakly labeled data achieves f-measures which are only slightly below the baseline which has access to "perfectly" labeled data. In terms of micro-averaged f-measure the most conservative weakly labeled setting (weakly-1-1) achieves a value of 80.4% (75.6%) which is only 1.3 (1.0) percentage points worse than the baseline. In other words, we can completely do without manual labeling and still achieve very competitive results. Considering the macro-averaged results, the gap is even slightly less.

When comparing the results with varying $index_{max}$, we observe the same tendency as with the intrinsic evaluation. With increasing tolerance of the heuristics (trading precision for recall), the classification performance deteriorates (more pronounced in terms of macro-averaged f-measure). Confirming the intrinsic evaluation results, we conclude that choosing the more precise heuristics ($index_{max} \leq 3$) is most reasonable.

[17] We do not consider different values for $dist_{max}$ as the intrinsic results with higher values were not promising.

	all		collapsed	
features	**micro-f**	**macro-f**	**micro-f**	**macro-f**
baseline	0.817	0.778	0.824	0.805
weakly-1-1	0.825 (+0.008**)	0.801 (+0.023**)	0.833 (+0.008**)	0.822 (+0.017**)
weakly-1-2	0.833 (+0.015**)	0.785 (+0.007**)	0.826 (+0.002**)	0.816 (+0.011**)
weakly-1-3	0.823 (+0.006**)	0.777 (-0.002)	0.820 (-0.004**)	0.814 (+0.009**)
weakly-1-4	0.820 (+0.003)	0.767 (-0.012**)	0.818 (-0.007**)	0.807 (+0.001)
weakly-1-10	0.783 (-0.034**)	0.742 (-0.036**)	0.814 (-0.010**)	0.800 (-0.005**)

(a) hotel corpus

	all		collapsed	
features	**micro-f**	**macro-f**	**micro-f**	**macro-f**
baseline	0.765	0.746	0.780	0.782
weakly-1-1	0.794 (+0.028**)	0.781 (+0.035**)	0.802 (+0.022**)	0.812 (+0.029**)
weakly-1-2	0.796 (+0.030**)	0.777 (+0.031**)	0.802 (+0.022**)	0.806 (+0.023**)
weakly-1-3	0.788 (+0.023**)	0.773 (+0.028**)	0.802 (+0.021**)	0.815 (+0.033**)
weakly-1-4	0.775 (+0.009**)	0.757 (+0.011**)	0.790 (+0.009**)	0.798 (+0.016**)
weakly-1-10	0.766 (+0.000)	0.746 (+0.000)	0.789 (+0.009**)	0.792 (+0.009**)

(b) digital camera corpus

Table 8.11.: Results for varying the parameter $index_{max}$ (weakly labeled data + manually annotated data). Differences to the baseline (gold standard corpus only) are shown in brackets. Double asterisks indicate a significance at the 99% confidence level.

We further examine the specific "collapsed" scenario which considers only the main topics. Here, we can even observe improved results (macro-averaged) with using the weakly labeled data. Obviously, the quality of the automatically extracted datasets is sufficiently high. The positive effects due to the larger size of the generated corpora overcompensate the negative effects introduced by slightly lower data quality.

In the following we present the results when using the weakly labeled data in addition to the manually labeled corpus. The results of the corresponding experiments are shown in Table 8.11. We consistently observe improved results for both product domains. Using the most conservative setting for the extraction heuristic, we improve the macro-averaged f-measure by 2.3 (3.5) percentage points compared to when not using the weakly labeled data. The increase in micro-averaged f-measure is slightly lower with 0.8 (hotel) and 2.8 (camera) percentage points. Whereas the absolute results achieved within the "collapsed" scenario are higher, the improvement by using weakly labeled data is less compared to the scenario with all topics.

In summary, we can conclude that exploiting the weakly labeled data is reasonable for both implementation approaches. If no manually labeled data is available or too costly to obtain, the automated extraction of training corpora promises to achieve competitive results. In the other case, if training data is available, but too sparse, additionally using weakly labeled data is very likely to improve the overall classification performance.

8.5. Summary and Conclusions

In this chapter, our goal was to provide a detailed study of approaches for handling the topic dimension of the discourse oriented model for customer reviews. In particular, our goal was to automatically attribute each discourse segment (sentence) with one or more coarse-grained aspects from a set of predefined topics.

Section 8.1 described the problem setting and formalized the task as an instance of a hierarchical multi-label classification problem. We pointed out that both, multi-label classification and hierarchical classification, can either be reduced (transformed) to traditional classification problems or can be tackled with dedicated algorithms. In both cases, we opted for the simpler transformation approaches.

In Section 8.2, we presented a simple lexicon-based approach for the (sentence-oriented) discovery of coarse-grained product aspects. Lexicons were automatically extracted by means of the terminology extraction techniques presented in the previous chapter. A manual post-processing step introduced the necessary semantic information: We associated each individual lexicon entry with one of the predefined topics and hierarchically structured them along semantic relations. The resulting knowledge base implemented the product type taxonomy as defined in Section 4.1. Classification was conducted by simply matching lexicon entries within a sentence and then looking up the related coarse-grained product aspect(s). Our main findings were:

- Even with the relatively simple lexicon-based approach and without performing any further fine-tuning, we achieve quite good results: When considering only polar sentences (which is most relevant in a review mining system), we observe micro-averaged f-measures of 75.7% (hotel) and 74.0% (digital camera).

- The main reason for reduced precision is the lack of context awareness with a lexicon-based approach (> 80% of all false positives). Classification is based on matching individual words and phrases only, disrespecting any contextual clues. Thus, the most promising way of improving the precision of a lexicon-based approach is to include rules that take the context into account. Further reasons for false positives were ambiguity of lexicon entries and partial matches because of missing entries.

- Loss in recall is mainly due to the failure of recognizing implicit mentions of aspects (> 80% of all false negatives). To improve the recall, a lexicon could (in addition to nominal mentions) include a set of the most common phrases that reviewers typically use to paraphrase specific aspects. Further causes for false negatives were missing lexicon entries and unrecognized variants.

- The results showed significant deviations in accuracy, depending on the particular class/aspect under consideration. Whereas very concrete aspects, which are mainly represented by nominal mentions of concrete entities, achieve good results, "abstract" topics are more often referred to by implicit mentions, thus receiving significantly worse results. The very low f-measure for some particular aspects is primarily due to very low recall, precision values are generally quite good.

In Section 8.3, we experimented with a supervised machine learning approach. In particular, we used our sentence level corpora to train maximum entropy classifiers. To tackle the hierarchical multi-label data, we used binary relevance transformation and flattened the hierarchy. With regard to our experiments, we were primarily interested in answering the following two questions: First, how does the supervised approach compare to a simple lexicon-based method and do results justify the additional costs caused by the necessity to provide labeled training data? Second, which set of features/-variables is most effective when classifying review sentences according to their topic? The following listing summarizes our findings:

- Also without further optimization and not using a dedicated classification algorithm for hierarchical multi-label data, we observed significant improvements in comparison to the purely lexicon-based method. With the best combination of features, the macro-averaged f-measure increases by around 13.5 (hotel) and 11.0 (camera) percentage points. Both, precision and recall lead to the improved f-measure.

- Most significant improvements are observed for the more "abstract" topics which were difficult to recognize with the purely lexicon-based approach. The learning algorithm successfully discovers also implicit mentions.
- Experimenting with different feature sets revealed that lemmatized unigram features exhibit the best results for our corpora. Including other features such as part-of-speech tags or bigrams does not improve performance.
- Using a hybrid (lexicon + supervised) approach, by providing lexicon information as features to the supervised classifier, showed significant improvements. By adding lexicon matches and lexicon predictions as features, we can increase macro-averaged f-measure up to 5.5 percentage points compared to the baseline (lemmatized unigram features).
- We hoped to increase precision by including features based on lexicon matches and lexicon predictions in adjacent sentences. But providing the classifier with this context information even slightly deteriorates its performance.
- Mistake analysis revealed mainly four types of failures, including "imbalanced data", "data sparsity", "information loss due to the binary relevance transformation", and "weak or missing context information". Based on this analysis, we pointed out most promising ways for improvement.

In Section 8.4 we examined the utility of weakly labeled data for extracting coarse-grained product aspects with a supervised classification approach. Our primary motivation was to reduce the costs of gathering training data needed for supervised learning. The basic idea was to exploit the semantic relation between section headings and entitled paragraphs, which reviewers often use to structure their writings. To this end, we split reviews in paragraphs (if possible) and identified associated section headings. By relating headings to the predefined topics, and interpreting them as labels, we could extract weakly labeled training sentences. Our main questions in this context were: How well does a system perform that is trained on the automatically extracted data alone? Further, we wanted to know whether a combination of weakly and manually labeled data can improve the overall classification performance. Our results were:

- Using the weakly labeled data alone, only slightly deteriorates performance in comparison to a system that has access to manually labeled data. The macro-averaged f-measure was at maximum 0.7 percentage points lower. We conclude that the positive effects due to the larger size of the generated corpora overcompensate the negative effects introduced by the slightly lower data quality (nonetheless, 91% of the "weakly" labeled data are correctly labeled by the extraction heuristics).
- We can also answer the second question in the affirmative: Enriching a manually labeled corpus by means of the heuristically extracted training data promises to increase overall classification performance. On our corpora we observed increases in macro-averaged f-measure by 1.7 (hotel) and 3.3 (camera) percentage points.
- Experiments with the tolerance of the extraction heuristics revealed that the most conservative setting (trading recall for precision) achieved best results. More precisely, it is most reasonable to only use the topic sentences in paragraphs directly following a section heading for training.

Your highness, when I said that you are like a stream of bat's piss, I only mean that you shine out like a shaft of gold when all around it is dark.

Monty Python's Flying Circus

9. Automatic Acquisition of Domain-Specific Sentiment Lexicons

In previous chapters we pointed out that aspect-oriented customer review mining mainly involves two core tasks: The first task aims at identifying the relevant *aspects* that are discussed within a review and the second task is to analyze the *sentiments* expressed towards these aspects. Whereas the previous two chapters extensively studied the "aspect dimension" of review mining, we will now focus on the "sentiment dimension". In this context we can distinguish two broad categories — namely, lexicon-based or supervised classification approaches to polarity detection[1]. In this chapter our goal is to examine lexicon-based approaches (we cover supervised methods in the next chapter).

More precisely, we are interested in the task of **automatically constructing sentiment lexicons**. As we have learned in previous chapters (see for example Section 6.2.2), one major challenge is that the actual sentiment polarity of an expression is often dependent on the referenced target/aspect (e.g., "long$^+$ battery life" vs. "long$^-$ flash recycle time"). Although this phenomenon is known and discussed in the literature [14, 77, 105, 125, 189, 214, 306], the vast majority of approaches focuses on creating general purpose lexicons. However, especially in the context of customer review mining, the use of such lexicons is rather suboptimal as they fail to adequately reflect the domain-specific lexical usage. To this end, we propose a novel method that allows to **automatically adapt and extend existing lexicons** to a specific product domain. We follow a corpus-based approach and exploit the fact that many customer reviews exhibit some form of semi-structure. In particular, we make use of the structural clues inherently provided by the pros and cons summaries of reviews. We sketched the basic idea of this approach in Bross and Ehrig [56].

The remainder of this chapter is organized as follows: Section 9.1 provides a general overview of lexicon-based polarity detection, categorizes different approaches, and considers the most relevant related work. In Section 9.2, we briefly review two existing approaches to lexicon construction that we will (amongst others) use for comparison and basis for adaptation. Subsequently, Section 9.3 presents our approach to automatic acquisition of a context-aware sentiment lexicon. Experiments and results are discussed in Section 9.4. We summarize our findings and point out our conclusions in Section 9.5.

9.1. Overview and Related Work

When reviewing the literature and inspecting available systems, it is apparent that the use of sentiment lexicons is the most common approach to sentiment analysis. In this section we provide an overview and introduce a framework for comparing different approaches. Our framework categorizes the properties of sentiment lexicons along different dimensions, which we structure in a hierarchical manner. Figure 9.1 illustrates the basic constituents of our framework. As with product aspect lexicons (Chapter 7), we basically differentiate between *construction* and *application* of lexicons. In the whole chapter, we are primarily concerned with the process of lexicon construction and touch its application only on the surface[2].

[1] Of course, we have approaches in between — for example, with semi-supervised methods or by combining lexicon-based and supervised detection.

[2] How to apply the lexicon for extracting the relevant information is a question on its own. For example, with regard to expression level analysis, a major challenge is to correctly identify the relations between sentiment expressions and sentiment targets. Take note that this step may actually be performed in a supervised manner [183, 207]

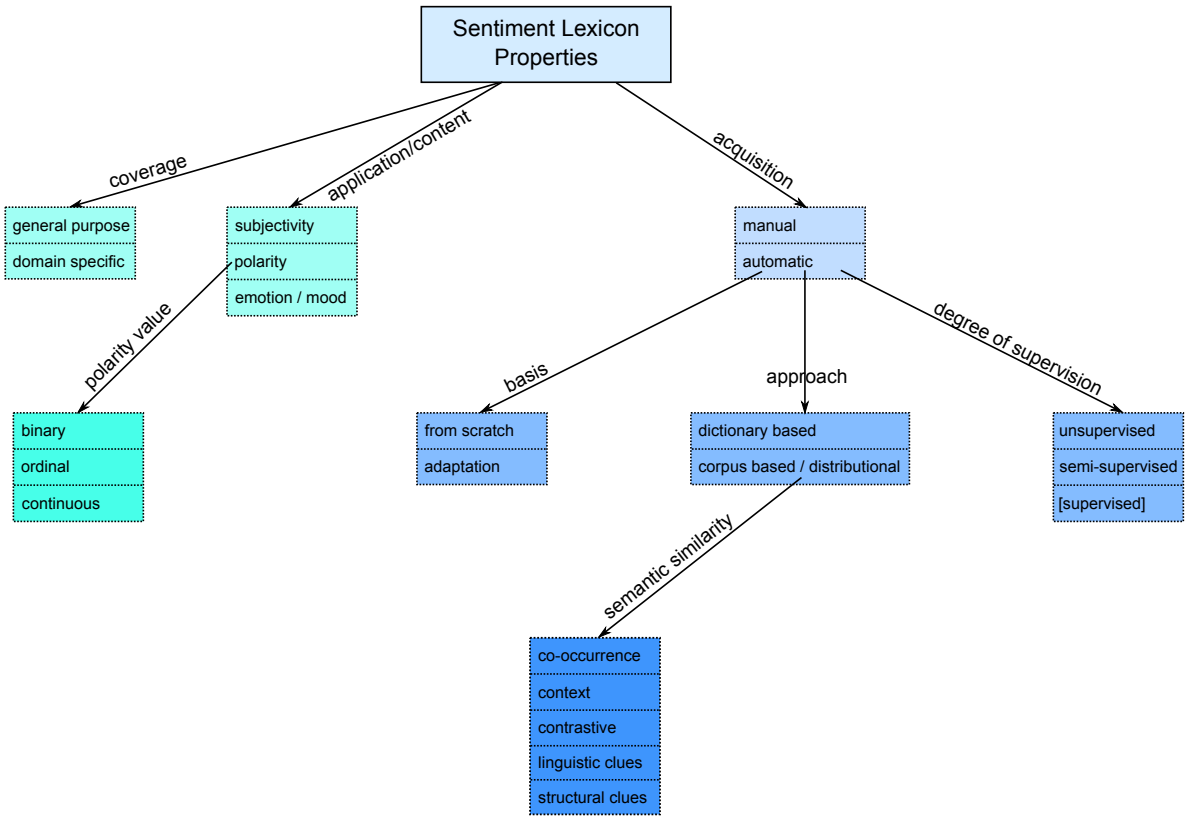


Figure 9.1.: A framework for the categorization of sentiment lexicons. Siblings in the tree represent properties that are considered to be orthogonal to each other.

The very basic properties we distinguish are concerned with a lexicon's *coverage*, its *type of content*, and the *acquisition process*.

9.1.1. Lexicon Coverage

With regard to coverage, we differentiate between general purpose and domain specific lexicons. The assumption with general purpose lexicons is that the sentiment status of lexicon entries is independent of the concrete application domain. Whereas this is assumption is true for a great share of words/phrases (e.g., "great", "love", "bad", "hate", etc.), we have seen earlier that the sentiment status may depend on the domain or context. Further, many expressions that are per se not sentiment laden (e.g., "long", "hot", "classic", "old"), may obtain an evaluative connotation within a specific context (e.g., "classic hotel building" vs. "old hotel building"). Our observation is that recent research has recognized the need for domain specific lexicons [77, 105, 214].

9.1.2. Content Type and Application Scenario

We further differentiate with respect to the lexicon's type of content. Obviously, the content is closely related to the application scenario. Concerning this dimension, the majority of lexicons can be attributed to one or more of the following three categories: The sentiment status of a lexicon entry may encode the word's *degree of subjectivity*, its *polarity*, or a related type of *emotion / mood*. Sentiment lexicons that primarily serve the task of subjectivity detection are for instance discussed in [26, 157, 245, 414, 415]. In contrast, lexicons that are used in the context of customer review mining most commonly address the polarity value of individual words and phrases. Naturally, this

information is most valuable, as we are primarily interested in identifying the positive and negative comments of reviewers. However, researchers also formulate an information need with regard to a more fine-grained analysis of textual sentiment expressions. Besides, or in addition to considering the polarity, they categorize expressions by different types of emotion or mood[3] (e.g., happiness, anger, sadness, etc.). Such an information need is for example postulated by Bollen et al. [48], who analyze microblogging messages to predict stock market changes, Garcia and Schweitzer [139], who examine emotions in product reviews, or more generally by Davidov et al. [98] and Strapparava and Mihalcea [365]. Most prominent lexical resources in this context (all manually compiled) are the *Harvard General Inquirer*[4] [360], the *Affective Norms for English Words* (ANEW) dictionary[5] [50], *WordNet-Affect*[6] [364, 395], and the *Dictionary of Affect in Language* (DAL) [408].

Polarity Value — Scale of Measurement

In case of a sentiment polarity lexicon, we can further distinguish by the specific representation of the polarity values — more precisely, by their *scale of measurement*. Many lexicons use a binary scale (i.e., "positive" vs. "negative") [146, 160, 177, 312], others provide more fine-grained information by using an ordinal scale. For example, quite common is a 5-point scale[7] that distinguishes "strong" and "medium" positive/negative expressions and a neutral category [450]. Also ternary scales ("positive", "negative", "neutral") are used [82, 438]. Furthermore, some lexicons make use of a continuous scale by computing scores for the polarity value [40, 320, 391, 399].

Concerning the **acquisition** of sentiment lexicons, we basically differentiate between manual assembly and automatized construction. The majority of publicly available, manually compiled resources are general purpose lexicons (e.g., the *OpinionFinder Subjectivity Lexicon* [438] or the *Harvard General Inquirer* [360]). Obvious disadvantages of the manual approach are the considerable manual effort, the typically low coverage of created lexicons, and the high costs of scaling vertically (i.e., to cover additional domains). With regard to automatized lexicon construction, we further distinguish three different dimensions: We differentiate according to the concrete *approach*, the *degree of supervision*, and whether the resource is constructed from scratch or on the *basis* of an existing sentiment lexicon.

9.1.3. Automatic Lexicon Construction

The vast majority of automatic approaches falls into two classes: They are either *dictionary-based* (exploiting the semantic relations between dictionary entries) or they are *corpus-based* (exploiting distributional properties to derive the sentiment status of individual words or phrases). Approaches following either of the two directions have typically in common that they involve a *seed set* of prelabeled examples, which is (often iteratively) used to derive the sentiment status of more and more entries. For this purpose, approaches test the semantic similarity between unknown words and labeled seed words.

Dictionary-based Approaches

For dictionary-based approaches, researchers most commonly use a thesaurus or more knowledge-rich lexical databases such as WordNet [263]. A common assumption is that semantic relations, such as synonymy or antonymy, transfer the sentiment status of related words (in case of antonymy the

[3] see Plutchik [302] for more information
[4] http://www.wjh.harvard.edu/~inquirer/Home.html
[5] also adapted to German language [394]
[6] http://wndomains.fbk.eu/wnaffect.html
[7] inspired by the psychometric *Likert-scale* commonly used in questionnaires [231]

polarity is flipped) [146, 177, 202, 211]. For example, the adjective "lovely" transfers positive polarity to its synonyms "admirable", "adorable", "amiable", "pretty", and transfers negative polarity to its antonyms "awful", "unlovely", "ugly". Albeit being transitive, the strength of the relations weakens with the distance (in fact there exist synonym paths from "good" to "bad" of length 3 in WordNet [146]). Appropriate measures need to be devised to account for the path length [59, 146, 180, 202, 211]. Besides synonymy and antonymy, some researchers propose to use additional WordNet relations, such as "similarity", "derived-from", "pertains-to", "attribute", or "also-see" [120, 395]. Takamura et al. [376] and Andreevskaia and Bergler [13] also consider the less intuitive hyponymy relation. A further dictionary-based approach is to infer semantic relatedness of entries by calculating similarity by means of the glosses [22, 120, 377].

Corpus-based Approaches

Also with corpus-based approaches the primary idea is to calculate a measure of semantic relatedness and use this (most often in conjunction with labeled seed words) to derive the sentiment status of other words or phrases. Examining these approaches more closely, we identify mainly four different ways to infer relatedness: It may be defined by *co-occurrence statistics*, *distributional context similarity*, *contrastive analysis*, or by exploiting *linguistic* or *structural clues*.

Co-occurrence Representatives for approaches based on co-occurrence statistics are for instance the works by Turney [390, 391]. Extending the general idea of co-occurrence, Turney and Littman [391] hypothesize that also the "semantic orientation[8] of a word tends to correspond to the semantic orientation of its neighbors". Using the Web as a corpus, they apply measures of association, such as pointwise mutual information (PMI), to derive a correlation statistic of an unseen word with a set of positive and negative seed words. Also Remus et al. [320] follow this basic approach to construct a sentiment lexicon for the German language.

Context Similarity Besides inferring relatedness of two terms directly by their co-occurrence, it is a common approach in *statistical semantics* to define similarity indirectly by means of the words' context. In Firth's *Contextual Theory of Meaning*, the basic assumption is that "a word is characterized by the company it keeps" [130]. Analogously to Turney, it is suggested that words with a similar context also exhibit a similar sentiment status. Corpus-based approaches that exploit this idea are for instance [29, 399, 414].

Contrastive Analysis In Section 7.3 we have learned that contrastive analysis of foreground and background corpora can be used to extract candidate terms for a product aspect lexicon. Analogously, this general approach can be applied to automatically generate sentiment lexicons. For example, Maks and Vossen [245] examine log-likelihood and relative frequency ratios to distill a lexicon of subjective words for the Dutch language. For this purpose they propose to use newspaper articles and comments on newspaper articles as (subjective) foreground corpus and a collection of Wikipedia articles as (objective) background corpus. A similar idea is introduced by Stepinski and Mittal [359], who compare corpora of editorial/opinion and general news articles in the context of sentiment classification[9].

Linguistic Clues The previously cited approaches to corpus-based lexicon generation are of purely statistical nature and do not rely on any deeper linguistic analysis. However, other studies have

[8] Turney and Littman use the term *semantic orientation* to refer to the prior sentiment polarity of a word.
[9] In fact, no sentiment lexicon is generated, but two contrastive corpora are extracted to train a sentence level sentiment classifier.

identified general linguistic patterns that help in detecting the sentiment status of words and phrases. The earliest work in this direction (we are aware of) is by Hatzivassiloglou and McKeown [160]. They basically exploit the observation that conjunctions (e.g., "and" and "but") "impose constraints on the semantic orientation of their arguments" [160]. Whereas the conjunction "and" generally implies that conjoined phrases exhibit the same sentiment polarity, the conjunction "but" implies opposing polarities. For instance, whereas the phrase "the hotel staff was helpful *and* courteous" sounds natural, the phrase "the hotel staff was helpful *and* impolite" does not. Hatzivassiloglou and McKeown [160] use these linguistic constraints to extract positive and negative oriented adjectives from a text corpus. In addition to the "conjunction rule", they also examine morphological clues to identify synonyms and antonyms (e.g., "adequate" vs. "inadequate" or "thoughtful" vs. "thoughtless"). In the context of customer review mining, the conjunction rule is for example applied by Popescu and Etzioni [304] or Fahrni and Klenner [125]. Kanayama and Nasukawa [203] extend the basic (intra-sentential) rule by also looking at inter-sentential conjunctions (e.g., "The hotel staff was generally helpful. *However*, the front desk staff was a bit impolite."). Also Ding et al. [102] exploit intra and inter-sentential conjunctions, but in addition analyze the target-specific prior polarity of words.

Structural Clues Kaji and Kitsuregawa [200, 201] examine a method that uses structural clues in HTML documents to extract corpora of positive and negative sentences. In particular they identify tables and listings in HTML documents that address the advantages and disadvantages of discussed entities. Indicative keywords (e.g., "pros", "cons", "weaknesses") that are structurally related to the tables and listings are used to filter out irrelevant data. They apply their method to a corpus of 120 million HTML documents and are able to extract around 500.000 polar sentences (with negative sentences being slightly more frequent). Whereas they show in [200] how to use the extracted data to train a sentiment classifier, they also consider generating sentiment polarity lexicons from such corpora [201].

9.1.4. Degree of Supervision

Orthogonal to the concrete method (i.e., dictionary vs. corpus-based), we distinguish the approaches' degree of supervision. If we count techniques that are based on the provision of small seed sets as semi-supervised, the majority of methods to sentiment lexicon generation falls into this category. To name a few, this includes dictionary-based approaches such as [40, 177, 211] or corpus-based approaches such as [390, 399, 414]. Completely unsupervised methods are for example by Hatzivassiloglou and McKeown [160], who use linguistic clues in combination with a clustering approach or Kaji and Kitsuregawa [201], who exploit structural clues in HTML documents. We are not aware of any fully supervised approach to sentiment lexicon generation (although some approaches rely on supervised classification after an initial bootstrapping phase, e.g., [22, 324, 367]).

9.1.5. Lexicon Adaptation

All previously considered approaches generate the sentiment lexicon from scratch, using either general purpose dictionaries or raw text corpora. More recently, researchers examine methods to adapt existing sentiment lexicons. Typically, the purpose of adaptation is either to augment a general purpose sentiment lexicon to better fit a specific *domain* or to extend a monolingual lexicon to cover *multiple languages*. Domain adaptation is for instance examined by Choi and Cardie [77] who propose an approach based on linear programming, Du et al. [105] who introduce an information theoretic framework, or Qiu et al. [306] who expand existing lexicons by means of linguistic patterns. Jijkoun et al. [189] consider domain adaptation in the context of sentiment retrieval and Gindl et al. [144] propose a method to identify terms that exhibit an ambiguous sentiment polarity in different domains.

An approach to language adaptation is for example presented by Mihalcea et al. [262]. In particular, they examine dictionary-based and corpus-based methods for translating a sentiment lexicon.

9.1.6. Hybrid Approaches

In addition to purely dictionary or corpus-based approaches, some researchers study the utility of hybrid approaches that combine indicators from the different sources. For instance, Hoang et al. [165] propose to use semantic relations in WordNet to create an initial sentiment lexicon, which is then refined by incorporating statistical information gathered from the Web as a corpus. Both sources (WordNet and the Web corpus) are combined by means of an error minimization algorithm. Also Lu et al. [241] propose to combine various sources of signals that indicate the sentiment polarity of words. In particular, they consider four types of signals: Information is collected from a general purpose sentiment lexicon, from a thesaurus, from linguistic clues, and from structural clues in domain specific documents. All different signals are combined in an optimization framework that is based on a linear programming approach.

9.1.7. Notes on Comparing Lexicons and Approaches

It is apparent that our analysis of related work does not provide any numbers which indicate the effectiveness of the different approaches. In fact, most of the results presented in the various studies cannot reliably compared to each other: Application scenarios differ widely, standardized reference corpora are lacking, and evaluation procedures vary. Citing reported results may thus be misleading and we decided not to do so. Statements on relative effectiveness are only possible if authors have tested their system in direct comparison to other approaches. Studies that provide such a comparative analysis are explicitly marked in Table 9.1 (see column "Evaluation").

As introduced in Chapter 7, procedures for evaluating lexical resources can be mainly subdivided into intrinsic (accuracy of the lexicon itself) and extrinsic evaluation (accuracy as part of an application). The cited works mostly follow this scheme. With regard to intrinsic evaluation, a common approach is to manually inspect the generated lexicon entries. In addition, some studies propose heuristics to estimate the lexicon accuracy. For example, [146, 160] test the ratio of "morphological antonym pairs" (e.g., helpful-unhelpful, polite-impolite, competent-incompetent, etc.) that exhibit opposing polarity (correct) and pairs that show the same polarity (incorrect). Intrinsic evaluation only allows to analyze the precision of the lexicon generation process. Recall is often (qualitatively) estimated by means of the lexicon size or by considering the overlap with other lexical resources [22, 146, 391, 399]. Extrinsic evaluation is dependent on the application scenario and most studies come up with a proprietary gold standard that suits their specific needs. Despite the use of different reference corpora, comparison of extrinsic evaluation results is particularly difficult as the actual techniques for applying the sentiment lexicon play a major role. For instance, the lexicon may be applied within a fine-grained, expression level sentiment analysis system or a more coarse-grained, sentence or document-level analysis. To summarize our overview, Table 9.1 presents (what we consider) the most relevant related work regarding sentiment lexicons, highlighting the main properties of the different approaches.

author	approach	coverage	size	content	degree of supervision	evaluation
Stone et al. [360]	manual acquisition	general purpose, words	10^3	binary polarity	—	—
Wilson et al. [438]	manual acquisition	general purpose, words	10^3	subjectivity + quaternary polarity	—	extrinsic with gold standard [413]
Liu et al.[10]	automatic acquisition + manual revision	general purpose, words	10^3	binary polarity	semi-supervised	extrinsic with gold standard [102, 177]
Hu and Liu [177]	*D*: WordNet relations, disregarding path lengths	general purpose, adjectives only	10^3	binary polarity	semi-supervised	extrinsic with gold standard [102, 177]
Godbole et al. [146]	*D*: WordNet relations + filter heuristics	general purpose, words	10^3	continuous, normalized polarity	semi-supervised	intrinsic with heuristic and gold standard [438]
Blair-Goldensohn et al. [40]	*D*: WordNet relations + label propagation	general purpose, words	10^4	continuous polarity	semi-supervised	extrinsic with own gold standard
Rao and Ravichandran [312]	*D*: WordNet relations + label propagation	general purpose, words	n/a	continuous, normalized polarity	semi-supervised	intrinsic with gold standard [360] (*comparative*)
Baccianella et al. [22]	*D*: WordNet relations + gloss classification	general purpose, words	10^5	continuous, normalized polarity	semi-supervised	intrinsic, gold standard [68]
Turney and Littman [391]	*C*: Web corpus + co-occurrence + PMI scoring	general purpose, words	n/a	continuous polarity	semi-supervised	intrinsic with gold standard lexicon [160, 360]
Fahrni and Klenner [125]	*C*: Domain specific corpus + conjunction rule	domain specific, adjectives only	n/a	ternary, target-specific polarity	semi-supervised	extrinsic with own gold standard (*comparative*)
Kaji and Kitsuregawa [201]	*C*: Web corpus + structural clues + statistical scoring	general purpose, adjective phrases	10^2-10^3	binary polarity	unsupervised	qualitative, intrinsic with own gold standard
Velikovich et al. [399]	*C*: Web corpus + context similarity + graph propagation alg.	general purpose, phrases	10^5	continuous polarity	semi-supervised	qualitative, extrinsic with own gold standard (*comparative*)
Lu et al. [241]	*D*+*C*: hybrid approach + optimization via linear programming	domain specific, adjectives	10^3	binary, target-specific polarity	semi-supervised	extrinsic with own gold standard (*comparative*)

Table 9.1.: Related work with respect to sentiment lexicon acquisition. The table distinguishes between manual lexicon acquisition, dictionary-based approaches (*D*), corpus-based approaches (*C*), and hybrid approaches (*D*+*C*).

9.2. Baseline Approaches - Label Propagation in WordNet

Before discussing our own contribution, we briefly review two existing approaches that we will use as a baseline for later experiments. We choose the algorithms by Blair-Goldensohn et al. [40] and Rao and Ravichandran [312] because they represent state-of-the art, dictionary-based approaches to generating a general purpose sentiment lexicon. Both exploit WordNet relations and are based on a semi-supervised learning method called *label propagation* [466]. Whereas the core idea is the same (label propagation), the algorithms differ in some important points and it is unclear which one leads to better results. So in addition to setting up a baseline, our goal is to compare the effectiveness of both existing approaches. In the following, we first provide a short overview of the common ideas to both algorithms and then present each in some more detail.

Input to both algorithms is a thesaurus (WordNet in this case) and sets of seed words with known labels (i.e., the prior polarity). Output is a lexicon where each entry is associated with either positive or negative prior polarity (in form of a polarity score). Higher (absolute) scores indicate the algorithm's stronger confidence in the polarity classification. The basic approach is to interpret WordNet's semantic relations as edges of a directed graph and to apply a label propagation algorithm to iteratively push the seed information along the edges. Similar to random walk algorithms (e.g., PageRank [290]), the result is that nodes (i.e., words) with many paths to seeds ("authorities") receive higher polarity scores[11]. A main advantage of the label propagation algorithm over ad-hoc heuristics, such as the ones used in [146, 177, 202], is its foundation on a well defined objective function, which is known to converge [466]. For these reasons and for the reason that label propagation has shown superior performance compared to other graph based algorithms (e.g., *mincuts*) [312], we opt for the mentioned baseline approaches.

9.2.1. Rao et al. Method

Let $G(V, E)$ be a graph with vertices V and edges E. Then the vertices $v \in V$ represent the lexicon entries in WordNet and an edge $e \in E$ indicates a relevant semantic relation between two entries. Unfortunately, Rao and Ravichandran [312] do not sufficiently describe the exact procedure of encoding WordNet relations in G. In particular, it is unclear how WordNet's synsets are treated and whether the available part-of-speech labels are used for disambiguation. We assume the following transformation procedure:

Let w be a word that is covered in the WordNet database. If applicable, WordNet distinguishes different part-of-speech labels for w, including *noun*, *verb*, *adjective*, or *adverb*. We create a vertex $v = w_{pos}$ for each part-of-speech label pos associated with w. For instance, for the word "nice" we create vertices $nice_{adj}$ and $nice_{noun}$ (the city Nice in France). This procedure is applied to the complete WordNet database, so that ultimately V represents a part-of-speech disambiguated set of all words in WordNet. In addition to distinguishing the four part-of-speech labels, WordNet differentiates multiple senses of a word by organizing its entries in *synonym sets* (synsets). For example, the adjective "nice" is associated with five different synsets. We fold the synsets part-of-speech-wise, that is, we create a single synset for each part-of-speech label. Let $syn_1, \ldots, syn_k$ be the synsets associated with a specific part-of-speech label pos of word w. Then we take the union $Syn(w_{pos}) = \bigcup_{i=1}^{k} syn_i$ and regard each $s \in Syn(w_{pos})$ as a synonym of w_{pos}.

In [312] the label propagation algorithm is implemented as follows: For each vertex $v_i \in V$ they consider the corresponding synonym set $Syn(v_i)$ and lookup related vertices v_j. For each such pair (v_i, v_j) an undirected edge e_{ij} is added to E. Based on this edge set E, an $n \times n$ stochastic transition matrix $T = (t_{ij})$ is created, where $n = |V|$. Let $A = (a_{ij})$ be the $n \times n$ adjacency matrix corresponding

[10] http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar

[11] Take note that the basic idea of applying the PageRank algorithm to WordNet for sentiment lexicon induction was also formulated by Esuli and Sebastiani [121].

to E, that is, $a_{ij} = \begin{cases} 1 & \text{if } e_{ij} \in E, \\ 0 & \text{otherwise.} \end{cases}$ Then T is derived by column-normalizing the matrix A:

$$t_{ij} = P(j \to i) = \frac{a_{ij}}{\sum_{k=1}^{n} a_{kj}},$$

where t_{ij} represents the transition probability from v_j to v_i. Now, label propagation can be viewed as iterative matrix multiplication of the transition matrix T and a label matrix Y, leading to the following algorithm:

1. Create a positive seed set $Pos \subset V$ and a negative seed set $Neg \subset V$ (naturally $Pos \cap Neg = \emptyset$).
2. Initialize an $n \times 3$ label matrix $Y = (y_{ij})$ by means of the seed sets, where $y_{i1} = 1$ if $v_i \in Pos$ and $y_{i2} = 1$ if $v_i \in Neg$. Further, set all entries $y_{i3} = 1$ if $v_i \notin Pos \cup Neg$. All other entries y_{ij} are initialized with 0.
3. Propagate the seed information by updating $Y = TY$.
4. Row-normalize Y so that each row sums up to 1 and the probability interpretation of the label matrix is preserved.
5. Clamp the positive and negative seeds in Y to their original value.
6. Repeat from step 3 until Y converges.

9.2.2. Blair-Goldensohn et al. Method

Rao and Ravichandran [312] point out that their label propagation implementation requires the exploited semantic relations to be transitive, that is propagating a label along relations must be consistent with regard to the sentiment polarity. A consequence is that they cannot utilize the antonym relation — sentiment polarity is assumed to be flipped along such paths. In contrast, Blair-Goldensohn et al. [40] adapt the standard label propagation algorithm and also allow to incorporate the non-transitive antonym relation. Further major differences are that they include a third seed set *Neut* which is composed of well-known neutral words and that they compute a polarity score instead of a polarity probability value. The concrete method is as follows:

The vertex set V is extracted from WordNet exactly the same way as described for the previous method. Analogously to deriving the folded synonym set $Syn(v)$ for each vertex, they construct antonym sets $Ant(v)$. Then they encode the synonym and antonym relations between vertices by a transition matrix $T = (t_{ij})$, where

$$t_{ij} = \begin{cases} 1 + \lambda & \text{if } i == j, \\ +\lambda & \text{if } v_i \in Syn(v_j) \ \& \ v_i \notin Neut, \\ -\lambda & \text{if } v_i \in Ant(v_j) \ \& \ v_i \notin Neut, \\ 0 & \text{otherwise} \end{cases}$$

and $\lambda \in]0, 1]$ is a decaying parameter that controls the effect of path lengths for label propagation. Smaller λ have the effect that the magnitude of propagated scores is reduced with increasing path length. Again, label propagation is implemented as matrix multiplication:

1. Create a positive seed set $Pos \subset V$, a negative seed set $Neg \subset V$, and a neutral seed set $Neut \subset V$ (naturally all sets must be pairwise disjoint).

2. Initialize an $n \times 1$ score vector $\mathbf{s}^0$ by means of the seed sets, where

$$s_i^0 = \begin{cases} +1 & \text{if } v_i \in \text{Pos}, \\ -1 & \text{if } v_i \in \text{Neg}, \\ 0 & \text{otherwise.} \end{cases}$$

3. Propagate the seed information by updating the score vector to $\mathbf{s}^{m+1} = T\mathbf{s}^m$.

4. Sign correct all scores of seed words $v_l \in \text{Pos} \cup \text{Neg}$ so that $sgn(s_l^{m+1}) = sgn(s_l^0)$.

5. Repeat from step 3 until M iterations have been conducted.

6. Compute the final score vector $\mathbf{s}$ by thresholding and then scaling logarithmically:

$$s_i = \begin{cases} log(|s_i^M|) * sgn(s_i^M) & \text{if } |s_i^M| > 1, \\ 0 & \text{otherwise.} \end{cases}$$

9.2.3. Adaptations

Sign-correct Step in Blair-Goldensohn et al. Method

During our experiments we found that the original approach by Blair-Goldensohn et al. [40] may lead to unintended low polarity scores for some seed words. For instance, we observed that the word "amazing" constantly received a polarity value close to zero in nearly all configurations. The main reason for this result is the fact that we tacitly disregard the different senses of a word by combining all synsets related to a single word. In the case of "amazing", there exists a synset with the sense "inspiring awe or admiration or wonder" which contains words such as "awesome", "awing", "awe-inspiring", but also "awful". Obviously most senses of the word "awful" and related words exhibit a negative connotation, which leads to a low polarity value for the original seed word "amazing". To avoid this behavior, we rewrite the "sign-correct" step of the original approach. Our goal is to ensure that the computed absolute polarity value for a seed word is at least as high as if it had no relations to other words. Algorithm 9.1 implements the new version of the "sign-correct" step.

Algorithm 9.1 Adapted version of the sign-correct step

```
function SIGN-CORRECT(iteration)
    minSeedScore ← (1 + λ)^iteration
    for all s ∈ Pos ∪ Neg do
        pol_s ← abs(polarity(s))          ▷ gets the current, absolute polarity value of s
        pol_s ← max(pol_s, minSeedScore)
        if s ∈ Neg then                   ▷ ensure that negative seeds have a negative sign
            pol_s = pol_s * (−1)
        end if
        updatePolarity(s, pol_s)          ▷ updates the current polarity of s
    end for
end function
```

9.3. Creating Domain-Specific Sentiment Lexicons Using Weakly Labeled Data

Both baseline approaches generate general purpose sentiment lexicons that associate a fixed prior polarity with each entry. However, we have learned in Section 4.3 that the sentiment polarity of words is typically context dependent. In particular, the polarity may depend on the sentiment target (cf., Section 6.2.2). In this section we propose and discuss an approach that exploits weakly labeled data in customer reviews to derive the domain and target-specific prior polarity of words. Recall that a major observation was that adjectives account for over 90% of these target-specific expressions. We thus concentrate on detecting **target-specific prior polarity of adjectives**. The approach is unsupervised and thus easily scales vertically to other product domains. We use the gained context-aware information to augment an existing general purpose lexicon. Experiments with our method show significant improvements in precision and recall compared to various state-of-the-art baseline approaches. In general, we define a target-specific sentiment lexicon as follows:

Definition 9.1 (Target-Specific Sentiment Lexicon)**.** *Let* $\boldsymbol{T}$ *be a sentiment target and* $\boldsymbol{S}$ *be a sentiment expression. Then a target-specific sentiment lexicon* $\boldsymbol{L}_{ts}$ *is a dictionary that maps tuples of type (T,S) to sentiment polarity values* $\boldsymbol{p} \in \mathbb{R}$*. Polarity values* $p < 0$ *indicate negative sentiment polarity of a tuple, whereas* $p > 0$ *indicates positive polarity. A target-independent sentiment expression S may be included by adding a tuple of the form (*, S), where the asterisk * refers to a wild card character.*

9.3.1. General Idea

Similar to the approach presented in Section 7.5, the general idea is to leverage the information contained in **pros and cons lists**, which are often attached to customer reviews. A reasonable assumption is that authors choose positive expressions when describing a product aspect in the pros, whereas negative expressions are used in the cons. In that sense, we regard "pros" and "cons" as labels and the associated text as positive and negative samples from which we can extract the desired information. Input to our algorithm is a collection of customer reviews C with associated pros and cons lists (C^+ and C^-) and a product type taxonomy P.

We illustrate the basic idea in Fig. 9.2. First (1.), we apply the taxonomy as a lexicon and identify all mentions of product aspects T within C^+ and C^-. Simultaneously, we use high precision heuristics to find potential sentiment expressions S (i.e., adjectives) that are linked to the aspects. This procedure generates a huge number of tuples (T,S). In a next step (2.), we utilize the semantic relations encoded in P to group tuples that have a similar target. Each such group is associated with a single "canonical tuple" as representative. Then, for each group, we acquire the occurrence counts (3.) in C^+ and C^- and apply statistical means (4.) to decide whether a group predominantly stems from the pros or from the cons. A group that significantly more often occurs in pros than in cons is considered as having positive semantic orientation and vice versa for groups originating from the cons. Optionally, we regard the acquired positive and negative sentiment expressions as (additional) seeds in a WordNet-based label propagation algorithm. The goal of this step (5.) is to increase the coverage of the final lexicon. For instance, if we have found that "large" in the context of "screen" is positive, we can expand this observation via WordNet to the assumption that also "big", "great", "bombastic", etc. are very likely to be positive in this context (and vice versa for "small", "little", "smallish", etc.). In the following we consider the individual steps in some more detail.

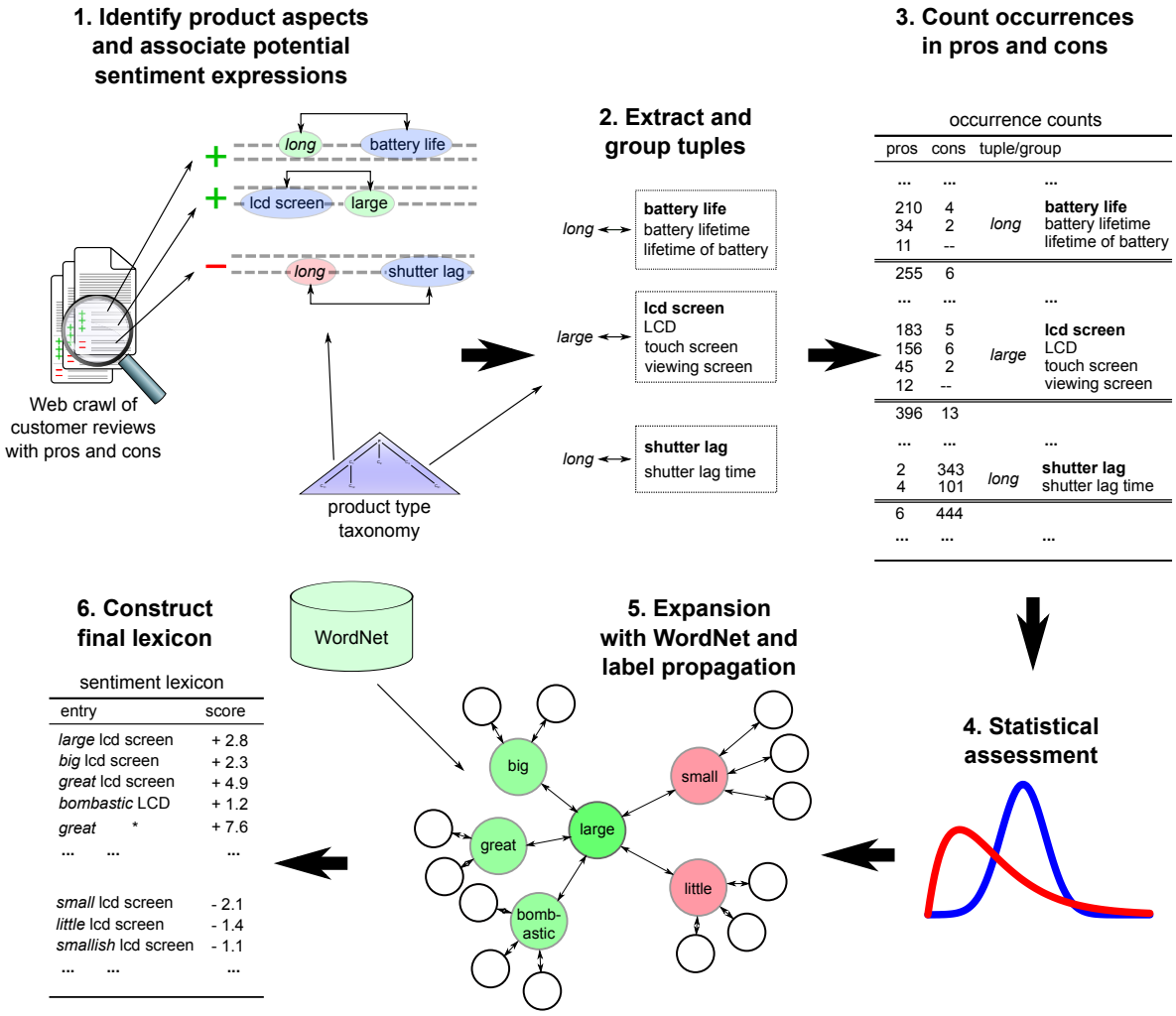


Figure 9.2.: Creating target-specific sentiment lexicons by exploiting the weakly labeled data in pros and cons summaries of customer reviews.

9.3.2. Extraction Process

Product Aspects

We use a lexicon-based approach to detect all known product aspects mentioned in the pros and cons. To this end, we employ the product type taxonomies created for the hotel and digital camera domains and use Algorithm C.2 for the actual matching. Thus, aspect detection only requires a lexicon and pre-processing of the texts with a part-of-speech tagger (again we use the Stanford POS tagger).

Sentiment Expression Candidates

Having identified the product aspects, our next goal is find adequate sentiment expression candidates and correctly relate these to the aspects. We use pattern-based extraction instead of relying on a natural language parser to guide the relation extraction process. The main reason is that pros and cons attached to customer reviews mostly do not consist of complete, grammatically correct sentences. Authors often simply enumerate the advantages and disadvantages in comma-separated or similarly structured lists. Our method is as follows:

First, we consider all commas and similar enumeration markers (e.g., "*", "a)", "1.") as boundaries

of so-called *extraction units*. We restrict valid co-occurrences of product aspects and sentiment expression candidates to these units — that is, a valid relation between an aspect and an expression must not span such a boundary. The basic idea with extraction units is to compensate for the fact that we cannot consider the real dependencies that a natural language parse would provide. Naturally, we also consider the beginning and end of complete sentences as boundaries. Second, for the actual extraction, we define high-precision patterns based on the part-of-speech tags associated with an extraction unit. For instance, consider an extraction unit that reads "the/DT zoom/NN buttons/NN are/VB quite/RB small/JJ" (abbreviated POS tags attached to each word). We are interested in extracting the tuple (zoom buttons, small). To generalize from this particular example, we can define a pattern such as `DT` A_1 `VB RB` JJ_1, where A_1 masks an identified product aspect (here "zoom buttons") and the subscripts denote the parts that should be extracted as a tuple. With this pattern, we can extract from other extraction units such as "the rooms were very outdated" (rooms, outdated) or "the battery life is rather short" (battery life, short). We can further generalize patterns by making parts optional or including valid alternatives. For instance, we can generalize the previous pattern to `[DT]` A_1 `VB [RB]` $JJ_{1.1}$ `[CC [RB]` $JJ_{1.2}$`]`, where the notation with square brackets marks optional parts. This pattern also matches token sequences like "rooms were small and very noisy" from which we can extract the two tuples (rooms, small) and (rooms, noisy)[12].

In total, we identified 9 high-precision patterns for the extraction of aspect/sentiment expression tuples from pros and cons. Table 9.3 (page 188) lists these patterns and provides information about their relative frequency in our corpora. Table 9.4 (page 189) gives additional insight by showing concrete instances of the patterns, including examples and extractions. Our basic procedure for gathering the patterns was to collect the part-of-speech sequences for all extraction units, measure the frequency of each distinct sequence, and then to analyze the 100 most frequent ones. Analysis of a part-of-speech sequence was mainly to look at the covered text of exemplary extraction units and to decide whether we can formulate a high-precision extraction pattern. Based on these very specific patterns, we formulated more general patterns, occasionally extrapolating from the known, valid sequences. In summary, gathering valid patterns was a completely manual task. We evaluate the effectiveness of the different patterns in Section 9.4.2.

Sentiment Shifters

The previous discussion tacitly disregarded the fact that the "labels" pros/cons indicate the *contextual polarity* of expressions, whereas we are interested in extracting the (target-specific) *prior polarity*. For example, excerpts from a cons-labeled text may read "display not very large" or "battery life not long enough". Clearly, we would make an error when counting the tuples (display, large) or (battery life, long) with negative polarity. We would have neglected the sentiment shifter "not". Another cons-labeled example may read "would like to have a large display". Here, sentiment is shifted by a neutralizer which would render an extraction (display, large) erroneous. To interpret contextual polarity correctly, we thus need to cope with sentiment shifters. We have learned earlier (cf., Section 4.3.2), that in terms of sentiment polarity, only **negation** and **neutralization** are relevant. As we can compensate potential errors in detecting sentiment shifters by the large number of extractions and subsequent statistical analysis, we can resort to heuristics. In other words, it does not count whether an individual extraction is correct or not, it is important that the vast majority is correctly handled.

Both, for negation and neutralization detection we use dictionary-based approaches. For negation detection we look for tokens such as "not", "n't", "never", "although", etc. and flip the polarities of affected sentiment expression candidates. The scope of negation is restricted to the extraction unit the indicating token occurs in. As it is very uncommon in lists of pros and cons, we do not further

[12] Take note that although the examples show the inflected forms, we actually extract the lemmatized results, e.g., (room, small) instead of (rooms, smaller).

consider the occurrence multiple negatives[13]. For detection of neutralization we use indicators such as "would", "could", "should, "might", "wish", "hope", etc. and simply discard the associated extraction unit(s). In contrast to negation, we allow that the scope of neutralization may span the boundaries defined by the extraction units. For example, we would discard the complete sequence "wish it had a larger display, stronger flash, otherwise perfect camera", instead of discarding only the first extraction unit "wish it had a larger display".

9.3.3. Grouping and Counting

The next step is to count the extracted tuples so that we can subsequently assess their polarity value by statistical means. We have learned earlier that the occurrence frequency of product aspect mentions approximately follows a Zipfian distribution. A few different aspects account for a large share of all mentions. For these very frequent aspects (and related tuples) it is very likely that we find statistically relevant correlations with target-specific sentiment expressions. With regard to more rarely occurring aspects (and the related tuples) we have a problem. The already low occurrence frequency of the aspect alone is even more reduced as it is distributed among multiple aspect-sentiment tuples. To relieve this situation and to eventually increase the coverage of the generated sentiment lexicon, we try to group similar aspects and subsequently calculate with the aggregated counts.

Recall that individual product aspects are related to each other via "part-of", "feature-of", "type-of", and "synonym-of" relations within our product type taxonomy. For grouping, we exploit the latter two relations. In particular, we postulate that the target-specific prior polarity of an adjective is consistent with regard to product aspects that are related to each other via the "type-of" or "synonym-of" semantic relation. For example, the positive prior polarity of "large" in the context of the aspect "screen" is also valid for the (near) synonyms "display", "monitor", or "video screen", as well as for the derived types "touch screen", "LCD screen", or "swivel screen". We are aware that this assumption is not always true and counter examples can be brought in. However, although this may cause that we construct some false entries, it is very unlikely that these mistakes lead to errors when applying the lexicon for polarity detection. If the association was erroneous, it is simply unlikely that we would find such a tuple in the actual data. For instance, if for some reason we falsely create an entry ("battery life", "cold", negative), chances to observe this adjective in the context of the aspect in real data are very low.

Our counts represent document frequencies, where we regard each pros or cons text as a single document (a single pros/cons document may consist of multiple extraction units). We use the following notation: Let $C^+(ws)$ be the document frequency of a word sequence ws in a corpus of pros documents, and let $C^-(ws)$ be defined analogously. Further, the variables $C^+(ws_1, ws_2)$ and $C^-(ws_1, ws_2)$ refer to the document frequency of co-occurrences (ws_1, ws_2) in the corresponding corpora. For all counts we will make use of aggregated group counts and statistically analyze the group as a whole.

9.3.4. Statistical Assessment

For statistical assessment of whether a group of tuples occurs significantly more often in the pros or cons, we design a hypothesis test analogous to the one introduced in Section 7.5. We consider the probabilities

$$\begin{aligned} p_1 &= Pr(S|T; Pros) \\ p_2 &= Pr(S|T; Cons), \end{aligned}$$

where S is a sentiment expression candidate and T is a product aspect. Then p_1 denotes the probability in a corpus of pros lists that we observe S, given that the product aspect T occurs in the same

[13] Double or multiple occurrences of negatives are handled as if only a single negation occurred.

extraction unit, and p_2 refers to the same probability, but in a corpus of cons lists. Again, the null hypothesis H_0 is that $p_1 = p = p_2$, that is, we assume that, independent of whether we consider the pros or cons corpus, the strength of association between the sentiment expression and product aspect is the same. The alternative hypothesis H_1 is $p_1 \neq p_2$, that is, either $p_1 > p_2$ or $p_2 > p_1$. If the test rejects H_0, we postulate that the tuple (T, S) has positive prior polarity if $p_1 > p_2$ and otherwise has negative polarity. We estimate p, p_1, and p_2 as

$$p = \frac{C^+(S,T) + C^-(S,T)}{C^+(T) + C^-(T)}, \quad p_1 = \frac{C^+(S,T)}{C^+(T)}, \quad p_2 = \frac{C^-(S,T)}{C^-(T)},$$

and as test statistics we use the log-likelihood ratio test as defined in Eq. (C.2).

9.3.5. Expansion and Incorporation to an Existing Lexicon

The previous steps showed how to acquire the target-specific polarity of adjectives from pros/cons summaries in customer reviews (e.g., "large" in the context of "screen"). In this section, we present methods for incorporating this knowledge into an existing general purpose lexicon. Before we do so, we would like to expand the previously acquired knowledge. For instance, we can conclude that also "big", "great", or "huge" have a positive connotation in association with "screen", whereas antonyms such as "little", "small", or "tiny" are negatively connoted. For this purpose we use WordNet and apply the label propagation algorithm in a similar fashion as described in Section 9.2.

In contrast to the baseline approaches, we cannot calculate a universally valid polarity score for the lexicon entries. Lexicon entries may now be dependent on a sentiment target. We thus need to adapt the lexicon construction procedure: The basic idea is to run the label propagation algorithm for each of the groups extracted in step 2. Let G be such a group of similar product aspects. Then we collect all adjectives S_G that have been found to exhibit a significant sentiment polarity in the context of G. Let S_G^+ be the subset of positively connoted adjectives and let S_G^- be the subset of expressions with negative polarity. For expansion we experiment with two different strategies:

- **Strategy A**: We extend the original positive seed set $Pos^* = Pos \cup S_G^+$ as well as the original negative seed set $Neg^* = Neg \cup S_G^-$ and run the label propagation algorithm as before. The intuition of combining the seed sets is to provide the algorithm with as much labeled data as possible. However, within the results we now have the problem to distinguish universally polar words from expressions that are only polar in the context of G. We only want to add lexicon entries $(t, s, score)$ where s is really dependent on $t \in G$. For instance, we want to add an entry ("screen", "huge", +), but not an entry ("screen", "excellent", +). Obviously, all words in $S_G^+ \setminus Pos$ and $S_G^- \setminus Neg$ belong to the target category. Out of the remaining sentiment expressions, we select the k ones with highest absolute polarity value. For these, we lookup their polarity in the lexicon that is to be extended and compare with the polarity we have found through target-specific expansion. We only include a new entry if the signs of the polarities differ. That may be the case if the target-specific polarity flips the original polarity or the original polarity value was zero. Algorithm 9.2 explains the procedure more formally.

- **Strategy B**: We only consider $S_G^+ \setminus Pos$ and $S_G^- \setminus Neg$ as positive and negative seed sets and run the label propagation algorithm as before. With this strategy we can assume that each word that reaches a sufficiently high polarity score is target-dependent on G and thus represents a valid extension to the existing lexicon. Again, we select only the k sentiment expressions with the highest absolute polarity value. See Algorithm 9.2 for more details.

Algorithm 9.2 Target-Specific Expansion and Lexicon Extension

```
Let L be an empty target-specific sentiment lexicon
Initialize L with entries from general purpose lexicon
Let SGr = S_G1, S_G2, ..., S_Gn be the set of group specific sentiment expressions S_G
Each (s, p) ∈ S_G is a tuple of the form (expression, polarity value)
for all S_G ∈ SGr do
    if strategy = A then
        S*_G ← S_G ∪ Pos ∪ Neg                          ▷ adds original seeds
    else if strategy = B then
        S*_G ← S_G \ {Pos ∪ Neg}                        ▷ removes original seeds
    end if
    S*_G ← EXPAND(S*_G)                                 ▷ Uses label propagation via relations in WordNet
    S*_G ← TOP-K({S*_G \ S_G}, k)                       ▷ selects the k entries with highest absolute polarity
    for all (s, p) ∈ {S*_G ∪ S_G} do                    ▷ adds original group specific sentiment expressions
        polarity_L ← POLARITY(L, s)                     ▷ looks up polarity in L
        if sign(polarity_L) ≠ sign(p) then
            L ← L + (s, G, p)                           ▷ adds to lexicon, G is the aspect's canonical form
        end if
    end for
end for
```

9.3.6. Gathering Domain Relevant Sentiment Expressions

In addition to evaluating the target-specific polarity of sentiment expressions, our goal is to exploit the extracted tuples for collecting a set of domain relevant, but target-independent sentiment expressions. For instance, in the hotel domain we observe tuples such as (room, homey), (room, well-furnished), (room, well-stocked), (room, mildewy), or (room, ratty). None of the enumerated sentiment expressions is contained in any of the general purpose sentiment lexicons. Some of the words are even not contained in WordNet. Further, we also observe many misspellings, for example (room, confortable), (room, beutiful), (room, spatious), or (room, cosy). All these exemplary tuples have in common that their frequency of occurrence is rather low, so that statistical assessment with the log-likelihood ratio test shows insufficient confidence in associating the tuples with either positive or negative polarity. Thus, in addition to statistically assessing the combinations of targets and sentiment expressions (i.e., for obtaining target-specific polarity), we evaluate the polarity of extracted sentiment expressions independently from their associated targets. Occurrence frequency is now defined for each distinct sentiment expression rather than for each distinct tuple. Considering an individual sentiment expression, we may now have more statistical evidence for deciding on its polarity. For instance, we may observe the tuples (room, confortable), (bathroom, confortable), (hotel, confortable), and (bed, confortable) all with a frequency of five occurrences in the pros. For each distinct tuple the statistical evidence may not be sufficient, but when aggregating the counts we have a frequency of 20 for the expression "confortable" in the pros part. This may be sufficient to find that it is positively connoted.

In particular, our procedure for finding domain relevant, target-independent sentiment expressions is as follows: First, we aggregate the counts for each distinct sentiment expression by summing up the counts for each tuple the expression occurs in. We now have frequencies C^+ (pros) and C^- (cons) for each distinct sentiment expression. Then, analogously to the assessment of target-specific expressions, we design a log-likelihood ratio test which considers the occurrence probabilities in the pros versus the cons corpus. Again, the null hypothesis is that the word is equally distributed in both corpora and the alternative hypothesis is that it is more likely to occur in either the pros or in the cons corpus. For this test we require a confidence level of 99% which corresponds to a log-likelihood

ratio of 6.63. We further reduce the size of the extracted lexicon by considering only the top-*k* positive and negative expressions (ordered by absolute score). In our experiments we set *k* to 750, so that the complete lexicon contains 1,500 entries. All results obtained with the approaches presented in this section will be discussed in Section 9.4.4.

9.4. Experiments and Results

9.4.1. Experimental Setup

Our main evaluation goal is to assess the effectiveness of the different construction procedures introduced previously. We do so by examining the quality of each of the resulting sentiment lexicons. In particular, we distinguish the following three types of failures:

- **Type-1 (false polarity)**: A lexicon entry truly exhibits a prior sentiment polarity, but the polarity value is incorrect. For example, a lexicon entry such as (*, fantastic, -5.9) correctly refers to a word with prior polarity ("fantastic"), but its polarity value is erroneously negative (-5.9). Another example would be a target-specific entry such as ("battery life", "short", +2.1), where "short" clearly is sentiment bearing, but was incorrectly classified as positive.

- **Type-2 (non-polar entry)**: The construction process may generate a lexicon entry that does not exhibit a prior sentiment polarity at all. For instance, entries such as (*, "analog", +1.0), (*, "put", -2.0), or ("battery life", "cold", -1.0) would be erroneous as the related sentiment words are not polar[14].

- **Type-3 (missing entry)**: This type of error occurs if the lexicon does not contain an entry that has been marked in some related gold standard dataset. For instance, if our expression level annotation dataset contains a comment "... staff behaved really ignorantly ...", but entries such as (*, "ignorantly", -1.8) or ("staff", "ignorantly", -2.1) are both missing, then the sentiment cannot be correctly detected.

Our evaluation procedures are designed to address these three error types. As in Chapter 7 we consider intrinsic and extrinsic evaluation.

Intrinsic Evaluation

With intrinsic evaluation, we manually inspect the lexicon entries. It allows us to find errors of type-1 and type-2. To enable a more fine-grained analysis, we consider six different regions of each lexicon: We subdivide the lexicon so that partition L^+ covers all positive and partition L^- all negative entries. We order the entries of each partition descending by their absolute polarity score. Then, each of the two partitions is further subdivided into three equally sized sub-partitions $L^{+/-}_{top}$, $L^{+/-}_{mid}$, and $L^{+/-}_{bottom}$. From each of these six sub-partitions we randomly sample 50 lexicon entries as representatives for the corresponding region of the sentiment lexicon (manually inspecting the complete lexicons is out of question at lexicon sizes of more than 10^3 entries). With this approach, we can evaluate the positive and negative polarity parts separately and can further verify the validity of the computed polarity scores as a confidence measure.

[14] Take note that, when regarding polarity detection as a three-way classification task (positive, negative, and neutral), type-2 errors are actually the same as type-1 errors. However, we believe that separating the two error types is more convenient with regard to following discussions.

Extrinsic Evaluation

With extrinsic evaluation, our goal is to measure the effectiveness of the different lexicons in the actual context of customer review mining. It further allows us to address errors of type-3 (missing entries). Take note that our main goal is to compare different lexicon construction approaches, rather than evaluating a complete lexicon-based customer review mining system. We thus mainly neglect challenges such as relation detection between sentiment expressions and targets or correct handling of sentiment shifters. Consequently, to consider lexicon quality in isolation, we design synthetic evaluation scenarios that have access to gold standard information (to a different extent). In particular, we consider the following two evaluation scenarios which both operate on the expression level corpora:

- **Scenario A** — Polarity classification when sentiment expressions S and related sentiment targets T are provided with perfect accuracy: Given a tuple (T, S) from the expression level corpus, the lexicon is used to look up the polarity value. Type-1 errors (misclassifications) produce a false positive for the predicted class and a false negative for the true class. Type-3 errors produce a false negative for the true class. Type-2 errors cannot occur in this scenario and thus cannot be measured. We use gold standard information with regard to sentiment targets and target-sentiment-relations. This allows us to examine the lexicon effectiveness in isolation from errors introduced by inaccurate target extraction or imprecise relation detection. The scenario gives us a realistic measure to compare the **recall** of different approaches and to assess their precision in terms of polarity classification.

- **Scenario B** — Polarity classification when the sentiment target T is provided with perfect accuracy, but sentiment expressions S and relations to targets must be algorithmically detected: Given a sentiment target T from the expression level corpus, we proceed heuristically by considering words in the same extraction unit as T (and occurring within a window of size δ_{window}) as potential sentiment expressions S and looking them up in the lexicon. Matches of type (T, S) or $(*, S)$ are used to define the sentiment polarity associated with T. If multiple matches are found, we simply sum up the corresponding polarity scores. Type-1 and type-3 errors may occur due to the same reasons as before. In addition they may be caused by an incorrect association of sentiment expressions with T. In this scenario type-2 errors may occur, so that we can measure **precision** more realistically than in scenario A.

Datasets and Corpora

- **Pros/cons datasets**: We use extended versions of the datasets described in Section 7.7. For the hotel domain, we use pros and cons parts of customer reviews that we extracted from the website Priceline.com. This dataset contains a sample of 150,000 reviews where pros and cons parts are both not empty. With regard to the digital camera domain, a similar sample of 100,000 reviews is extracted from the websites Epinions.com, Buzzillions.com, and Reevoo.com.

- **Test corpora**: The final results of the extrinsic evaluation are reported based on the gold standard annotations of the expression level hotel and digital camera corpora.

- **Set of polar seed words**: For our experiments with the label propagation approaches we use a set of words with unquestionable prior polarity (e.g., "good", "bad", "like", "love", "hate"). In total, the set consists of 120 polar words, where 54 words exhibit a positive polarity and 66 words a negative polarity. All words are tagged with part-of-speech information to guarantee a shallow word sense disambiguation. We cover the four major parts of speech, namely nouns, adjectives, adverbs, and verbs. The seed words are manually selected to best fit the domain of customer reviews. However, they are not optimized for any of the two specific target domains (hotel or digital camera reviews). For the label propagation approach by Blair-Goldensohn et al.

[40] an additional set of explicitly neutral words is needed. We adapted a standard stop word list by including some supplementing entries that have been proven useful during our experiments. The lists of seed words are presented in Appendix F.

Parameter Values

Table 9.2 summarizes all parameters used in the experiments and defines their values.

parameter	fixed value	component	description
λ	0.2	Blair-Goldensohn approach	decrease of influence with increasing path lengths in WordNet
M	5	Blair-Goldensohn approach	number of iterations
$\theta_{lrt-target}$	3.84	statistical assessment	minimum score to reject H_0 in the LRT-test for target-specific entries
$\theta_{lrt-domain}$	6.63	statistical assessment	minimum score to reject H_0 in the LRT-test for domain-specific entries
k_{domain}	750	domain lexicon generation	upper bound for the number of positive and negative expressions (each) included in the domain-specific lexicon
$\lambda_{expansion}$	0.2	lexicon expansion	decrease of influence with increasing path lengths in WordNet
$M_{expansion}$	3	lexicon expansion	number of iterations
θ_{score}	0.5	evaluation	minimum absolute (aggregated) polarity score for classifying a token sequence as polar
δ_{window}	5	evaluation	width of the window used in evaluation scenario B

Table 9.2.: Definition of parameter values for the experiments with sentiment lexicon construction.

9.4.2. Effectiveness of Extraction Patterns

In this section we take a closer look at the effectiveness of the different patterns we have designed for extracting tuples of sentiment expressions and product aspects from pros and cons texts. In particular, we consider the recall of each of the patterns within the different corpora. The precision of the patterns will be indirectly evaluated as part of subsequent sections. Table 9.3 lists the 9 high-precision patterns we use for our extractions. Two-character sequences, such as "RB", "NN", or "JJ", refer to part-of-speech tags of a token. The special character "A" refers to an identified product aspect, which may be a single token or a sequence of tokens. Patterns 4 and 8 also include a word (e.g., "no" or "too"). Such a pattern only matches if the relevant token equals the corresponding character sequence. Whereas Table 9.3 shows the generalized forms of patterns, Table 9.4 lists individual instances of the patterns. The table is sorted by the total occurrence frequency of the patterns in both pros/cons corpora. The column "pattern-id" refers to the "id" in Table 9.3 and indicates the corresponding generalized form of the pattern instance.

When analyzing the numbers in Table 9.3, the major observation is that a single pattern (pattern 1) accounts for far more than the half of all extractions. In the camera corpus the proportion is even nearly 80% of all extractions. Taking a closer look at the pattern instances, we find that it is the simple pattern $\texttt{JJ}_1\ \texttt{A}_1$ (i.e., an adjective preceding a product aspect) that accounts for nearly 50% of all extractions in both corpora. The very similar pattern 3 ($\texttt{JJ}_{1.1}\ \texttt{JJ}_{1.2}\ \texttt{A}_1$) accounts for additional

id	freq.	part-of-speech/token pattern	propn. camera	propn. hotel
1	167,913	[RB | DT] JJ_1 $A_{1.1}$ [(IN [DT] (A | NN)) | (CC ($A_{1.2}$ | NN | JJ))]	77.89%	55.25%
2	41,368	[DT] A_1 VB [RB] $JJ_{1.1}$ [CC [RB] $JJ_{1.2}$]	1.61%	29.02%
3	21,838	$JJ_{1.1}$ $J_{1.2}$ A_1	16.50%	2.12%
4	7,792	[EX VB] "no"$_1$ $A_{1.1}$ [(CC $A_{1.2}$) | VB]	2.66%	3.40%
5	4,981	A_1 [RB] $JJ_{1.1}$ [CC [RB] $JJ_{1.2}$]	0.86%	3.06%
6	4,770	(JJ | [DT] A | NN) CC JJ_1 A_1	0.44%	3.23%
7	4,493	[RB] JJ_1 A_1 CC [RB] JJ_2 A_2	0.39%	3.11%
8	1,296	[DT] A_1 [VB] ("too" JJ)$_1$	0.11%	0.86%
9	609	PR VB DT [RB] JJ_1 A_1	0.04%	0.43%
	255060		100%	100%

Table 9.3.: List of generalized high-precision extraction patterns used for the detection of sentiment expressions in pros and cons texts.

16.5% in the camera dataset, which together with pattern 1 is nearly 95% of all extractions. Within the hotel dataset the situation is different. The dominance of only a few patterns is less pronounced.

Thus, the second major observation is that the effectiveness of the individual patterns is to a large extent dependent on the underlying dataset. The pros and cons texts underlying the camera dataset are mostly incomplete sentences, simply enumerating the positive and negative aspects of the reviewed product. On the other hand, in the hotel pros/cons dataset extracted from Priceline.com, the ratio of grammatically correct sentences to simple enumerations is much higher. In consequence, more "elaborated" patterns, such as pattern 2, exhibit a higher relative frequency. In the hotel dataset this pattern accounts for roughly 30% of extractions, whereas in the camera corpus it is less than 2%.

In summary, we believe that the presented set of 9 high-precision patterns is generally applicable to different types of pros/cons datasets and allows for a reasonably high rate of extraction per pros/cons text. Taking the $2 * 150,000$ hotel pros and cons texts and the $2 * 100,000$ texts in the camera dataset, we have $500,000$ pros/cons in total. From these we were able to extract roughly $270,000$ tuples[15], which results in a rate of one extraction for every second pros/cons text.

[15] We found 255,060 matching patterns, but most of them allow for multiple extractions.

#	freq.	pattern-id	pattern instance	example	**extraction** (aspect, sentiment expr.)	propn. camera	propn. hotel
1	148,720	1	JJ_1 A_1	$short_1$ $(lag\ time)_1$	(lag time, short)	44.47%	48.59%
2	21,838	3	$JJ_{1.1}$ $JJ_{1.2}$ A_1	$free_{1.1}$ $hot_{1.2}$ $breakfast_1$	(breakfast, free); (breakfast, hot)	16.03%	2.02%
3	12,463	2	A_1 VB JJ_1	$(battery\ life)_1$ is $short_1$	(battery life, short)	0.74%	8.51%
4	10,687	2	DT A_1 VB JJ_1	the $location_1$ was $great_1$	(location, great)	0.34%	7.57%
5	7,374	1	RB JJ_1 A_1	not $many_1$ $features_1$	(features, many)	1.71%	3.93%
6	7,206	4	$"no"_1$ A_1	no_1 $(image\ stabilization)_1$	(image stabilization, no)	2,57%	3.05%
7	5,685	2	DT A_1 VB RB JJ_1	the $buttons_1$ are quite $small_1$	(buttons, small)	0.14%	4.06%
8	5,343	2	A_1 VB RB JJ_1	$lens_1$ is very $basic_1$	(lens, basic)	0.29%	3.68%
9	3,786	7	JJ_1 A_1 CC JJ_2 A_2	$compact_1$ $size_1$ and $light_2$ $weight_2$	(size, compact); (weight, light)	0.29%	2.53%
10	3,512	6	JJ CC JJ_1 A_1	clean and $attentive_1$ $staff_1$	(staff, attentive)	0.36%	2.27%
11	3,142	1	DT JJ_1 A_1	no $free_1$ $wifi_1$	(wifi, free)	0.27%	2.07%
12	2,851	1	JJ_1 $A_{1.1}$ CC $A_{1.2}$	$plastic_1$ $buttons_{1.1}$ and $casing_{1.2}$	(buttons, plastic); (casing, plastic)	0.32%	1.82%
13	2,339	5	A_1 JJ_1	$elevators_1$ $slow_1$	(elevators, slow)	0.51%	1.27%
14	2,026	5	A_1 RB JJ_1	$price_1$ very $high_1$	(price, high)	0.30%	1.23%
15	1,885	2	DT A_1 VB RB $JJ_{1.1}$ CC $JJ_{1.2}$	the $room_1$ was very $quiet_{1.1}$ and $clean_{1.2}$	(room, quiet); (room, clean)	0.01%	1.38%
16	1,837	2	DT A_1 VB $JJ_{1.1}$ CC $JJ_{1.2}$	the $colors_1$ are $true_{1.1}$ and $natural_{1.2}$	(colors, true); (colors, natural)	0.02%	1.33%
17	1,693	2	A_1 VB $JJ_{1.1}$ CC $JJ_{1.2}$	$flash_1$ is $minimal_{1.1}$ or $inadequate_{1.2}$	(flash, minimal); (flash, inadequate)	0.04%	1.21%
18	1,145	1	JJ_1 A_1 IN NN	$questionable_1$ $value_1$ for money	(value, questionable)	0.39%	0.50%
19	1,088	1	JJ_1 A_1 IN A_2	$mono_1$ $(sound\ recording)_1$ in $video_2$	(sound recording, mono)	0.19%	0.63%
20	1,068	2	A_1 VB RB $JJ_{1.1}$ CC $JJ_{1.2}$	$towels_1$ were so $hard_{1.1}$ and $rough_{1.2}$	(towels, hard); (towels, rough)	0.01%	0.78%
21	954	5	A_1 CC JJ_2 A_2	$location_1$ and $fast_2$ & $(check\ in)_2$	(check in, fast)	0.06%	0.65%
22	723	1	JJ_1 A_1 IN DT A_2	$spotty_1$ $(internet\ service)_1$ in the $room_2$	(internet service, spotty)	0.11%	0.43%
23	570	1	JJ_1 A_1 IN DT NN	$minimal_1$ $parking_1$ on the street	(parking, minimal)	0.11%	0.32%
24	550	8	A_1 VB $("too"\ JJ)_1$	bed_1 was $(too\ soft)_1$	(bed, too soft)	0.03%	0.38%
25	532	1	JJ_1 A_1 CC NN	$noisy_1$ $aircon_1$ and street	(aircon, noisy)	0.07%	0.33%
26	511	1	JJ_1 A_1 CC JJ	$lightweight_1$ $construction_1$ but sturdy	(construction, lightweight)	0.07%	0.32%
27	506	9	PR VB DT JJ_1 A_1	it has a $short_1$ $(battery\ life)_1$	(battery life, short)	0.03%	0.34%
28	476	8	A_1 $("too"\ JJ)_1$	$pool_1$ $(too\ small)_1$	(pool, too small)	0.78%	0.28%
29	334	7	RB JJ_1 A_1 CC JJ_2 A_2	very $quiet_1$ $rooms_1$ & $fast_2$ $internet_2$	(rooms, quiet); (internet, fast)	0.04%	0.21%
30	332	7	JJ_1 A_1 CC RB JJ_2 A_2	$many_1$ $features_1$ & very low_2 $noise_2$	(features, many); (noise, low)	0.04%	0.21%

Table 9.4.: The 30 most frequent instances of the nine extraction patterns presented in Table 9.3. The pattern id refers to the corresponding generalized pattern.

9.4.3. Comparison of Baseline Approaches

In this section, our goal is to compare several approaches and dictionaries that set up a baseline for our following experiments. In particular, we consider the two label propagation methods by Rao et al. [312] and Blair-Goldensohn et al. [40], as well as two manually compiled sentiment lexicons. For the two label propagation methods, we report results for the parameter sets that we have found to perform best in preliminary experiments (see Table 9.2). Further, in contrast to the original approaches, we use additional WordNet relations including "similar-to", "derivationally related", and "see also". Anticipating results presented in Section 9.4.6, we found that such a configuration leads to improved results.

The lexicon, which we here denote as "MPQA", is an excerpt of the *MPQA Subjectivity Lexicon* [438]. Out of the original lexicon we extract those terms that are marked as "strongly subjective" and which exhibit either a positive or negative prior polarity. Further removing some stemmed forms, results in a lexicon of 4,422 entries (the original lexicon consisted of roughly 8,700 entries, but led to inferior results in comparison). The "Liu" sentiment lexicon[16] is a dictionary that has been created specifically for the task of customer review mining. It consists of roughly 6,800 entries, explicitly containing many misspellings and colloquial expressions that are frequently used in user generated content. The lexicon has been created with the help of automatic methods (synonym/antonym expansion), but was manually revised and extended over a period of many years. In addition to the four baseline dictionaries, we consider using the set of polar seed words alone, that is, without further expansion by means of label propagation in WordNet. The configuration is denoted as "polar-seed-words" or shortly "seed" in the following. Configurations prefixed with "BG" indicate that label propagation with the Blair-Goldensohn et al. method is used. Analogously, the prefix "Rao" is used.

Extrinsic Evaluation

Tables 9.5 and 9.6 present the results of the extrinsic evaluation for the hotel and digital camera corpora. Each table differentiates between the precision, recall, and f-measure values for the positive and negative polarity classes and further provides the macro-averaged results computed over both classes.

The first observation is that the Rao et al. method (Rao) performs consistently worse than the Blair-Goldensohn et al. method (BG). The differences in macro-averaged f-measure are 15-25 percentage points in both scenarios and in both domains (hotel and camera). Whereas the precision is very high with the Rao method (> 90%), its recall is comparably low at around 40%. The main reason is that with the Rao method the resulting lexicon is much smaller than the lexicon obtained with the BG approach. The first method expands the set of polar seed words to a lexicon of around 800 entries, whereas the second method generates roughly ten times more entries ($\sim$ 7,500). Besides the distinct ways of implementing label propagation, another cause for the smaller lexicon sizes with the Rao method may be that it does not directly allow to incorporate the antonym relation.

The second main observation is that the f-measure is significantly higher for the positive polarity class than for the negative class. We find this result for all approaches in all scenarios and both domains. The average difference between the f-measure obtained for the positive and negative class is about 20 percentage points. While both precision and recall are lower for the negative polarity class, it is primarily the lower recall causing inferior results. Looking at the false negatives of both classes, we find that in the negative class the ratio of complex sentiment expression (in contrast to single words) is higher. Such complex expressions (e.g., "left something to be desired" or "should come with") are lacking in the considered sentiment lexicons. Closely related is another reason. Recall that our corpus analysis (cf., Section 6.2.2) revealed that the lexical diversity of negative sentiment

[16] See http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html for some further explanations.

		positive			negative			macro-average		
Lexicon	Size	P	R	F1	P	R	F1	P	R	F1
polar-seed-words	120	**0.991**	0.433	0.602	**1.000**	0.256	0.408	**0.996**	0.344	0.505
BG-seed	2850	0.955	0.659	0.780	0.981	0.438	0.606	0.968	0.548	0.693
Rao-seed	776	0.974	0.497	0.658	**1.000**	0.273	0.429	0.987	0.385	0.543
MPQA	4422	0.981	0.454	0.621	0.891	0.270	0.414	0.936	0.362	0.517
Liu	6789	0.964	**0.749**	**0.843**	0.938	**0.501**	**0.654**	0.951	**0.625**	**0.748**

(a) Scenario A

		positive			negative			macro-average		
Lexicon	Size	P	R	F1	P	R	F1	P	R	F1
polar-seed-words	120	**0.939**	0.534	0.681	0.915	0.250	0.393	**0.927**	0.392	0.537
BG-seed	2850	0.892	**0.788**	0.837	0.833	0.383	0.525	0.862	0.586	0.681
Rao-seed	776	0.921	0.587	0.717	**0.919**	0.263	0.409	0.920	0.425	0.563
MPQA	4422	0.934	0.544	0.687	0.752	0.273	0.401	0.843	0.409	0.544
Liu	6789	0.922	0.782	**0.846**	0.873	**0.437**	**0.582**	0.898	**0.609**	**0.714**

(b) Scenario B

Table 9.5.: Hotel dataset: Comparison of the results for the baseline approaches.

expressions is generally higher than for positive expressions (higher root-ttr). In consequence, the task of recognizing negative sentiment is more difficult than recognizing positive sentiment.

Comparing both handcrafted sentiment lexicons, we find that the Liu lexicon by far outperforms the MPQA lexicon. The macro-averaged f-measure is roughly 15 to 25 percentage points higher in the various evaluation configurations. Again, it is mainly the improved recall which causes this difference. It is obvious that the Liu lexicon was primarily designed for sentiment analysis in customer review data, whereas the MPQA lexicon was mainly devised for subjectivity detection in newswire text. These basic results thus further pinpoint the need for domain specific lexicons. A closely related observation is that the Liu lexicon performs better on the digital camera dataset than on the hotel dataset, whereas it is vice versa for the BG approach. We assume that here the reason is that the Liu lexicon was designed and tested in the context of customer reviews on consumer electronics (including digital cameras). In consequence, it contains many entries that are especially relevant in this specific domain (e.g., "easy-to-use", "clear", or "durable").

Among the baseline approaches, the handcrafted Liu lexicon and the automatically created BG-seed lexicon exhibit the best performance. Comparing both configurations, the results are similar with macro-averaged f-measures of approximately 75-80% in scenario A and 70-75% in scenario B. In both scenarios and both domains, the recall obtained with the automatically created lexicon is significantly higher (trading for precision), which is mainly a result of the larger lexicon size (7,500 compared to 6,800 entries). The BG label propagation approach successfully expands the seed set of 120 polar words. Compared to using only the seed words as a lexicon, the f-measure is increased by roughly 15-30 percentage points in the different evaluation configurations. Interestingly, we can achieve better (camera) or similar (hotel) results with the small set of polar seed words compared to using the much larger MPQA lexicon.

A further observation is that even with the relative simple approach of computing polarity scores by matching lexicon entries in a co-occurrence window of (albeit perfectly known) product aspects (scenario B), we can achieve quite good results with f-measure values of up to 75%. For the positive polarity class the f-measure is even up to 85%. Naturally, results for scenario A are generally better than for scenario B as type-2 errors cannot be counted in scenario A. Thus, with regard to precision, scenario B shows the more realistic results. Concerning recall, scenario A is more realistic, as no influ-

		positive			negative			macro-average		
Lexicon	Size	P	R	F1	P	R	F1	P	R	F1
polar-seed-words	120	**0.994**	0.564	0.720	0.989	0.275	0.430	**0.992**	0.420	0.575
BG-seed	2850	0.975	0.702	0.816	0.845	0.411	0.553	0.910	0.556	0.685
Rao-seed	776	0.979	0.564	0.716	**0.989**	0.284	0.441	0.984	0.424	0.579
MPQA	4422	0.970	0.468	0.631	0.850	0.275	0.416	0.910	0.371	0.523
Liu	6789	0.958	**0.733**	**0.830**	0.926	**0.644**	**0.759**	0.942	**0.688**	**0.795**

(a) Scenario A

		positive			negative			macro-average		
Lexicon	Size	P	R	F1	P	R	F1	P	R	F1
polar-seed-words	120	**0.949**	0.587	0.726	0.932	0.242	0.384	**0.941**	0.415	0.555
BG-seed	2850	0.908	0.693	0.786	0.631	0.396	0.487	0.770	0.545	0.636
Rao-seed	776	0.924	0.597	0.726	**0.934**	0.249	0.393	0.929	0.423	0.559
MPQA	4422	0.916	0.509	0.655	0.748	0.270	0.397	0.832	0.390	0.526
Liu	6789	0.922	**0.728**	**0.813**	0.891	**0.516**	**0.653**	0.907	**0.622**	**0.733**

(b) Scenario B

Table 9.6.: Camera dataset: Comparison of the results for the baseline approaches.

ence from potentially erroneous relation detection is incorporated. In summary, extrinsic evaluation shows that the BG method and the Liu lexicon exhibit a similar performance and that both clearly outperform the other baseline approaches. Label propagation with the Rao method led to significantly worse results than with the BG method. Following experiments therefore only cover the BG method as representative for label propagation.

Intrinsic Evaluation

Table 9.7 presents the results of our intrinsic evaluation for the BG and the Rao method. Recall that we evaluate the generated lexicons by sampling from different partitions and inspecting the corresponding entries. We can measure type-1 (false polarity) and type-2 errors (non polar entry) and thus can report an estimate for the precision of the different partitions of the lexicon. We observe

	positive			negative			
statistic	sample top	sample mid	sample bottom	sample top	sample mid	sample bottom	average
type-1 errors	4	3	4	2	1	1	2.5
type-2 errors	9	20	21	10	10	16	14.33
estimated precision	0.74	0.54	0.50	0.76	0.78	0.66	0.66
lexicon sizes		2505			5011		—

(a) Blair-Goldensohn method

	positive			negative			
statistic	sample top	sample mid	sample bottom	sample top	sample mid	sample bottom	average
type-1 errors	0	4	5	1	0	0	1.67
type-2 errors	6	16	20	13	17	13	14.17
estimated precision	0.88	0.60	0.50	0.72	0.66	0.74	0.68
lexicon sizes		324			564		—

(b) Rao method

Table 9.7.: Results for the intrinsic evaluation of the baseline approaches.

relative low precision values for the BG method, which may be astonishing when considering the comparably high precision obtained with extrinsic evaluation. From the set of 300 (6*50) randomly sampled lexicon entries we estimate an overall precision of 66%. The precision is higher (at 75%), when only considering the top third of the positive and negative lexicon entries (i.e., with highest absolute score). For the bottom third it is around 20 percentage points lower, namely at 58%. We find that the main reason for low precision is a relatively high number of type-2 errors. The lexicon contains many entries that we annotated as not having any prior sentiment polarity. The reason why the lexicon still performs very good in the sentiment polarity detection task of our extrinsic evaluation scenario is twofold: First, the most frequent sentiment expressions (which are not already contained in the seed word list) are correctly identified and classified by the algorithm (we have relatively few type-1 errors). Second, the vast majority of false (i.e., neutral) entries are very uncommon words that simply do not occur very often in our evaluation datasets (and even more rarely in the direct context of a product aspect). Sorted in descending order by absolute polarity score, we observe for example the following false entries: "collateral", "powdered", "fixture", "develop", or "lubricated oil".

Looking at the WordNet entries for this type of entries, we find mainly two reasons that cause their erroneous inclusion. Often, the error stems from disregarding the different senses of a word (recall that we fold the different senses/synsets related to a word to a single sense). For instance, the word "collateral" is included as there exists a sense in WordNet of type "serving to support or corroborate" with synonyms "confirmative", "confirmatory" "substantiating", "substantiative", and more. From these synonyms there exists a path of length two to the seed word "positive". Further, for this particular sense, many other positive words are reachable with short paths, for instance, the words "helpful", "heartening", or "encouraging". For the other exemplary words such paths exist, too (e.g., for "powdered" we find a direct relation to "fine", which is a positive seed word). Another reason is that words that have many connections in WordNet are favored by the Blair-Goldensohn method. As the score for a word is derived by summing up the scores obtained through incoming edges, non-polar words with many different senses or many synonyms may receive high scores just by the large number of (individually low-scored) links.

For the Rao method, we observe a slightly higher average precision of 68%[17]. The distribution of the two error types is quite similar in comparison to the BG method. The average number of type-2 errors is 14.17 in each 50-word partition, compared to 14.33 for the BG method. Type-1 errors are equally low for both methods with around 2 errors in average. In addition, for both methods we can observe a correlation between the calculated polarity scores and the estimated precision. Higher scores correlate with higher precision values — we have fewer errors in the partitions sampled from the entries with highest absolute polarity scores.

9.4.4. Incorporating Domain and Target-Specific Lexicon Entries

Extrinsic Evaluation

In this section, we discuss the results with gathering domain and target-specific sentiment expressions from the pros and cons parts of customer reviews (as described in Section 9.3). In particular, we consider configurations where the domain and target-specific expressions are incorporated into the seed word list, the original Liu lexicon, and the lexicon obtained with the BG approach. The results are presented in Tables 9.8 and 9.9. The tables show macro-averaged results for each configuration in both evaluation scenarios. The three original (unextended) lexicons "seed", "BG-seed", and "Liu" are taken as references for the results obtained with the extended, domain specific lexicons. The notation with the "+" symbol states that entries of a certain lexicon are added to another one. For instance the configuration "BG-seed+dom+targ" refers to the setting where we construct a lexicon from the seed words by label propagation with the BG method and then add to this lexicon the extracted domain-

[17] Recall that the precision values are only estimated from a sample of 300 lexicon entries.

		scenario A			scenario B		
lexicon	size	macro-p	macro-r	macro-f1	macro-p	macro-r	macro-f1
seed	120	**0.996**	0.344	0.505	**0.927**	0.392	0.537
seed+dom	1590	0.894 (-0.102)	0.806 (+0.461)	0.847 (+0.342)	0.854 (-0.073)	0.701 (+0.309)	0.770 (+0.233)
seed+targ	4227	0.944 (-0.052)	0.649 (+0.305)	0.767 (+0.261)	0.867 (-0.060)	0.610 (+0.218)	0.711 (+0.174)
seed+dom+targ	4712	0.903 (-0.093)	0.810 (+0.466)	0.854 (+0.349)	0.844 (-0.083)	0.733 (+0.341)	0.784 (+0.247)
BG-seed	7517	0.881	0.726	0.796	0.808	0.703	0.752
BG-seed-dom	23437	0.881 (+0.000)	0.897 (+0.171)	0.888 (+0.092)	0.795 (-0.013)	0.795 (+0.093)	0.795 (+0.044)
BG-seed+dom	8428	0.891 (+0.010)	0.851 (+0.125)	0.870 (+0.074)	0.821 (+0.013)	0.757 (+0.055)	0.788 (+0.036)
BG-seed+targ	10819	0.910 (+0.029)	0.809 (+0.082)	0.856 (+0.060)	0.831 (+0.023)	0.759 (+0.057)	0.793 (+0.042)
BG-seed-dom+targ	25753	0.890 (+0.009)	**0.903** (+0.176)	**0.895** (+0.100)	0.803 (-0.005)	**0.805** (+0.103)	0.804 (+0.052)
BG-seed+dom+targ	11195	0.908 (+0.028)	0.865 (+0.138)	0.886 (+0.090)	0.833 (+0.025)	0.787 (+0.084)	**0.809** (+0.057)
Liu	6789	0.951	0.625	0.748	0.898	0.609	0.714
Liu+dom	7687	0.901 (-0.050)	0.861 (+0.236)	0.880 (+0.132)	0.863 (-0.034)	0.739 (+0.129)	0.796 (+0.082)
Liu+targ	10421	0.936 (-0.015)	0.797 (+0.172)	0.860 (+0.112)	0.862 (-0.036)	0.721 (+0.111)	0.783 (+0.069)
Liu+dom+targ	10802	0.910 (-0.042)	0.865 (+0.240)	0.887 (+0.139)	0.851 (-0.047)	0.770 (+0.161)	0.808 (+0.094)

Table 9.8.: Hotel corpus: Results obtained by incorporating domain and target-specific sentiment expressions into the baseline lexicons "seed", "BG-seed", and "Liu". The table shows the macro-averaged results for polarity classification in scenarios A and B.

specific and the target-specific entries. The notation with the "-" symbol as in "BG-seed-dom" refers to a setting where the domain-specific words are used as additional seeds within the BG method. Numbers in brackets refer to the differences calculated in comparison to the respective reference lexicon.

The first and main result is that our approach to exploit the weakly labeled pros/cons data leads to significantly improved results compared to all baseline approaches. This result is consistent with regard to the two evaluation scenarios as well as with respect to both product domains.

Generally, the improved f-measure is due to a significantly higher recall. When comparing the configurations where we either add only the domain-specific or the target-specific sentiment expressions, we find that the major share of improved recall is due to the addition of the domain-specific expressions. For instance, considering the configuration with the Liu lexicon in scenario A, we observe increases in recall of 23.6 (hotel) and 14.3 (camera) percentage points when adding the domain-specific expressions. Increases in recall are less when adding only the target-specific entries. Here, we measure improvements of "only" 17.2 (hotel) and 7.3 (camera) percentage points. The best results are obtained when adding both, domain-specific and target-specific sentiment expressions. Again considering the Liu lexicon, we observe improvements in f-measure of 13.9 (hotel) and 8.5 (camera) percentage points in scenario A. For the evaluation scenario B, the numbers are +9.4 and +6.5 percentage points. When considering the BG-seed baseline, the improvements are slightly less with +9.0 and +8.9 percentage points in scenario A and +5.7 and +7.8 percentage points in scenario B. The main reason for the less steep increase is that the recall achieved with the BG-seed baseline is already (comparably) high.

Comparing the three baseline approaches, we find that extending the Liu lexicon shows the best results (in f-measure) for the camera datasets and the BG-seed method is slightly better for the hotel dataset. In scenario A, we obtain results of nearly 90% f-measure and in scenario B the f-measure is still high with around 80%. Expanding the basic seed word list with the extracted domain and target-specific sentiment expressions also shows astonishingly good results. The f-measure is at maximum only 5 percentage points lower compared to the corresponding best configuration with the other two methods.

We further experimented with incorporating the extracted domain-specific sentiment expressions as additional seed words for the label propagation approach (instead of simply adding them to an existing lexicon). The four configurations "BG-seed-dom" and "BG-seed+dom", as well as "BG-seed-

		scenario A			scenario B		
lexicon	size	macro-p	macro-r	macro-f1	macro-p	macro-r	macro-f1
seed	120	**0.992**	0.420	0.575	**0.941**	0.415	0.555
seed+dom	1596	0.912 (-0.079)	0.739 (+0.320)	0.816 (+0.241)	0.858 (-0.083)	0.641 (+0.227)	0.733 (+0.178)
seed+targ	2075	0.969 (-0.022)	0.569 (+0.150)	0.710 (+0.135)	0.910 (-0.031)	0.554 (+0.139)	0.680 (+0.125)
seed+dom+targ	3091	0.930 (-0.062)	0.748 (+0.328)	0.827 (+0.252)	0.867 (-0.074)	0.679 (+0.265)	0.760 (+0.205)
BG-seed	7517	0.854	0.707	0.773	0.780	0.660	0.715
BG-seed-dom	21840	0.877 (+0.023)	0.846 (+0.139)	0.861 (+0.088)	0.779 (-0.001)	0.747 (+0.087)	0.763 (+0.048)
BG-seed+dom	8597	0.881 (+0.027)	0.803 (+0.095)	0.840 (+0.067)	0.796 (+0.016)	0.720 (+0.059)	0.755 (+0.041)
BG-seed+targ	9049	0.902 (+0.048)	0.766 (+0.058)	0.828 (+0.054)	0.829 (+0.050)	0.729 (+0.068)	0.776 (+0.061)
BG-seed-dom+targ	22882	0.890 (+0.036)	**0.850** (+0.143)	0.869 (+0.096)	0.810 (+0.030)	**0.777** (+0.116)	0.793 (+0.078)
BG-seed+dom+targ	9863	0.907 (+0.053)	0.822 (+0.114)	0.862 (+0.089)	0.824 (+0.045)	0.763 (+0.103)	0.792 (+0.078)
Liu	6789	0.942	0.688	0.795	0.907	0.622	0.733
Liu+dom	7845	0.910 (-0.032)	0.832 (+0.143)	0.869 (+0.074)	0.866 (-0.041)	0.702 (+0.080)	0.775 (+0.042)
Liu+targ	8540	0.939 (-0.003)	0.761 (+0.073)	0.840 (+0.046)	0.896 (-0.010)	0.686 (+0.064)	0.775 (+0.042)
Liu+dom+targ	9340	0.923 (-0.019)	0.840 (+0.152)	**0.879** (+0.085)	0.868 (-0.038)	0.739 (+0.118)	**0.798** (+0.065)

Table 9.9.: Camera corpus: Results obtained by incorporating domain and target-specific sentiment expressions into the baseline lexicons "seed", "BG-seed", and "Liu". The table shows the macro-averaged results for polarity classification in scenarios A and B.

dom+targ" and "BG-seed+dom+targ" represent these differing approaches. Our results are that utilizing the domain-specific words as additional seeds leads to slightly better results with regard to recall. Take note that the lexicons obtained with "BG-seed-dom(+targ)" configurations are supersets of "BG-seed+dom(+targ)" lexicons (Algorithm 9.1 ensures that label propagation does not remove any seed word). With regard to precision, we consider the more realistic scenario B. We find that precision is slightly worse for the "add as seed" approach, showing that label propagation adds some noise to the lexicons. Also, when adding target-specific entries to the lexicons, no significant difference in f-measure is observable in scenario B. Additionally considering the fact that the resulting lexicon is only half of the size, we conclude that simply adding the domain-specific words to the lexicon is preferable over utilizing the words as additional seeds.

Generally, when considering the handcrafted baselines ("seed", "Liu"), we can observe that adding the extracted domain and target-specific entries decreases the precision. This result was foreseeable as the heuristically extracted data naturally contains some noise in terms of type-1 (false polarity) and type-2 errors (non polar entry). However, the decrease in precision is rather moderate compared to the increase in recall. We still observe high absolute precision values of over 90% in scenario A and 85-90% in scenario B. We conclude from these results that our approach of gathering domain and target-specific sentiment expression exhibits a high accuracy (see also the following intrinsic evaluation). When considering the BG-seed baseline, we even observe an increase in precision with added domain and target-specific expressions. Here, the additional data is capable of compensating mistakes introduced by the label propagation procedure.

The provided results in Tables 9.8 and 9.9 let us also analyze the relative importance of considering target-specific prior polarity in comparison to domain-specific polarity. For this purpose, we consider the differences when adding target-specific entries to a lexicon that already contains domain-specific entries (e.g., Liu+dom compared to Liu+dom+targ). We observe improvements in f-measure of roughly 2-4 percentage points when considering scenario B for the different configurations and both domains. We can conclude that our approach successfully determines target-specific polarity and further that improvements achievable with target-specific lexicons are at least 2-4 percentage points in f-measure. The improvements are slightly more pronounced for the digital camera evaluation corpus.

	positive			negative			
statistic	sample top	sample mid	sample bottom	sample top	sample mid	sample bottom	average
type-1 errors	0	1	0	0	1	1	0.5
type-2 errors	2	10	6	5	5	6	5.67
estimated precision	0.96	0.78	0.88	0.90	0.88	0.86	0.88
lexicon sizes		750			750		—

(a) Hotel dataset

	positive			negative			
statistic	sample top	sample mid	sample bottom	sample top	sample mid	sample bottom	average
type-1 errors	0	1	0	0	0	2	0.5
type-2 errors	6	7	9	5	14	9	8.33
estimated precision	0.88	0.84	0.82	0.90	0.72	0.78	0.82
lexicon sizes		750			750		—

(b) Digital camera dataset

Table 9.10.: Accuracy of the approach for extracting domain-specific sentiment lexicon entries.

Intrinsic Evaluation

We have seen that including the extracted domain and target-specific sentiment expressions allows for a major improvement in recall while precision only slightly deteriorates. We now consider precision more closely by manually inspecting and verifying the extracted sentiment expressions. As described in Section 9.4.1, we randomly sample from six different partitions of the data. Table 9.10 shows the result of our intrinsic evaluation for the domain-specific part, whereas Table 9.11 presents the results for the extraction of target-specific entries. For the extraction of domain-specific sentiment expressions we observe high (estimated) precision values of 88% for the hotel dataset and 82% for the digital camera dataset. Especially type-1 errors (false polarity) are very rare with on average 0.5 occurrences in a 50-word partition in both evaluation datasets. Considering only the two top partitions, the precision is even at 93% (hotel) and 89% (camera), respectively. Looking more closely at type-2 errors, we find that they either stem from errors of the part-of-speech tagger[18] or from missing entries in the product type taxonomy. Missing entries may lead to mistakes if a modifier, which is actually part of the term, is erroneously identified as a sentiment expression. In general, we can conclude that our heuristics to derive domain-specific sentiment expressions from the extracted target-sentiment tuples are sufficiently accurate.

Evaluating the target-sentiment tuples themselves shows an even higher precision of our extraction and scoring process. In both datasets, we observe very high precision values of 95% (hotel) and 94% (camera), respectively. Comparing positive to negative partitions of the extracted lexicons, we find that the precision for extracting negative tuples is higher with an (estimated) perfect precision of 100% for the hotel dataset and 98% for the camera dataset. Type-1 errors are very unlikely to occur. In the combined sample of 600 lexicon entries (hotel and camera) we find only a single entry with false polarity ("simple decoration" is estimated as positive). The majority of type-2 errors either refers to senseless combinations, such as (tv, clean), (room size, clean), etc., or misinterpreted term modifiers such as (tv, flatscreen) or (parking, underground). Whereas the former errors are unlikely to affect precision in a real application, the latter may in fact decrease the accuracy. In summary, we can conclude that our pattern-based methods (in conjunction with the statistical assessment) indeed constitute high-precision extraction heuristics that allow to generate highly accurate target-specific supplements to general purpose sentiment lexicons.

[18]The tagger was not explicitly trained on the short pros/cons texts.

	positive			negative			
statistic	sample top	sample mid	sample bottom	sample top	sample mid	sample bottom	average
type-1 errors	0	1	0	0	0	0	0.17
type-2 errors	3	4	6	0	0	0	2.17
estimated precision	0.94	0.90	0.88	1.00	1.00	1.00	0.95
lexicon sizes		1993			2149		—

(a) Hotel dataset

	positive			negative			
statistic	sample top	sample mid	sample bottom	sample top	sample mid	sample bottom	average
type-1 errors	0	0	0	0	0	0	0.00
type-2 errors	3	3	8	0	1	2	2.83
estimated precision	0.94	0.94	0.84	1.00	0.98	0.96	0.94
lexicon sizes		1275			701		—

(b) Digital camera dataset

Table 9.11.: Accuracy of the approach for extracting target-specific sentiment lexicon entries.

9.4.5. Effectiveness of Expansion Strategies

The goal of this section is to analyze the effectiveness of the two different strategies for expanding target-specific sentiment lexicons (as described in Section 9.3.5). We compare strategies A and B and use as reference the "BG-seed+dom+targ" configuration, which does not expand target-specific entries. Table 9.12 presents our results obtained for the hotel and digital camera datasets. The single

		scenario A			scenario B		
lexicon	size	macro-p	macro-r	macro-f1	macro-p	macro-r	macro-f1
BG-s+d+t	11195	**0.908**	**0.865**	**0.886**	**0.833**	**0.787**	**0.809**
BG-s+d+t-exp-A	12592	**0.908** (+0.000)	**0.865** (+0.000)	**0.886** (+0.000)	**0.833** (+0.000)	**0.787** (+0.000)	**0.809** (+0.000)
BG-s+d+t-exp-B	12253	**0.908** (+0.000)	**0.865** (+0.000)	**0.886** (+0.000)	0.832 (-0.001)	0.785 (-0.002)	0.808 (-0.001)

(a) hotel corpus

		scenario A			scenario B		
lexicon	size	macro-p	macro-r	macro-f1	macro-p	macro-r	macro-f1
BG-s+d+t	9863	0.907	0.822	0.862	0.824	**0.763**	0.792
BG-s+d+t-exp-A	10374	0.908 (+0.002)	**0.822** (+0.001)	**0.863** (+0.001)	0.825 (+0.001)	**0.763** (+0.000)	**0.793** (+0.000)
BG-s+d+t-exp-B	10434	**0.909** (+0.002)	0.821 (-0.000)	0.863 (+0.001)	**0.826** (+0.002)	0.762 (-0.001)	0.793 (+0.000)

(b) digital camera corpus

Table 9.12.: Effectiveness of the two different strategies for the expansion of target-specific sentiment lexicons.

and main observation is that neither strategy A, nor strategy B lead to improved results. The idea to apply label propagation to further expand a target-specific sentiment lexicon has proven useless, at least for our evaluation corpora. In both evaluation scenarios and in both product domains differences in f-measure are not significant. Although the size of the lexicons is slightly expanded with the approach (roughly +500-1500 entries), no significant influence on recall can be measured. More closely inspecting the really target-specific sentiment expressions in our corpora shows that most of them describe (physically) quantifiable properties such as size, length, weight, speed, temperature, or brightness. We further find that the lexical variability to describe such properties is rather low. For instance, to describe the size of an entity the vast majority of reviewers refers to only six adjectives

		scenario A			scenario B		
lexicon	size	macro-p	macro-r	macro-f1	macro-p	macro-r	macro-f1
BG-s-syn	2847	0.876	0.546	0.673	0.776	0.578	0.663
BG-s-sim	4752	0.843 (-0.033)	0.651 (+0.105)	0.735 (+0.062)	0.777 (+0.001)	0.663 (+0.085)	0.715 (+0.053)
BG-s-also	3236	0.872 (-0.004)	0.608 (+0.061)	0.715 (+0.042)	0.780 (+0.005)	0.614 (+0.036)	0.687 (+0.024)
BG-s-deriv	4184	**0.937** (+0.060)	0.597 (+0.051)	0.728 (+0.055)	0.789 (+0.013)	0.635 (+0.056)	0.703 (+0.041)
BG-s-hyp	16255	0.830 (-0.046)	0.626 (+0.080)	0.714 (+0.041)	0.671 (-0.105)	0.701 (+0.123)	0.681 (+0.019)
BG-s-all\hyp	7517	0.881 (+0.004)	0.726 (+0.180)	0.796 (+0.123)	0.808 (+0.032)	0.703 (+0.124)	0.752 (+0.089)
BG-s+d+t-syn	7059	0.899	0.843	0.870	0.823	0.770	0.796
BG-s+d+t-sim	8604	0.896 (-0.003)	0.845 (+0.002)	0.870 (-0.000)	0.816 (-0.007)	0.776 (+0.006)	0.795 (-0.000)
BG-s+d+t-also	7342	0.905 (+0.006)	0.845 (+0.003)	0.874 (+0.004)	0.831 (+0.008)	0.774 (+0.003)	0.801 (+0.005)
BG-s+d+t-deriv	8338	0.904 (+0.005)	0.856 (+0.013)	0.879 (+0.009)	0.827 (+0.004)	0.784 (+0.013)	0.805 (+0.009)
BG-s+d+t-hyp	20384	0.878 (-0.021)	**0.875** (+0.032)	0.876 (+0.007)	0.775 (-0.048)	**0.814** (+0.044)	0.794 (-0.002)
BG-s+d+t-all\hyp	11195	0.908 (+0.009)	0.865 (+0.022)	**0.886** (+0.016)	**0.833** (+0.010)	0.787 (+0.016)	**0.809** (+0.013)

Table 9.13.: Hotel corpus: Effectiveness of different sets of WordNet relations for the label propagation approaches.

("small", "little", "large", "big", "tiny", and "huge"). However, nearly all of these types of adjectives are already included within our extracted target-specific lexicon. Thus, further expanding it does not lead to better results.

9.4.6. Influence of Considered WordNet Relations

We now analyze the effect of utilizing different WordNet relations for the label propagation algorithms. In particular, we consider using the *similar-to*, the *see-also*, and the *hyponymy* relation as well as the "cross part-of-speech" relation called "*derivationally related form*". For the hyponymy relation, we add directed edges to the (virtual) graph that represents the relevant WordNet relations (in fact, we adjust the stochastic adjacency matrix accordingly). The intuition is that sentiment polarity is transferred from a word to its hyponym (subtype), but not from a word to its hypernym (supertype). With regard to sentiment polarity, we believe that the relation is unidirectional. For instance, the word "imperfection" transfers its negative polarity to its direct hyponyms, such as "flaw", "defect", "weakness", or "fault", but not to its direct hypernym which is "state" in WordNet. In contrast, we regard the other considered relations as bidirectional. The relations are symmetric and we thus postulate that sentiment polarity is transferred in either direction. The similar-to and the see-also relations generally link semantic "near synonyms". As an example, the similar-to relation links the adjective "polite" to adjectives such as "courteous", "gracious", "well-mannered", or "mannerly". The similar-to relation has also been used in other approaches [120, 395] and we too believe that the relation transfers the sentiment polarity of words. Regarding the "derivationally related form" relation we cite WordNet's website[19]: The relation includes "'morphosemantic' links that hold among semantically similar words sharing a stem with the same meaning [...]". For instance, the relation links the verb "to love" with the corresponding adjective "lovable" and noun "love". As a baseline, we use the original configuration of the Blair-Goldensohn et al. approach, which only considers the synonym/antonym relation. For our experiments, we separately incorporate each of the other relations as supplement to this basic configuration. In addition, we also report results for a combination of all relations (excluding the hyponymy relation), which is denoted as "BG-s+all\hyp" in the following.

Tables 9.13 and 9.14 present our results with additionally incorporating the mentioned relations. We analyze the effect when using the BG method alone, as well as the influence when the resulting lexicon is extended with domain and target-specific entries (configurations with prefix "BG-s+d+t").

As a first result, when considering the obtained f-measure, we find that all relations, except for the hyponymy relation, lead to improved results. The increases are more pronounced for the con-

[19] http://wordnet.princeton.edu/

		scenario A			scenario B		
lexicon	size	macro-p	macro-r	macro-f1	macro-p	macro-r	macro-f1
BG-s-syn	2847	0.867	0.554	0.675	0.750	0.563	0.642
BG-s-sim	4752	0.848 (-0.019)	0.650 (+0.096)	0.736 (+0.060)	0.761 (+0.012)	0.635 (+0.072)	0.691 (+0.049)
BG-s-also	3236	0.874 (+0.007)	0.625 (+0.071)	0.726 (+0.050)	0.766 (+0.016)	0.608 (+0.045)	0.678 (+0.036)
BG-s-deriv	4184	0.875 (+0.008)	0.618 (+0.064)	0.723 (+0.048)	0.747 (-0.002)	0.612 (+0.049)	0.671 (+0.029)
BG-s-hyp	16255	0.754 (-0.113)	0.624 (+0.070)	0.681 (+0.006)	0.608 (-0.142)	0.614 (+0.051)	0.602 (-0.040)
BG-s-all\hyp	7517	0.854 (-0.013)	0.707 (+0.153)	0.773 (+0.098)	0.780 (+0.030)	0.660 (+0.097)	0.715 (+0.072)
BG-s+d+t-syn	5567	0.906	0.788	0.843	0.805	0.730	0.765
BG-s+d+t-sim	7223	0.904 (-0.002)	0.794 (+0.006)	0.845 (+0.003)	0.812 (+0.007)	0.753 (+0.024)	0.781 (+0.016)
BG-s+d+t-also	5870	**0.912** (+0.006)	0.796 (+0.007)	0.849 (+0.006)	0.810 (+0.005)	0.735 (+0.005)	0.770 (+0.005)
BG-s+d+t-deriv	6843	0.907 (+0.002)	0.805 (+0.016)	0.853 (+0.010)	0.801 (-0.004)	0.746 (+0.016)	0.772 (+0.007)
BG-s+d+t-hyp	18898	0.839 (-0.066)	0.816 (+0.028)	0.827 (-0.015)	0.704 (-0.101)	0.730 (+0.000)	0.715 (-0.050)
BG-s+d+t-all\hyp	9863	0.907 (+0.001)	**0.822** (+0.033)	**0.862** (+0.019)	**0.824** (+0.020)	**0.763** (+0.034)	**0.792** (+0.027)

Table 9.14.: Digital camera corpus: Effectiveness of different sets of WordNet relations for the label propagation approaches.

figurations without adding domain or target-specific expressions. Results show that the similar-to relation causes the greatest increase with around +5-6 percentage points, followed by the see-also (+3-5 percentage points) and the derivationally-similar relation (+3-5 percentage points). Including the hyponymy relation consistently leads to worse results (f-measure). It expands the lexicon to a size that is around five times greater than the reference. Naturally, this causes a higher recall, but on the other hand too much noise is introduced so that precision is lowered disproportionately (scenario B). As the hyponymy relation is only defined for nouns, the vast expansion stems exclusively from additional nouns. However, we also know that most sentiment expressions are adjectives and only less than 10% are nouns (cf., Section 6.2.2). In consequence, we may not expect a major improvement in recall by using this relation. In fact, the increase in recall is not greater than for the similar-to relation.

In case the configuration is enriched with domain and target-specific entries, improvements due to including the additional relations are relatively small (< 1 percentage point). We receive the best results with including all relations in combination. The f-measure improves around 2.7 percentage points for the camera corpus and 1.3 percentage points for the hotel corpus. In summary, we can conclude that in our context it is reasonable to additionally consider the listed relations as part of the label propagation algorithm[20]. We also tested the additional relations for the Rao method, but no major improvements were observed. Results were always 15-25 percentage points lower than with the BG method.

9.5. Summary and Conclusions

In this chapter our goal was analyze the use of sentiment lexicons for the task of fine-grained polarity detection in customer reviews. We set focus on approaches to automatically generate such lexical resources. In Section 9.1, we first provided a general overview of existing approaches. For this purpose, we developed a hierarchical classification framework that allowed us to categorize the various approaches along different dimensions. As part of this analysis, we pointed out that the vast majority of currently existing approaches is based on general purpose sentiment lexicons. Such lexicons typically neither fit the general application domain (e.g., customer reviews), nor the concrete application domain (e.g., hotel or digital camera reviews). Besides this domain dependence, we further highlighted the fact that for some expressions the sentiment status is even dependent on the related product aspect (sentiment target). We thus derived a strong need for incorporating domain and target-specific knowledge into existing general purpose sentiment lexicons. Our primary goal was to devise a highly

[20] Take note that the results presented in the previous subsections are already based on this best performing configuration.

accurate, fully automatic method that easily scales out to different product domains. To this end, we proposed a method that leverages the short, semi-structured pros and cons text that are attached to most customer reviews. Section 9.3 provided a detailed description of our proposed approach.

We evaluated our approach in comparison to four baseline approaches. For this comparison we chose two state-of-the-art, thesaurus-based, automatic approaches [40, 312] as well as two prominent, handcrafted sentiment lexicons (the MPQA Subjectivity Lexicon and the lexicon by Liu et al.). The two automatic methods follow a semi-supervised approach. They apply a label propagation algorithm to a semantic graph that is derived from the WordNet knowledge base and a set of handcrafted, polar seed words. Section 9.2 provided a brief overview of both approaches. Besides evaluating our own approach, we were also interested in comparing the effectiveness of these two approaches. We further proposed and evaluated some adaptations to the original label propagation approaches.

In the following, we summarize the main results and conclusions of this chapter. Our main findings were:

- Our result of comparing the two label propagation approaches for sentiment lexicon construction clearly indicated that the method proposed by Blair-Goldensohn et al. [40] outperforms the method by Rao and Ravichandran [312]. For both evaluation datasets and for all evaluation scenarios we observed differences in f-measure of 15-25 percentage points. The major problem with the Rao method was a low recall caused by the fact that it was unable to expand the seed word set significantly. The resulting lexicon was around ten times smaller than the lexicon obtained with the BG method.

- Intrinsic evaluation of the label propagation methods revealed that the greatest loss in precision stems from the erroneous inclusion of terms with neutral polarity. Major reason for this effect was missing word sense disambiguation and disrespect of the node degree within the semantic graph. Further attempts for improvement of the approach may use these observations as starting points.

- Comparing the two handcrafted lexicons, we find that the Liu et al. lexicon shows significantly better results than the (adapted) MPQA lexicon (around 20-30 percentage points difference in f-measure). In comparison to the automatic BG method, results are slightly better for the camera dataset, but slightly worse for the hotel corpus. Whereas the Liu et al. lexicon generally is more precise (it was handcrafted), the recall is higher with the BG method.

- The short pros and cons texts attached to customer reviews mostly consist of incomplete, grammatically incorrect sentences. To extract information from this kind of data, we find that high-precision patterns defined over part-of-speech tag sequences show a good performance. With this approach we were able to extract relevant information (target/sentiment tuples) from every second pros/cons text. Out of 500,000 texts we generated a set of 250,000 tuples.

- The recall of the individual extraction patterns is strongly dependent on the actual format of the pros/cons texts. In general we found that the vast majority of extractions can be obtained with very few, simple patterns. For the digital camera dataset we observed that two patterns yielded nearly 95% of all extractions. Regarding the results obtained with our set of patterns, we concluded that it sufficiently generalizes from the concrete datasets and is applicable to various domains.

- Extending the baseline lexicons with the extracted domain and target-specific sentiment expressions led to major improvements, mostly in recall. Both extensions, either only the domain-specific entries or only the target-specific part yielded significant improvements. We obtained the best results by combining the extracted domain and target-specific expressions and adding them to an existing general purpose lexicon. For this configuration, we observed increases in f-measure of up to 14 percentage points compared to the baseline approaches without extension.

- We experimented with incorporating the extracted domain-specific expressions as additional seed words for the label propagation algorithm. However, we found that this approach does not lead to better results. Considering the doubled lexicon size generated by this method, we concluded that simply adding the extracted terms as a lexicon extension is more reasonable.
- Intrinsic evaluation of the extracted domain and target-specific expressions indicated that our extraction heuristics are highly accurate. For the domain-specific lexicons we observed precision values of over 80% and for the target-specific lexicons precision was even at around 95%. Most errors stem from the inclusion of non-polar entries. False polarity classification occurred very rarely.
- In Section 9.3.5 we proposed and described an idea to use label propagation for expanding target-specific sentiment lexicons. We experimented with this approach, but even the best results obtained with the method did not lead to improvements. More closely inspecting the really target-specific sentiment expressions in our corpora revealed that most of them describe (physically) quantifiable properties (e.g. "size", "length", "weight"). As the lexical variability to describe such properties is rather low, further expanding the lexicons was useless.
- As part of our experiments with the label propagation approach, we also analyzed the influence of utilizing different semantic and lexical relations available in WordNet. We found that extending the original approach by also incorporating the "similar-to", the "see-also", and the "derivated-from" relations yielded the best results. Incorporating the "hyponymy" relation led to worse results.
- In general, we conclude that domain adaptation of sentiment lexicons is important to improve the accuracy of a sentiment analysis system. Also recognizing the target-specific polarity of sentiment expressions further promises to improve results. Our proposed method of leveraging pros and cons texts supports both tasks in a fully automatic and thus "cheap" way. The results of our experiments show that the approach is highly accurate and truly improves results compared to state-of-the art baseline methods.

Knowledge is true opinion.

Plato

10. Polarity Classification at the Sentence Level

A core task in sentiment analysis and customer review mining is to detect whether a text expresses positive or negative sentiments towards some topic or entity. In the previous chapter, we have pointed out that approaches to this sentiment polarity detection task can be broadly categorized into methods that primarily rely on lexical resources (i.e., sentiment lexicons) and methods that primarily make use of machine learning techniques. The main advantage of using sentiment lexicons is that there is no need for annotated text corpora. Generating such training corpora for supervised machine learning algorithms is generally costly, time-consuming, and needs to be done for each new target domain. On the other hand, creating a sentiment lexicon and integrating this knowledge base into some sort of rule-based system is a one-time effort. Furthermore, such systems have proven to obtain quite good results in customer review mining [45, 102, 204, 205, 214, 215, 241, 258, 453, 468] and other sentiment analysis tasks [24, 100, 146, 344, 354, 374, 380]. However, we also know (see the last chapter) that, in order to achieve the best results, sentiment lexicons need to be adapted to the target domain too. In general, lexicon-based approaches are limited by the dictionary size and by the coverage or complexity of the rules developed for polarity detection. Machine learning often allows to learn more complex "rules" and to exploit rather hidden correlations which may exist in the actual data. For example, phenomenons such as polar facts (factual information implying positive or negative appraisal, see Section 4.2) are very difficult or impossible to detect by means of sentiment lexicons.

In this chapter, we study the applicability of supervised machine learning methods for sentiment polarity detection in customer reviews. Regarding the document level (that is, to determine the general tone of a text), the use of machine learning approaches for this task has been extensively studied in the past [9, 92, 96, 135, 252, 269, 279, 295, 297]. In this particular context, the task is most commonly denoted as *sentiment classification* or *polarity classification*. We are interested in a more fine-grained analysis. Our goal is to determine the polarity of individual sentences of a document. For instance, we want to classify a sentence such as "The check-in process took ages." as negative, whereas a sentence such as "The check-in process went smooth and the reception staff was most welcoming." should be classified as positive. Corpus analysis in Chapter 6 has shown that reviewers typically take a nuanced view on the reviewed entity: Even generally negative reviews (one or two stars as rating) contain a significant amount of positive utterances (in average around 15% of all polar sentences). And vice versa, the same observation is made for overall positive reviews (four or five stars), where around 15% of all polar sentences are negatively connoted. Analyzing polarity at the sentence level thus allows to provide more detailed insights into the true sentiments expressed by a reviewer. Further, for an aspect-oriented review mining system such finer-grained analysis renders an inevitable requirement.

Sentence level polarity classification is not a new research problem, but there has been much less work in this direction. One of the main reasons is presumably the effort involved when creating annotated corpora for this task. We will make use of our discourse-oriented corpora (both comprising more than 3,000 annotated sentences) to examine the problem more closely. A main contribution of this chapter is to study the applicability of using weakly labeled data for sentence level polarity detection. In particular, we again propose to exploit the information contained in pros and cons summaries of customer reviews. We describe our approach to automatically creating huge training corpora from this data. We further experiment with different strategies of incorporating this weakly labeled data into a supervised polarity classification scheme. One specific challenge is that pros and cons primarily provide labeled samples of evaluative, polar language, but not of factual, objective

language. Typically a polarity classification task involves a subjectivity detection step. However, to build a classifier that can distinguish between factual/objective and polar sentences, we also need samples for the objective class. To overcome this problem, we examine the applicability of *one-class classification* algorithms. With such an approach it is possible to learn a concept in the absence of counter examples. Our results show that incorporating weakly labeled data from pros/cons summaries is generally helpful for sentence level polarity classification. For binary polarity classification (positive vs. negative) the benefits are twofold: We can perfectly substitute the (expensive) manually labeled data with the (cheap) weakly labeled data and even achieve better classification performance. On the other hand, our approach to use one-class classification to create subjectivity classifiers from weakly labeled data is not successful. For this task, we achieve better results with an unsupervised, lexicon-based approach.

The remainder of this chapter is organized as follows: In Section 10.1, we provide an overview of supervised sentiment polarity detection. We categorize different subtasks, describe typical approaches, and point out some specific challenges. In the context of machine learning approaches, feature engineering plays an important role. In Section 10.2, we concentrate on this topic and provide a critical review of the relevant literature. The subsequent two sections present our main contributions in this chapter. Section 10.3 describes our approach to gathering weakly labeled data for sentence level polarity classification and briefly introduces the concept of one-class classification. Section 10.4 covers the experiments we have conducted and reports our results. We summarize our main findings and conclusions in Section 10.5.

10.1. Overview

10.1.1. Sentiment Polarity Classification Tasks

Sentiment polarity classification is presumably the most well studied subtask in sentiment analysis. Nonetheless, it is difficult to provide a single, clear-cut definition. Problem settings which are subsumed under the umbrella term "polarity classification" are quite heterogeneous. Depending on the application domain and even more on the examined dataset, researchers and practitioners set out varying goals. We have already learned that one distinction is the granularity of analysis — for example, document versus sentence level classification. In fact, we may categorize the different polarity classification tasks along this dimension. More precisely, we distinguish granularity of analysis in terms of the examined unit of text (document, sentence, or sub-sentence) and the type of the "dependent variable" (the possible outcomes of the classification). Figure 10.1 illustrates both dimensions. With regard to the type of the dependent variable, we differentiate between binary/dichotomous (i.e., "positive" vs. "negative"), ternary (i.e., "positive" vs. "negative" vs. "objective"), and ordinal (i.e., classification by means of a rating scale or by sentiment strength) classification tasks.

Document Level

Binary Polarity Classification When exploring the relevant literature, it is most apparent that the vast majority of work considers the task of binary, document level polarity classification. Here, the goal is to determine whether the overall tone of an entire document is either positive (recommendation) or negative (disapproval). Early and most influencing works such as the ones by Pang et al. [297] or Dave et al. [96] set out this direction. Research in this line was and is primarily fueled by the fact that sample data is conveniently available for many application scenarios via indirect crowdsourcing methods. For instance, in the customer review mining scenario, sample data for binary, document level polarity classification is easily obtained by exploiting the overall rating, which is explicitly pro-

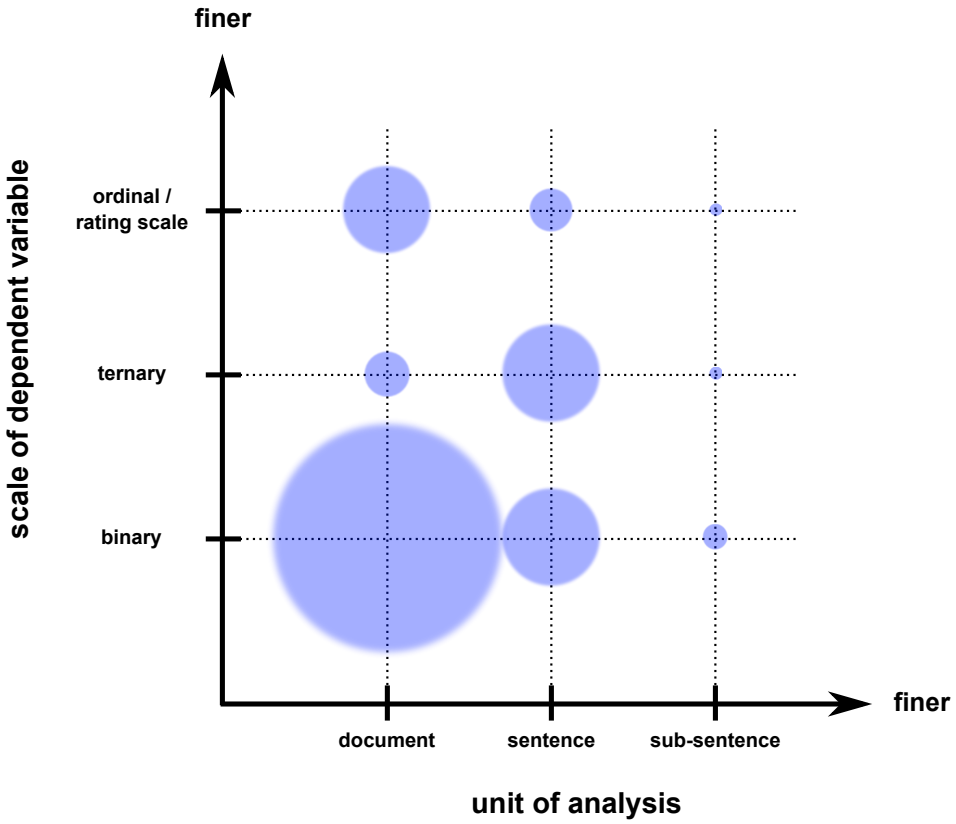


Figure 10.1.: A framework for the categorization of sentiment polarity classification tasks: The size of the circles at each data point illustrates the relative amount of publications which discuss a specific task. The numbers merely represent our subjective impression after studying the related literature.

vided by a reviewer (e.g., the "star rating" on Amazon.com). Pang et al. [295, 297] derive a corpus[1] by extracting high and low rated movie reviews from the Internet Movie Database[2] (IMDb). The corpus has been used in more than hundred other experiments[3,4] since then. Similar corpora have been crawled by Cui et al. [92] (320k product reviews), Dave et al. [96] (6k product reviews), Gamon [135] (40k customer feedback reports), or Thomas et al. [381] (3k pages of U.S. floor debate transcripts), to name a few.

However, when simply crawling data from the Web, assuming one part to be positive samples, the other part negative samples, and then feeding these to an arbitrary supervised machine learning algorithm to learn a binary classification model, at least two legitimate questions emerge:

- Purpose: Nearly all customer reviews on the Web already provide the overall rating of the reviewer. Then, why do we need to learn a classification model to predict the rating? Many studies tacitly ignore that the actual goal is to apply the learned model in a different domain, where explicit ratings are not available. For example, we may learn a polarity classifier from product reviews and then try to predict the general tone of blog entries (discussing similar products). Consequently, this involves that we reason about the domain adaptability of the trained classifiers. However, in most studies this question for adaptability remains unanswered (often

[1] http://www.cs.cornell.edu/people/pabo/movie-review-data/
[2] http://www.imdb.com/reviews/index.html
[3] http://www.cs.cornell.edu/people/pabo/movie-review-data/otherexperiments.html
[4] Only a subset of the listed works make use of supervised machine learning techniques.

because appropriate, manually labeled corpora are lacking). Some works which *do* examine domain adaptation[5] are for instance [14, 20, 44, 237, 260, 293, 315].

- Findings: If, for example, domain adaptation is not considered, then what do we actually learn from such a study? In many cases the simple, but in our context unsatisfactory answer is: Yes, machine learning works — we are able to separate two classes of documents. Taking a less polemic, more differentiated perspective, we identify mainly two points: First, sentiment polarity classification of documents seems to be more complicated[6] than topic classification; the reported accuracies are typically lower compared to standard text categorization tasks. We summarize the main challenges of the polarity classification task in Section 10.1.2. Second, a great share of studies experiments with "new", often linguistically inspired feature types. Findings then refer to the classification performance with different feature sets. We take up this discussion in Section 10.2.

Ternary Polarity Classification and Subjectivity Detection Besides ignoring the question of domain adaptability, another difficulty is most often ignored or set aside: The binary polarity classification task tacitly assumes that a document is either predominantly positive or negative. Whereas this is (mostly) true for the domain of customer reviews, this is not the case for other genres (e.g., newswire text, blog, or microblog postings). Documents may not be subjective at all or may contain rather mixed polarities without clearly revealing an either positive or negative viewpoint. For instance, if we train a binary classifier on customer review data to predict the polarity of blog entries, we are unable to handle objective entries. To overcome the issue in this context, mainly two strategies may be followed. The first strategy is to sort out documents which do not match the genre/domain of the training set. For instance, we may try to separate blog entries (or arbitrary documents) into "review-like" and "non-review-like". Postulating that a review either recommends or disapproves, we can use the learned binary polarity prediction model. The strategy is for example examined by Barbosa et al. [28] or van der Meer and Frasincar [396]. The second, more general approach is to extend the binary classification task to a ternary classification task:

Besides the two outcomes "positive" and "negative", a third category "neutral" or "objective" is introduced. Be aware that the term "neutral" is not used consistently in the literature[7]. The neutral category may either describe objective/factual documents or texts which express neutral polarity (i.e., an attitude in between "positive" and "negative"). When following the former interpretation, ternary polarity detection actually involves a **subjectivity detection** task that separates subjective from objective documents. In contrast, the latter interpretation is rather a special instance of the ordinal (or rating scale) classification task, which we discuss later. The objective category may be either covered directly, by learning a multi-class classification model, or by using a cascaded approach. The cascaded approach involves to train a binary subjectivity classifier in addition to a binary polarity classifier. Documents which have been identified as subjective are fed to the polarity classifier, other documents are predicted as objective. Fig. 10.2 illustrates this approach. The cascaded approach re-

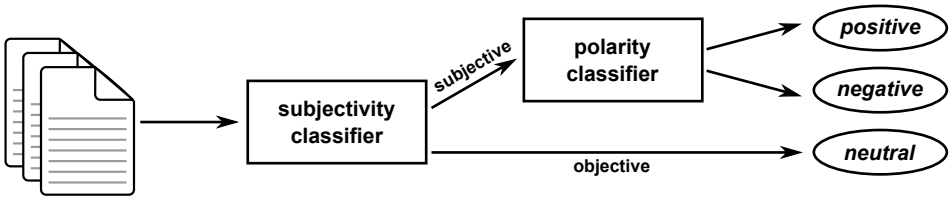


Figure 10.2.: The cascaded approach to ternary polarity classification.

[5] The cited works consider adaptation with respect to genre, topic, and/or time.
[6] see also Pang and Lee [294, chap. 3])
[7] see also Koppel and Schler [219]

gards subjectivity detection as a separate task. The classifiers may be trained on a different sample set than the polarity classifiers and also the applied machine learning algorithms may differ. Most influencing works which examine supervised approaches to subjectivity detection are amongst others by Yu and Hatzivassiloglou [455], Wiebe et al. [416], Pang and Lee [295], or Ng et al. [279]. Subjectivity detection was also part of the 2006-2008 TREC Blog tracks [242, 243, 289]. A direct approach to ternary polarity detection is for instance taken by Koppel and Schler [219] or Das and Chen [95]. We will experiment with both approaches in Section 10.4.4.

Rating Inference We have seen that the convenient availability of training data often plays a major role in the concrete definition of sentiment polarity classification tasks. Following the observation that many customer review sites encourage the review authors to classify their overall impression on a rating scale, a more fine-grained polarity classification task has been postulated: The goal is to further differentiate between various degrees of polarity. That is, the dependent variable in this task is defined on an ordinal scale. The actual task can thus be considered as an instance of an *ordinal regression* problem [256]. Most works consider a five point scale as it is used by many of the popular review/e-commerce sites such as Amazon.com, Tripadvisor.com, or Buzzillions.com. This "rating inference problem" [296] is generally assumed to be more difficult than binary polarity classification. For instance, accurate handling of sentiment shifters (e.g., intensifiers or downtoners) may become more important [308]. Most relevant works that examine the rating inference task are [21, 38, 148, 154, 284, 296, 345, 348].

Sentence Level

Sentence level polarity classification allows for more fine-grained analysis. For example, this model supports tasks such as aspect-based review mining or sentiment aware question answering. However, much less work has been published in this direction. We already pointed out that the main reason is presumably the lack of appropriate datasets. In comparison to document level classification, training data is often not conveniently available. Further, polarity classification at the sentence level is inherently more difficult due to the relative sparsity of information. Sentences simply contain less words, thus, resulting feature vectors are typically quite sparse. In consequence, feature engineering may play a more important role — for example, Wiegand and Klakow [422] report improved results by incorporating various knowledge-based and linguistic features.

As with document level classification, we can subdivide different studies according to the granularity of the dependent variable. Our impression is that the number of works considering binary or ternary classification is roughly equal. The binary model is for instance examined by [16, 111, 181, 258, 273, 422]. The ternary model (i.e., including an objective class) is studied by [40, 129, 136, 210, 307, 375, 423, 455]. A cascaded approach to combined polarity and subjectivity detection is for example taken by Qu et al. [307] or Yu and Hatzivassiloglou [455]. Most others (e.g., Blair-Goldensohn et al. [40] or Kim and Hovy [210]) model the objective class directly by learning a multi-class classifier. To the best of our knowledge, no work exists that tries to classify sentences according to a polarity rating scale.

We found quite a large amount of studies where evaluation is conducted on corpora that were automatically extracted from the Web. The validity of such an evaluation method is highly questionable. Neither the true labels of individual samples, nor the error rate of the extraction heuristic are generally known. Most often, the corpus introduced by Pang and Lee [296] is used as a reference. This movie review corpus[8] has been constructed by exploiting the review snippets (similar to pros/cons) available on the Rottentomatoes.com[9] website. Only a few manually labeled sentence level corpora

[8] http://www.cs.cornell.edu/people/pabo/movie-review-data/rt-polaritydata.README.1.0.txt
[9] http://www.rottentomatoes.com

have been presented in the literature. For instance, Ganu et al. [138] construct a dataset of 3,400 sentences (taken from restaurant reviews), which are categorized into four classes, including a "mixed" category. McDonald et al. [257] crawl product reviews and manually label a dataset of roughly 4,000 sentences according to the ternary model. Also Gamon et al. [136] evaluate on a manually labeled dataset (approx. 3,000 sentences, ternary model). Täckström and McDonald [375] propose a method to learn a sentence level classifier by utilizing document level labels. Evaluation is based on a 4,000 sentence, manually labeled corpus.

Sub-sentence Level

Even sentence level analysis is sometimes too coarse-grained. Different clauses of a sentence may express opposing polarity. For example, conjuncts such as "but", "although", or "however" often introduce a contrasting clause (e.g., when an author makes a comparison). Opposing polarity may also be expressed towards different sentiment targets. Capturing such fine-grained information, as is for example allowed by our expression level model (cf., Section 4.3), thus requires analysis at the sub-sentence level. Most approaches to clause or phrase level polarity detection are rule-based, using a sentiment lexicon (e.g., [102, 452]). Less works consider supervised approaches. Again, this is mainly due to the lack of easily available training corpora. Annotating at the sub-sentence level involves even more effort than manually creating a sentence level corpus. We have seen in Chapter 5 that only a few appropriate corpora exist — most prominently, the MPQA corpus [413]. The majority of works on supervised polarity detection at the sub-sentence level is based on this corpus.

When discussing sub-sentence polarity detection, we may further differentiate between clause and phrase level analysis. Clause level polarity classification is for instance examined by Meena and Prabhakar [258] or Zirn et al. [469]. Both works try to identify polarity shifting conjuncts. Zirn et al. [469] explicitly position their work in the context of analyzing *discourse relations* (see also Section 4.2). Such relations are also incorporated by Somasundaran et al. [356], but on a more fine-grained level.

Polarity detection at the phrase level is more closely related to information extraction problems than to traditional text classification. Thus, works considering a supervised approach, often propose to model the problem as a *sequence labeling* task. For instance, Breck et al. [53] propose to use conditional random fields to identify sentiment expressions. A similar approach is taken by Jakob and Gurevych [183], but here the focus is on extracting sentiment targets. Traditional classification models are used for example by Wilson et al. [438] or Choi and Cardie [76]. The former propose a cascaded approach which first classifies phrases as subjective/objective and then determines their polarity. Wilson et al. [436] examine supervised methods to classify phrases with regard to their sentiment strength.

Structured Models

In addition to explicitly focusing on a single level of analysis, some researchers propose models that capture the inherent dependencies between individual levels. For instance, knowing that the number of positive sentences in a document is roughly equal to the amount of negative sentences, may indicate that the entire document is of rather mixed polarity. Sentence level analysis may thus help when the task is to infer the overall tone of a document. A first work in this direction was presented by Pang and Lee [295]. They perform subjectivity detection at the sentence level and only use the derived subjective extracts of a document for subsequent polarity classification. A similar approach is examined by Yessenalina et al. [451], but here the sentence level is modeled as a hidden variable so that training can be performed at the document level. Also McDonald et al. [257] study the applicability of structured models for jointly classifying sentiment polarity at different levels of granularity.

10.1.2. Challenges

We may wonder whether polarity classification is really a new problem: Which properties distinguish the task, for example, from traditional text categorization? Is classification according to sentiment polarity different from classification by topic? Our short answer is: Yes, to a certain extent, and it is generally more difficult. In the following we briefly summarize the main properties which pose extra challenges when classifying by sentiment.

The primary factor is that sentiment polarity often cannot be determined without considering the context. We know that **sentiment shifters** may alter the sentiment status of individual expressions, entire sentences, or even the whole document. Citing Pang and Lee [294, chap. 3], "compared to topic, sentiment can often be expressed in a more subtle manner, making it difficult to be identified by any of a sentence or document's terms when considered in isolation." In consequence, it is assumed that simple bag-of-words representations lead to inferior results (in comparison to traditional topic classification). Linguistic analysis shows that word order and the dependency structure of individual phrases may indeed influence the polarity status of a sentence. As an example, consider the sentence "I like the camera despite the fact that any other reviewer seems to hate it." Not considering the word order of "like" and "hate" would result in estimating false sentiment polarities.

We already discussed sentiment shifters in detail in Section 4.3. Our corpus analysis has shown that, among the shifters which affect polarity, negation is the most frequent one (5-7% affected sentiment expressions). Neutralizers shift around 3-8% of the sentiment expressions in our corpus. The other measured shifters (e.g., amplifiers, downtoners, solidifiers) typically do not affect the polarity, but rather influence sentiment strength or intensity. Polanyi and Zaenen [303] name further linguistic phenomenons and shifter types, such as the use of sarcasm or irony, the expression of presuppositions, or the occurrence of certain discourse relations. Earlier we pointed out that contrasting conjuncts such as "but" or "although" may also shift polarities.

10.2. Feature Engineering

Machine learning approaches cannot directly operate on text. A document typically needs to be mapped to a numerical vector (*feature vector*), which is then interpretable by the classification algorithm. Each dimension of this vector represents a single feature that may characterize the document. Choosing an adequate mapping function (i.e., extracting meaningful features) is essential in building good classification models. For text categorization by topic, the simple bag-of-words document representation achieves very good results and, for instance, Sebastiani [338] points out that "representations more sophisticated than this do not yield significantly better effectiveness." In the last section we learned that polarity classification is generally assumed to be a more difficult task than categorization by topic. Linguistic phenomenons and subtleties (e.g., sentiment shifters) play a more important role. It is thus unclear if more sophisticated, linguistically inspired text representations are of any help. Many different feature sets have been proposed, trying to encode linguistic and other properties which are deemed beneficial for polarity classification.

In this section, we provide a critical and balanced review of the relevant literature. We concentrate on results that have been reported with regard to **document or sentence level** classification. We often find contradicting and/or inconclusive results so that it is unclear which feature sets are really helpful. Our goal is to distill and summarize the most valuable insights. That is, we try to answer the question which feature sets have been found to consistently improve results compared to a bag-of-words baseline.

When comparing reported results, we need to be aware that the optimal choice of features may depend on the actual dataset (i.e., application domain). Further, it may depend on the unit of analysis (i.e., document vs. sentence level). Nonetheless, we are trying to draw some general conclusions.

For document level classification, typically much redundant information is available. Missing or misinterpreting a few complex linguistic constructs may not harm as enough, easier to interpret information is available. At the sentence level (or even sub-sentence level) such redundant information is typically lacking so that complex linguistic features may become more important.

10.2.1. Lexical Features

We speak of lexical features when referring to properties that can be easily extracted from the surface form of a text with very shallow analysis. Most notably, these are simply the terms constituting the text.

N-grams Whereas individual terms in a bag-of-words model[10] can capture *lexical semantics*, they fail to encode *compositional semantics*[11] (i.e., in our case the contextual polarity). A simple tool for encoding compositional semantics in a bag-of-words model is to use higher order token n-grams. In a very early work, Pang et al. [297] compare unigram and bigram models[12] for document representation. Their finding is that adding bigrams to the unigram model does not improve performance (1,400 documents corpus). This result is contradicted by a set of other works: Cui et al. [92] examine the utility of n-grams up to an order of six and report that 6-grams perform best (320k documents corpus). Whereas improvements in f-measure for the positive class are marginal, performance regarding the negative class raises by approximately 7 percentage points. Their results also show that the improvement is already achieved with a bigram model. Differences between bigram and higher order models are marginal (at maximum 0.17 percentage points). As they do not report statistical significance regarding the difference between various higher order models, it must be assumed that the bigram model performs equally well compared to the other models. Also Ng et al. [279] report improved results by adding bigrams and trigrams. These results were obtained on a similar sized corpus as used by Pang et al. [297], but feature selection[13] was conducted. Further, Dave et al. [96] indicate that higher order n-grams meliorate performance (6k documents). Here, the inclusion of all lower order n-grams degraded performance, but no feature selection was conducted.

We summarize and conclude as follows: Considering higher order n-grams in a bag-of-words model is generally helpful for polarity classification and can improve results significantly. Care must be taken to circumvent problems with data sparsity. The benefits of higher order n-grams come into play with "sufficiently" large training corpora. When using smaller corpora, conducting a feature selection step is a critical factor. Our own experiments support this view.

Bag of Opinions Qu et al. [308] propose a "bag-of-opinions" model to overcome the "sparsity bottleneck" with higher order n-gram models and small corpora. In their model an *opinion* is a triplet that consists of a sentiment expression, a set of associated modifiers (amplifier or downtoner), and a set of related negation words. Sentiment expressions and negation words are identified by means of a lexicon that also provides prior polarity scores. The contextual polarity score of an opinion triplet is then learned by *ridge regression* on a corpus with polarity labeled documents. The opinion triplets and the associated scores serve as features in their "bag-of-opinions" model. Their results show that a bigram model performs better than an unigram model. Both bag-of-words models are outperformed by their "bag-of-opinions" model. However, taking a closer look at the experimental setup, we believe that the comparison of both models is unfair and results are less insightful. Whereas the "bag-of-opinions"

[10] In the following we tacitly neglect the distinction between set-of-words (presence) and bag-of-words models (frequency) and only use the latter term. Further, sometimes "bag-of-words" actually refers to "bag-of-n-grams". We do not make such a distinction as it is typically implied by the context.

[11] The concepts of lexical and compositional semantics are for instance discussed by Manning and Schütze [249, chap. 3.3].

[12] In the following, if not otherwise stated, we assume that an n-gram model also includes all lower order k-grams with $1 \leq k < n$ (e.g., a trigram model also includes all bigrams and unigrams).

[13] The top k features according to the weighted log-likelihood ratio were chosen.

model is fueled with expert knowledge from a lexicon (known sentiment expressions with prior polarities), the bag-of-words model does not have this information. Any other representation which includes features obtained from lookups in a sentiment lexicon (e.g., by a simple voting heuristic, see later sections) may perform equally well. Unfortunately this comparison is lacking. It is thus unclear whether the improvement stems from their model or simply from including knowledge-based features.

Term Substitution Dave et al. [96] experiment with substituting individual terms by a corresponding "type". The intuition is to learn a more generalized classification model. Experiments with masking all numbers with a single "NUMBER" symbol did not have significant effects. Masking low frequency words with a "UNIQUE" symbol degraded performance.

10.2.2. Knowledge-based Features

We speak of knowledge-based features when referring to features that are extracted in consideration of existent, prior knowledge. For example, prior knowledge may be encoded in an annotated dictionary (e.g., a sentiment lexicon) and features are generated by dictionary lookups (e.g., looking up the prior polarity value).

Sentiment Lexicons Many works examine the value of sentiment lexicon information for supervised polarity classification. We can summarize that encoding such prior knowledge consistently and significantly improves results:

Ng et al. [279] handcraft a sentiment lexicon with 3,599 positively labeled and 3,204 negatively labeled adjectives. Knowledge-based features are created by substituting each sentiment expression in a bigram with the polarity label (e.g., "nice movie" or "great movie" both become "*positive* movie"). Their experiments with document level polarity classification show that the accuracy improves by over 3 percentage points on two different datasets (both movie reviews). Nakagawa et al. [273] consider binary, sentence level polarity classification for different Japanese and English corpora. Their sentiment lexicons contain approximately 15,000 (Japanese) and 6,000 (English) sentiment expressions. In addition, they handcraft lexicons of "reversal" words that flip prior polarities (e.g., negations such as "not" or "never"). Features are generated by a simple majority voting heuristic, optionally incorporating the reversal information. Experiments on 8 different corpora (4 Japanese, 4 English) show that adding the sentiment lexicon features to a bag-of-words model consistently improves results (compared to "bag-of-words" alone). The minimum reported increase in accuracy is 0.4 percentage points, the maximum is 6.5 percentage points. Additionally considering the reversal words, only marginally increases accuracy (unknown statistical significance). Also Wiegand and Klakow [422] examine the influence of prior polarity features for binary, sentence level classification. Utilizing the subjectivity lexicon compiled by Wilson et al. [438], they also consider the strength of sentiment expressions. Features are generated by counting the number of occurrences of (weak/strong) positive/negative/neutral expressions in a sentence. Adding these prior polarity features leads to an improvement of 6.8 percentage points over a bag-of-words baseline.

Baccianella et al. [21] incorporate prior polarity information by substituting sentiment expressions by their label (similar to Ng et al. [279]). Prior polarity values are obtained from the *Harvard General Inquirer* lexicon [360]. Their experiments in an ordinal regression setting are unfortunately not conclusive. Depending on the evaluation method, the inclusion of prior polarity features either decreases or increases the mean squared error. Further, a test for statistical significance is not conducted. Blair-Goldensohn et al. [40] use a scored sentiment lexicon to compute polarity scores for a sentence. These scores (instead of a single positive/negative label) are used as features for sentence level classification. Unfortunately, they do not compare polarity features with a bag-of-words baseline. During our literature review we did not find any work that reports negative results regarding the inclusion of prior

polarity information. We thus repeat our earlier conclusion that features obtained from sentiment lexicons are very helpful for polarity classification.

WordNet Dave et al. [96] and Wiegand and Klakow [422] experiment with adding features obtained from the WordNet database [263]. The basic intuition in using WordNet is to provide a classifier with more generalized information (e.g., in the form of synonyms or hypernyms). However, the results are contradicting and inconclusive:

Dave et al. [96] report that incorporating synonyms as additional features does not improve accuracy. Detecting the correct synset in WordNet requires word sense disambiguation and lacking this information may "produce more noise than signal". "Attempts to develop a custom thesaurus from word collocations [...] were also unsuccessful" [96].

Wiegand and Klakow [422] include all direct hypernyms of a word as additional features. They (heuristically) avoid word sense disambiguation by simply choosing the first synset in WordNet. In their experiments, using hypernyms improves the accuracy for a binary, sentence level classification task by 1.1 percentage points. Despite the fact that the improvement is relatively low, no test for statistical significance is conducted. Using hypernyms as additional features was originally inspired by Scott and Matwin [337] who examine this feature type in the context of traditional text categorization. Also here, results are inconclusive. The effectiveness much depends on the dataset. We summarize and conclude that the use of WordNet derived features is unlikely to significantly improve performance for polarity classification tasks.

Review Label When performing sentence level polarity classification, the overall, document level polarity is a good indicator. Blair-Goldensohn et al. [40] propose to include the user-provided rating of a review as an additional feature. Their experiments show that, at least for the positive class, the f-measure increases significantly over a baseline with prior polarity features. Obviously, such a feature is only relevant for sentence level classification (at the document level the feature is equal to the class label). Further, user-provided labels are not available in each application domain.

Substitute Entities Dave et al. [96] also experiment with knowledge-based substitution. To provide more general features, they substitute occurrences of product names with a single symbol "productname" and aspects with a symbol "producttypeword" (aspects are determined heuristically). Their experiments show that both substitution methods do not have significant effects.

10.2.3. Linguistic Features

Linguistic features also incorporate a form of prior knowledge. In this case it is the knowledge that certain linguistic phenomenons are relevant for determining the sentiment polarity. Extracting such features often involves costly (in terms of processing time) linguistic preprocessing steps such as natural language parsing.

Stemming and Lemmatization The most simple form of including linguistic information is to reduce terms to their stem or lemma. Again, the intuition is to provide a machine learning algorithm with more generalized features. We may also regard stemming or lemmatization as a linguistically inspired form of feature reduction. Results are inconclusive:

Dave et al. [96] obtain mixed results with using the *Porter stemmer*[14]. They hypothesize that negative results stem from possible overgeneralization. They point out that their "corpus of reviews is highly sensitive to minor details of language [...]. For example, negative reviews tend to occur more frequently in the past tense, since the reviewer might have returned the product." Matsumoto et al. [252]

[14] http://tartarus.org/~martin/PorterStemmer/

report similarly mixed results regarding lemmatization. On the other hand, Wiegand and Klakow [422] report an improvement by using the WordNet lemmatizer. However, it is unknown whether these results are statistically significant. Considering that potential improvement is relatively low, we conclude that stemming and lemmatization is not necessary during feature extraction. This conclusion is further supported by our own experiments with sentence level polarity classification (see Section 10.4.2).

Part-of-speech for Shallow Word Sense Disambiguation Suffixing terms with their part-of-speech tag is often used as a shallow form of word sense disambiguation. For example, POS tag suffixes distinguish the adverb "like_RB" (no prior polarity) from the verb "like_VB" (positive prior polarity). However, at least in the case of document level sentiment polarity detection, incorporating part-of-speech information does not seem to be beneficial. We are not aware of any work that reports positive results. Instead, for example Pang et al. [297], Na et al. [272], or Go et al. [145] report either negative results or find no effects. For sentence level classification, Wiegand and Klakow [422] or Agarwal et al. [2] propose to aggregate prior polarity values grouped by part-of-speech. But no explicit experiments are conducted which evaluate the potential benefits of this grouping. We conclude that using part-of-speech tags for shallow word sense disambiguation does not help for polarity classification. We hypothesize that redundant information (for document level classification) compensates for potential mistakes with word sense disambiguation. Further, higher order n-grams may also encode hints for disambiguation (e.g., consider the bigrams "i like" and "is like"). Our experiments (Section 10.4.2) support this view.

Syntax The main argument for considering syntax features is that compositional semantics cannot be captured with simple n-gram models. Citing for example Ng et al. [279]: "While bigrams and trigrams are good at capturing local dependencies, dependency relations can be used to capture non-local dependencies among the constituents of a sentence." The intuition is that complex linguistic constructs that are used to express sentiment can be better captured when considering the syntax of a sentence (e.g., based on *dependency grammar* or *constituency grammar* models). Many works propose different methods for extracting features from syntax, but it is often questionable whether such a representation really helps. We find positive and negative results. However, taking a closer look, most positive results are inconclusive for different reasons. In the following we summarize our findings:

One of the earliest works that examine the use of more complex linguistic analysis for sentiment classification is by Gamon [135]. He compares a simple model, covering "surface features" only (trigram bag-of-words + POS tags + lemmatization), with a set of features extracted from the output of a natural language parser (including constituent structures and form features, such as tense). The linguistic features are very abstract and are not explicitly modeled to cover phenomenons with regard to expressing sentiment. Experiments with binary polarity classification are conducted on a 40,000 documents customer feedback corpus. Best results are obtained when using the abstract linguistic features in addition to the surface features. But the influence is minimal. Comparing the optimal results obtained for both settings (feature reduction to 2,000 features), the difference is only 0.2 percentage points in accuracy in two different experiments. Whereas Gamon's initial statement that "the addition of deep linguistic analysis features [...] contributes consistently to classification accuracy [...]" is often taken to argue in favor of syntax features, his own assessment, namely that "the improvement in practice may be too small to warrant the overhead of linguistic analysis", is mostly neglected.

Another work that reports positive results is for instance by Arora et al. [16]. To capture linguistic knowledge, they propose to extract relevant subgraphs from a dependency parse of a sentence. Subgraphs are augmented with prior polarity information from a sentiment lexicon. The resulting set of subgraphs is reduced by frequent subgraph mining algorithms. As simple n-gram features are highly correlated to the subgraph features, a method is proposed that only adds complex features if these

are more discriminative than simpler forms. Their results (binary, sentence level classification) show a small improvement of 1.3 percentage compared to an unigram baseline. Unfortunately, they do not compare their model with a higher order n-gram model (which would have been the real competitor). In consequence, we believe that the results are of no value regarding the claimed utility of complex linguistic features. We also observe (not commented on in the paper) that after feature selection, only 30 subgraph features are added to the over 8,000 unigram features. This further indicates that the complex linguistic features are only of marginal value.

Joshi and Penstein-Rosé [197] propose to use "lexicalized dependency relation features". The dependency graph representation of a sentence is transformed to a set of triplets, each covering the relation type, the head, and the modifier of a single dependency relation. To generalize the triplets, additional features are generated by masking either the head or modifier with the corresponding part-of-speech tag. Results show that their approach outperforms an unigram baseline with statistical significance. However, a simple unigram+bigram+POS model comes very close and they do not report whether this difference is significant. In consequence, their conclusion, stating that generalized lexicalized dependency relations are helpful for polarity classification, is not justifiable.

Also Pak and Paroubek [292], inspired by the observation that the "n-gram model has problems with capturing long dependencies", examine the utility of dependency relations. Again, questionable positive results are reported. Similar to Arora et al. [16], their proposal is to construct subgraphs from a dependency parse. Based on relation types (e.g., "negation" or "copula") some nodes are combined and based on part-of-speech tags some nodes are masked with a wild card symbol. Using the proposed document representation, they report an improvement of 2.4 percentage points. Unfortunately, several flaws in the experimental setup make the real value of their model disputable. First, no comparison to higher order n-gram models in conjunction with feature selection is made. Second, their optimal result is obtained by using a weighting scheme (TF/IDF[15]) that is not used for the baseline. In consequence, the improvement may as well stem from the differing weighting schemes. In fact, TF/IDF may be regarded as a form of feature selection[16], rendering comparison to the baseline unfair.

Nakagawa et al. [273] examine sentence level, binary polarity classification. They point out that the polarity of an entire sentence is defined by the polarities of sub-sentence structures. As polarities of sub-sentence structures are not known during training, they are modeled as hidden variables within a dependency tree based probabilistic framework (a conditional random field with hidden variables). Experiments are conducted on Japanese and English language corpora. For English, reported results are mixed. For two out of four corpora no effects are measured. On the other hand, for the Japanese corpora their model consistently shows an improvement of 1.7 to 2.4 percentage points. It is however not directly clear whether improvements stem from the additional linguistic features or from the different machine learning methods. Whereas the baseline uses SVMs, the examined method is based on a probabilistic framework closely related to CRFs.

Explicitly negative results with regard to the utility of linguistic features are for example reported by Ng et al. [279], Dave et al. [96], or Riloff et al. [325]. Ng et al. [279] extract lexicalized subject-verb, verb-object, and adjective-noun relations from a dependency parse. The tuples are used as additional features to an unigram model. But their results show that linguistic features do not perform better than a higher order n-gram model. Dave et al. [96] examine the use of triplets based on head, modifier, and relation type, but also find that these features are ineffective. Riloff et al. [325] examine a document representation based on "lexico-syntactic patterns" that encode the syntactic role of words in a sentence. For two out of three dataset no improvements are measured compared to a unigram or bigram baseline. Improvement on the one dataset is 1.5 percentage points, but statistical significance is unknown in this case. A further negative result is reported by Kudo and Matsumoto [221], who

[15] TF/IDF refers to a weight which is based on term frequency (TF) and inverse document frequency (IDF). See for instance Manning et al. [250, chap. 6.2]

[16] Although it does not incorporate class information, TF/IDF integrates knowledge taken from the complete evaluation corpus. The baseline weighting schemes, presence and term frequency, only have access to a single document.

conclude that in their experiments "n-gram features showed comparable performance to dependency features".

Our research revealed very few works that indicate that syntax features really help in sentiment polarity classification. One such work is for example by Wiegand and Klakow [422], who examine the utility of linguistic (including syntax) features for sentence level classification. They encode abstract linguistic properties, such as clause types, depth of constituents, main predicates, or part-of-speech tags as features. Adding these features to a bag-of-words baseline increases accuracy significantly by 4.5 percentage points. Their best performing model with all linguistic plus prior polarity features exhibits a 2.1 percentage point higher accuracy than a baseline with bag-of-words and prior polarity features.

We conclude as follows: Regarding the utility of syntax features, we have seen sound negative results and many works that argue in favor, but fail to provide explicit evidence for their claim. Considering this, the very few positive results we are aware of, and the costs (in processing time) that are involved with using syntax features, we are thus not convinced. We must assume that at present time, given the current accuracy of natural language parsers, it is unlikely that syntax features can improve classification results significantly.

10.2.4. Sentiment Shifter Features

We have seen that sentiment shifters are a major factor that render sentiment polarity classification more difficult than traditional text categorization. Capturing the effects of these shifters by modeling appropriate features may thus be a promising way to increase classification accuracy.

Negation and Intensification The most important shifter type is negation. In the context of polarity classification, one of the earliest work that explicitly considers negation is by Das and Chen [95]. Using a dictionary of common negation words, occurrences in the text are detected and relevant words in the context are suffixed with a negation marker. Pang et al. [297] adapted this simple method by adding a negation marker to each word following a negation up to the next punctuation symbol (e.g., "I did not like the hotel." becomes "I did not like_not the_not hotel_not."). Whereas Pang et al. [297] report a "negligible" positive effect, Dave et al. [96] observe a decreased accuracy with the same negation encoding.

Kennedy and Inkpen [205] study more sophisticated methods of creating features from sentiment shifters. In particular, they consider negation and modifiers such as amplifiers and downtoners. The scope of each shifter is determined by examining the output of a natural language parser. For each word that is associated with a shifter, they create a synthetic bigram feature, encoding the shifter type and the word itself. For instance, from the sequence "really do not like" they extract the bigram features "*amplifier*_like" and "*negation*_like". Their experiments show a marginal (however statistically significant) improvement of 0.7 percentage points over a baseline with unigrams only. Unfortunately they do not report results for a simple bigram model. We do not know whether this simpler, but potentially equally expressive model, may perform on the same level or even better.

Instead of using a lexicon or exploiting parse tree information, Ikeda et al. [181] propose to learn which terms function as sentiment shifters. Given a set of labeled sentences, they detect all sentiment expressions (lexicon-based) where the prior polarity contradicts the sentence label (i.e., they determine all shifted expressions). By examining the words in the context of shifted expressions, they can learn the sentiment shifters. In fact, a weight vector is learned which indicates the ability of individual words to shift polarity. The weights can be used as additional features for polarity classification. The results with this method are mixed. For a larger corpus (9,500 movie review sentences), a simple trigram bag-of-words model achieved the best results, whereas for a smaller corpus (1,400 customer review sentences), the enhanced model performed best (+3.1 percentage points). It seems that, if

enough training data is available, the more complex model does not provide any benefit. Additionally, it must be noted that their approach is only applicable to sentences which contain at least one sentiment expression.

Wiegand and Klakow [422] recognize negators, intensifiers, and neutralizers (modality indicators) by means of lexicons and parse tree information. Resulting features are encoded on the word (presence) and on the sentence level (frequency). They report that considering negation "did not notably increase the baseline performance". Unfortunately, the effect of the other features is not separately evaluated. In conjunction with other (abstract) linguistic features small improvements are reported. But we do not know whether the shifter features are relevant.

Councill et al. [89] explicitly concentrate on the detection of negation in customer reviews and manually annotate a corpus with fine-grained negation information. Based on features generated from a lexicon with negation clues and parse tree information, a CRF for negation scope detection is trained. Experiments which include the negation detector as part of a simple sentiment voting heuristic show improved results (the experiment is restricted to sentences that truly contain a negation). Unfortunately, no comparison with a supervised polarity classifier is reported.

A recent overview of modeling negation in the context of sentiment analysis is provided by Wiegand et al. [425]. Discussing various computational approaches, they finally conclude that "[...] in order to make general statements about the effectiveness of specific methods[,] systematic comparative analyses[,] examining the impact of different negation models [...][,] still need to be carried out."

Neutralization Besides negation, neutralization is the second most relevant shifter concerning sentiment polarity. A common type of neutralization becomes manifest in the use of conditional sentences. This particular form is for instance addressed by Narayanan et al. [274]. They construct patterns based on cue words and part-of-speech tags to detect condition and consequent clauses of conditional sentences. Verbs and modal auxiliary verbs that are found in such clauses are extracted as special features. Adding these features to a set of standard features (bag-of-words + prior polarity + softener cue words) improves the f-measure for binary, sentence level polarity classification by nearly 4 percentage points. However, in comparison to a model that includes these features and other (simpler) "condition features", it performs worse. We do not know whether the simpler features may already suffice to increase performance on conditional sentences. A rule-based system for detecting the "truth-value" of sentence is for instance presented by Kessler [208]. Such a system may also be used to create features for a supervised classifier. It is however unknown whether such an approach is beneficial for polarity classification.

10.2.5. Summary

Our goal in this section was to distill the most valuable insights regarding the text representation for document and sentence level polarity classification. We tried to answer the question which feature types have been found to consistently improve results compared to a simple bag-of-words baseline. To this end, we studied the relevant literature and categorized reported results by four different feature classes: namely, *lexical features*, *knowledge-based features*, *linguistic features*, and *sentiment shifter features*. Table 10.1 summarizes our detailed discussion in a compact way. For each feature type, we list the proponents and opponents as well as works that discuss the feature type, but either make no statement (neutral) or report mixed or inconclusive results. The most important results of our analysis are:

1. We found no convincing evidence that complex linguistic features are generally superior to simple bag-of-words models with higher order n-grams.

feature type	pro	con	neutral	mixed/inconclusive	conclusion
lexical					
n-gram	[92, 96, 279]	[297]	—	—	⊕
term masking	—	[96]	—	—	⊖
knowledge-based					
prior polarity	[237, 269, 273, 279, 422]	—	[40]	[21]	⊕
thesaurus	—	[96]	—	[337, 422]	⊖
review rating	[40]	—	—	—	⊕
entity masking	—	[96]	—	—	⊖
linguistic					
stemming	—	—	—	[96, 252, 422]	⊖
part-of-speech	—	[145, 272, 297]	[2, 422]	—	⊖
syntax	[422]	[96, 221, 279, 325]	—	[16, 135, 197, 273, 292]	⊖
sentiment shifters					
negation	[297]	[422]	[89, 425]	[181, 205]	⊕/⊖
neutralization	[274]	—	[208, 422]	—	⊕/⊖

Table 10.1.: Related work that addresses feature engineering in the context of document and sentence level polarity classification. The "neutral" column refers to works that discuss a specific feature type, but do not explicitly evaluate its influence. The "mixed" column refers to works which either report mixed results or where a closer look reveals inconclusive results. The last column represents our personal assessment of the utility of the different feature types (based on the literature review).

2. Including higher-order n-grams improves results over a unigram baseline. If training data is sparse, feature selection is important.

3. Adding features based on the prior polarity of words (e.g., derived from a sentiment lexicon) significantly improves classification accuracy.

4. Shallow linguistic features, derived from stemming, lemmatization, or part-of-speech tagging, do not seem to have significant effects.

5. Considering signals at the document level (e.g., the user provided review rating) improves the classification accuracy at the sentence level.

6. Different methods of generalizing from concrete lexical features (e.g., by term/entity substitution or by WordNet lookups) do not seem to have positive effects.

10.3. Exploiting Weakly Labeled Data for Sentence Level Polarity Classification

In Chapter 8 we have already seen that weakly labeled data may in fact reduce labeling costs and in addition can boost classification performance. In this and the following sections, we will examine the utility of weakly labeled data for sentence level polarity classification more closely. We collect appropriate data from pros and cons summaries of customer reviews. Before discussing more details, we define our concrete problem setting in Section 10.3.1. We will then reason about the availability and quality of such data, especially in comparison to document level, weakly labeled data (Section 10.3.2). Subsequently, Section 10.3.3 describes our extraction process and points out some filtering/cleansing techniques that increase the data quality. In the last part, Section 10.3.4, we discuss different ways of exploiting the extracted training data for polarity classification (including our approach with one-class classification).

10.3.1. Problem Description

We examine ternary, single label, sentence level polarity classification. Our setting is supervised. The goal is to learn a classification model that can categorize a sentence into exactly one of the three classes "positive", "negative", or "objective". All classes are defined in adherence to our annotation guidelines for sentence level annotation (see Section 5.2 and Appendix A.2.2). The class "objective" refers to sentences that neither explicitly, nor implicitly (i.e., polar facts) express any sentiment. According to our annotation schema, objective sentences are identified by an empty sentiment polarity label. We simplify the problem setting by ignoring "neutral" and "mixed/both" polarity sentences. We remove those sentences from our evaluation corpora. As our primary goal is extraction of positive and negative sentences from customer reviews, we are interested in achieving high f-measures for the "positive" and "negative" class.

10.3.2. Availability and Quality of Weakly Labeled Data

Document Level

In the introductory part, we pointed out that for document level sentiment classification training data is conveniently available, at least for the popular customer review mining task. Being more precise, *weakly labeled* data is conveniently available. Nearly all popular review sites force a reviewer to provide a numerical, overall rating. These ratings can be easily extracted and can serve as (weak) label for the related review. Quasi all works on document level sentiment classification both, train and evaluate their proposed methods on such data. Very strictly speaking, and as pointed out earlier, those results obtained by evaluating on a corpus where the labels are only estimated, do not necessarily mirror the true accuracy of the examined method. However, we argue that for the document level this use of weakly labeled data is justified, simply because the estimate is quite precise. Considering user-provided review ratings, we make the following observations:

- Review labels are generated by a human. As with a true gold standard corpus, the label decision is based on a human cognitive process.
- Labeling errors are unlikely. Three, five, or ten point rating scales are so common that misinterpretation is not likely (further assuming that a review author is also a review consumer).
- The label matches the unit of evaluation. In other words, the label truly describes a property of the document and the document is truly the unit of analysis.
- The concrete process for extracting a review label is easy. It is very unlikely that mistakes are introduced by erroneous extraction. Often data is available in machine-readable form (e.g., via some public API).

Sentence Level

Sentence level labels are not directly available from customer reviews. For good reasons, reviewers simply do not provide rating information for individual sentences. A quite naive approach to acquire sentence level labels would be to regard all sentences in a positively labeled review as positive and vice versa for negative reviews. Obviously, such a method is not very precise. We know that also very positive or negative reviews contain relatively mixed content (regarding the polarity). For example, our analysis of the digital camera review corpus has shown that, with regard to the label, only slightly more than 50% of the sentences follow the document label[17]. Roughly 35% are factual, the remainder has a different polarity label than the document. In other words, with this naive method we would

[17] The number is even lower if we also count the less extreme reviews with 2 or 4 star ratings as negative or positive.

generate a corpus with every second label being erroneous. It is doubtful that such a corpus helps in training a polarity classifier and surely it is not suited for evaluation purposes.

In the previous chapters we have seen that "pros" and "cons", as compact summaries of the main advantages and disadvantages of a product/service, also convey useful polarity information. The general assumption was that information from the pros exhibits positive polarity and vice versa for the cons. These summaries are much better suited to collect training data for polarity classification. Although the label (i.e., "pros" or "cons") is again not provided at the sentence level, we can exploit the fact that summaries are typically very homogeneous with regard to polarity. In a preliminary study, we randomly sampled 400 sentences (200 pros, 200 cons) from the pros/cons summaries of hotel reviews (taken from Priceline.com) and manually verified the (weak) label decision. We did the same for digital camera reviews (obtained from Epinions.com and Reevoo.com). For the hotel dataset, about 89% of the weak labels are correct (89.0% pros, 89.5% cons). In other words, 11% of the sentences taken from the pros do not exhibit a positive polarity and 10.5% of the cons sentences are not negatively connoted. For the camera dataset, the precision is a little lower with about 86% correct labels (87.5% pros, 83.5% cons). Wrong label decisions in both datasets are mainly due to objective/factual sentences, which apparently appear in pros/cons summaries. In Section 10.3.3 we describe filtering techniques that can improve the precision to (estimated) 97% for the camera dataset and 95% for the hotel dataset.

Despite such high precision values, we believe that a weakly labeled corpus obtained from pros/-cons summaries of customer reviews cannot function as a proper evaluation dataset for our purposes. Our goal is to classify sentences taken from the free text part of a review. These sentences are typically longer and more complex than sentences found in pros/cons texts. Summarizing the pros and cons of a product in a compact form is different from writing a complete review. The style of writing is different — for example, enumerations are much more common in summaries. Thus, to measure the performance of a polarity classification model for review sentences, we should not evaluate on a weakly labeled pros/cons dataset. Results achieved by such an evaluation may not represent the true performance (i.e., for classifying "normal" review sentences). We may use the weakly labeled samples perfectly as additional training data, but must use a manually labeled corpus for evaluation.

10.3.3. Extraction of Weakly Labeled Data from Pros/Cons Summaries

Types of Pros/Cons Summaries

Depending on the specific review site, pros/cons summaries differ in presentation and style. For some sites, very compact, often comma-separated enumerations prevail (e.g., Buzzillions.com), other sites encourage reviewers to provide longer descriptions of their pros and cons (e.g., Priceline.com or Reevoo.com). Whereas compact enumerations are perfectly suited for extracting target-specific prior polarity information (as we described in the previous chapter), they do not fit well when creating a corpus for sentence level classification. We are thus interested in sources that provide pros/cons summaries where longer descriptions with complete (optimally grammatical correct) sentences predominate.

Homogeneity of Pros/Cons Summaries

We have seen that the homogeneity with regard to sentiment polarity is already quite high in sentences extracted from pros/cons summaries. However, when using the raw data without some data cleansing, roughly more than 10% of the label decisions are incorrect. Our preliminary study revealed some typical patterns that induce such erroneous decisions:

- Listing aspects: A common pattern is to simply name the aspects that are liked/disliked. For instance, in a positive hotel summary, we may find a text such as "The location near the airport

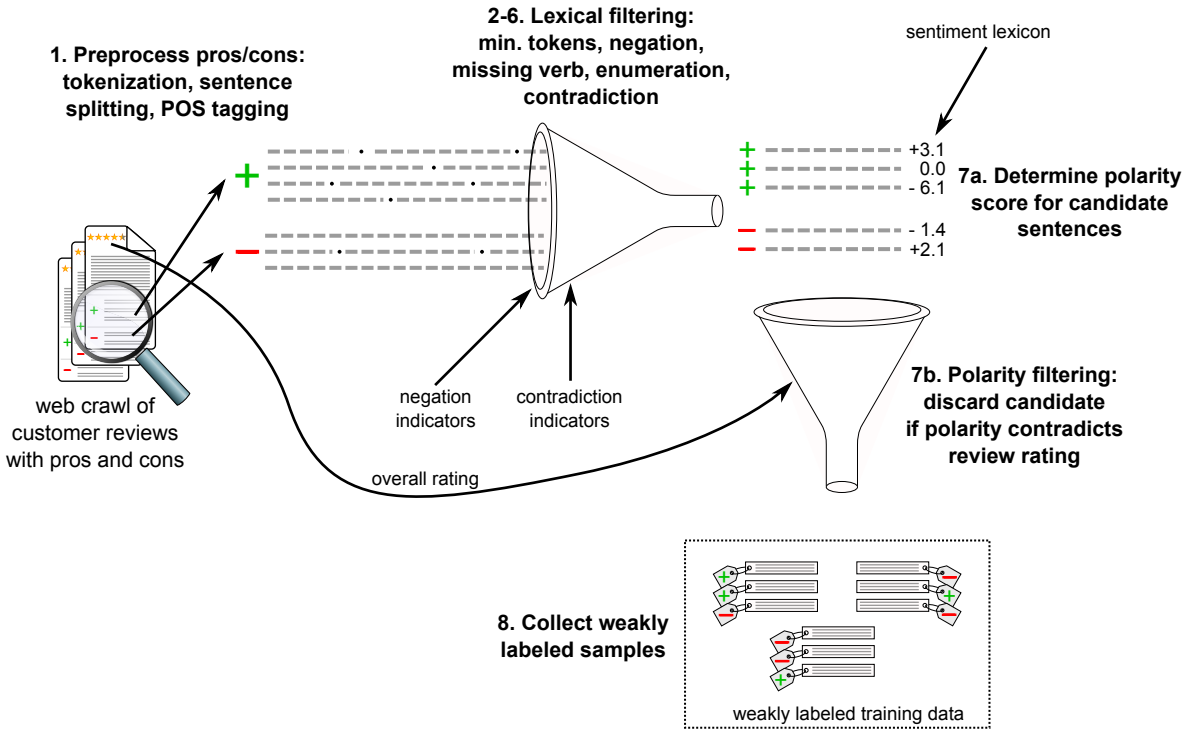


Figure 10.3.: The filtering and data cleansing steps.

and the shuttle service." Contained in the pros, we know that the reviewer liked the proximity to the airport. However, the text may stem equally well from the cons (airport means noise, the shuttle service may be bad). The text alone does not convey any particular sentiment polarity.

- Absence of pros/cons: Another common pattern is to express the non-existence of any advantages/disadvantages. For instance, we may read a sentence such as "Can't think of any, the whole experience was awful and I'll never go to this place again." in the pros. On the other hand, in the cons we might read a sentence such as "Nothing, everything from staff to room was perfect." Often, as shown in the preceding examples, pointing out the absence of any pros (cons) is accompanied by rephrasing the bad points (good points).

- Comparison/Contradiction: Mixed sentiment is often conveyed by comparing/contradicting a positive aspect with a negative aspect. For example, we may find a sentence such as "The room was clean and comfortable, however quite small." Sentences with mixed polarity must be considered as erroneous (for our task) regardless of whether being extracted from the pros or the cons.

Filtering and Cleansing of Pros/Cons Summaries

Based on the previously mentioned patterns, we heuristically detect and remove potentially erroneous sentences to increase the precision of the (weak) label decisions. Fig. 10.3 gives an overview of this filtering process. In particular, we propose the following preprocessing and cleansing steps:

1. Preprocessing: We use the Stanford CoreNLP tools for tokenization, sentence splitting, and part-of-speech tagging of the pros/cons summaries.

2. Minimum length: We remove each sentence which consists of less than 4 tokens.

Algorithm 10.1 Sentence Polarity Score Calculation

Requires: a (context-aware) sentiment lexicon, a product aspect lexicon, a lexicon with negation indicators
Input: a sentence s
Output: a polarity score $p(s)$ for sentence s

1. Apply the Aho-Corasick algorithm [5] to find all non-overlapping matches of sentiment lexicon entries E in s. Each match is a potential sentiment expression.
2. Analogously, find all product aspect mentions A in s.
3. Analogously, find all negation indicators N in s
4. For each sentiment expression $e \in E$ calculate its contextual polarity score p_e:
   a) To calculate the target specific polarity consider the following three cases:
      i. sentiment expression e has only target specific polarity: Check in a window of five preceding and five succeeding tokens whether the aspect $a \in A$, with minimal distance to e (ties solved by preferring the succeeding over the preceding a), is associated with e in the sentiment lexicon. If yes, set p_e to this target specific polarity value, otherwise remove e from the set of sentiment expression E.
      ii. sentiment expression e has target specific and target independent polarity: Same as before, but set p_e to the independent polarity score if the closest a is not associated with e.
      iii. sentiment expression e has only target independent polarity: Set p_e to the target independent polarity score.
   b) If a negation indicator precedes e with a distance of at most two tokens, flip the polarity by setting $p_e = (-1) * p_e$
5. return $p(s) = \sum_{e \in E} p_e$

3. Absence: We create a dictionary (approx. 200 entries) of very common lexical patterns that are used to describe the non-existence of pros or cons (e.g., "nothing", "hardly any", "loved it all", "haven't found", "not come across", etc.). We examine the first sentence of a pros (cons) text and if one of the pattern matches, we discard the entire text (i.e., also all other sentences in that snippet are dropped). If the first sentence denies the existence of any pros (cons), then all following information is likely to be in contrast to the label.

4. Missing verb: We remove each sentence that does not contain a verb. For example, this prevents us from extracting enumerations of aspects.

5. Enumeration: We remove each sentence where the ratio of separator symbols (commas, semicolons) and tokens is above 1:3. This heuristic removes some more aspect listings that have not been detected by the previous step (for instance due to POS tag errors or because some aspect term contains a verb).

6. Contradiction: We discard each sentence that contains a "contradiction indicator". For this purpose, we compiled a list of such indicators (e.g., "however", "despite", "although", "but", "other than", etc.). The intuition for this filter is to prevent the inclusion of mixed polarity sentences.

7. Sentiment lexicon: We use our domain-adapted, context-aware sentiment lexicons (see previous chapter) to calculate a polarity score for each candidate sentence. We apply Algorithm 10.1 to derive a sentence level score that is based on prior polarities and a simple vote flip heuristic for handling negations. Depending on this score and the overall rating of a review, we decide whether a sentence is added to the training data or whether it is discarded. Algorithm 10.2 formally describes this decision process. It is further summarized graphically by Fig. 10.4. The basic intuition is to discard all sentences where the calculated polarity score drastically contradicts the overall review rating.

Algorithm 10.2 Polarity Score Filter

Input: candidate sentence c, weak label l, review rating r (on five-point scale)
Output: *true* (keep), *false* (discard)

```
s ← calculatePolarityScore(c)                          ▷ use Algorithm 10.1
if l = 'positive' then
    if s > 0 then                                      ▷ keep c if polarity score is positive
        return true
    else if s ≥ −5 and r ≥ 4 then                      ▷ keep c if score is not "too" negative and rating is at least 4
        return true
    else
        return false
    end if
else if l = 'negative' then
    if s < 0 then                                      ▷ keep c if polarity score is negative
        return true
    else if s ≤ 1 and r ≤ 3 then                       ▷ keep c if score is only slightly positive and rating is <= 3
        return true
    else if s ≤ 5 and r ≤ 2 then                       ▷ keep c if score is not "too" positive and rating is <= 2
        return true
    else
        return false
    end if
end if
```

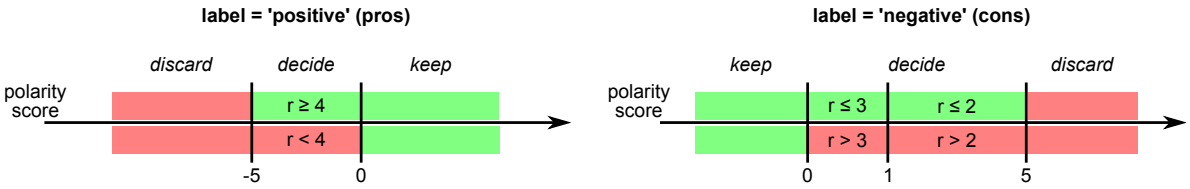


Figure 10.4.: Illustration of Algorithm 10.2. The variable r refers to the (scaled/mapped) overall rating of a review. The colors indicate the decision value (green: "keep", red: "discard").

Weakly Labeled Data for the Objective Class

Sentences extracted from the pros/cons summaries of customer reviews either belong to the "positive" class or to the "negative" class. As our goal is to learn a ternary polarity classification model, the lack of objective training samples is a problem. In the context of movie reviews, Pang and Lee [295] exploit plot descriptions crawled from IMDb.com to obtain samples for the objective class. They point out that the label decision is mostly correct, but "[. . .] plot summaries can occasionally contain subjective sentences that are mis-labeled as objective."[18] Unfortunately they do not examine the concrete precision of this heuristic.

Inspired by this idea, we experimented with extracting product/service descriptions from appropriate sources. In particular, for the hotel domain we collected descriptions from Priceline.com and Booking.com. For the digital camera domain, we used Amazon.com. Preliminary results with extracting objective sentences from this data and using these as objective training samples were unsatisfactory. The product descriptions are too noisy (with regard to the label) and do not really fit the type of objective sentences we find in customer reviews:

- Most product descriptions are simply not objective. In fact, the vast majority of descriptions seems to originate from the marketing department of the respective manufacturer or hotel owner. Clearly, such a description praises the product instead of providing objective facts.

[18] Cited from the README file describing the corpus: http://www.cs.cornell.edu/people/pabo/movie-review-data/subjdata.README.1.0.txt

- Objective sentences in typical product descriptions differ widely from objective sentences found in customer reviews. This is best explained by considering the prevailing discourse functions. In a product description the function is either "praise the product" or "describe specification" (e.g., available amenities for hotels). Thus, in case we find an objective sentence in a product description, it most likely refers to the product specification. However, we know that in customer reviews much more diverse discourse functions exist (e.g., "advice", "general remark", "personal context", "usage"). This type of sample data is not available in product descriptions.

For movie reviews, the situation is apparently different. Most objective information provided in movie reviews refers to the plot of the movie. In consequence, the types of sentences extracted from (objective) plot summaries and the types of sentences found in the objective parts of movie reviews are very similar. Unfortunately, for product and service reviews this observation is not true. We cannot use product descriptions for extracting training samples for the objective class.

Another heuristic to obtain weakly labeled samples for the objective class is proposed by Kim and Hovy [210]. They collect phrases from the pros/cons summaries and search for them in the review text. Sentences containing such a phrase are labeled as positive or negative, respectively. All remaining sentences in the review are labeled as objective samples. In comparison to the previously discussed approach, this method guarantees that weakly labeled samples match the style of the target dataset. However, they also do not examine the accuracy of their heuristic. We believe that it is too naive and doubt that it allows to collect an adequate set of samples for the objective class. Subjective sentences in the review that do not have a "partner" in the pros/cons get labeled as objective. This may happen either because the "mapping heuristic" is not perfect or simply because the review mentions more or other pros/cons than the summaries do. In conclusion, obtaining weakly labeled data for the objective class remains a problem. We did not find an appropriate data source, nor an adequate extraction heuristic.

10.3.4. Incorporating Weakly Labeled Data into a Polarity Classifier

Whereas in the previous section we described how to obtain weakly labeled samples from pros and cons of customer reviews, we now discuss how to incorporate this data into a supervised approach to ternary polarity classification. We mainly differentiate between three ways of incorporating the data:

- **Enrich the manually labeled dataset**: We may simply incorporate the weakly labeled positive and negative samples as additional training data to a manually labeled corpus. Fig. 10.5 illustrates this approach. To obtain a ternary polarity classifier, a multi-class machine learning algorithm[19] is provided with positive, negative, and objective samples from the manually labeled corpus, as well as with additional positive and negative samples from the weakly labeled corpus. Typically, the amount of weakly labeled training data is significantly larger than the amount of manually labeled data (e.g., in our experiments the ratio is up to 100:1). Whereas this procedure does not eliminate the need for manually labeling training data, the goal is to improve classification performance by drastically increasing the number of training samples.

- **Only objective samples are manually labeled**: The goal of this approach is to significantly reduce the effort involved with providing training data. For the positive and negative polarity classes we use the weakly labeled data only and completely go without manually labeling positive and negative samples. We only provide manually labeled samples for the objective class. Thus, the total manual effort is reduced to about one third of the original amount of work.

[19] Learning a multinomial classification model may be implemented in different ways. We may apply a machine learning algorithm that naturally models a multinomial outcome variable (e.g., multinomial logistic regression, see Section 8.3). We may alternatively use strategies such as one-vs.-rest (OvR) or one-vs.-one (OvO) that combine multiple binary classifiers to support for multinomial outcomes (see for example Hsu and Lin [174]). Additionally, we may follow a cascaded approach as described in Section 10.1.1.

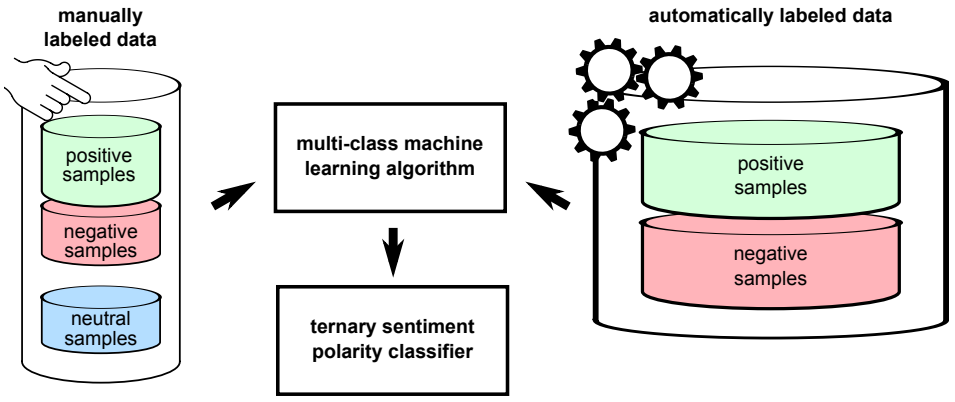


Figure 10.5.: Learning a ternary polarity classifier by enriching a manually labeled dataset with weakly labeled data.

As before, we can use some multi-class machine learning scheme to obtain a ternary polarity classifier.

- **Use weakly labeled data only**: Following this approach, we completely go without providing manually labeled data. In consequence, and as discussed in the previous section, no samples for the objective class are available. To still allow for ternary polarity classification in this setting, we consider the use of a one-class classification algorithm. Such an algorithm is capable of learning a binary classification model when training samples for only one class are available and samples for the other class are missing. As shown in Fig. 10.6, we use the one-class machine learning method to train a subjectivity classifier that distinguishes objective from polar/subjective sentences. We construct samples for the "subjective" class by combining the sets of positive and negative polar samples. In addition, we train a traditional binary polarity classifier, again using the weakly labeled positive and negative samples. We follow the "cascaded approach" to combine both classifiers to a ternary polarity classifier.

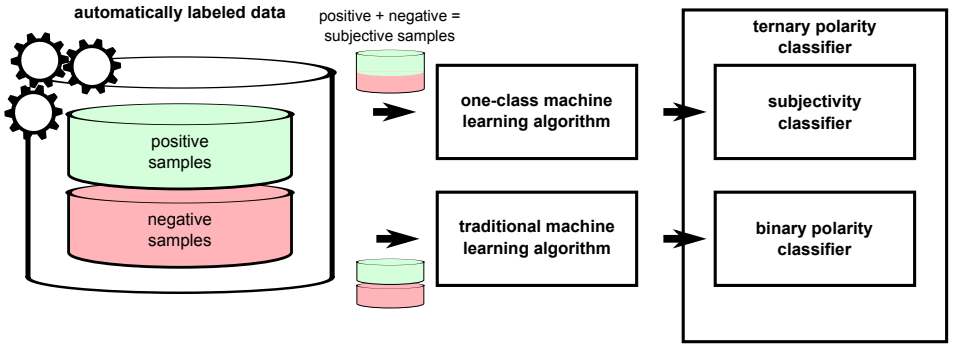


Figure 10.6.: Learning a ternary polarity classifier when using weakly labeled data only.

Whereas the first two approaches of incorporating the weakly labeled data make use of conventional, supervised classification algorithms, the third approach involves the application of a rather uncommon technique, namely the use of one-class classification methods. The following paragraph therefore provides some more details on one-class classification.

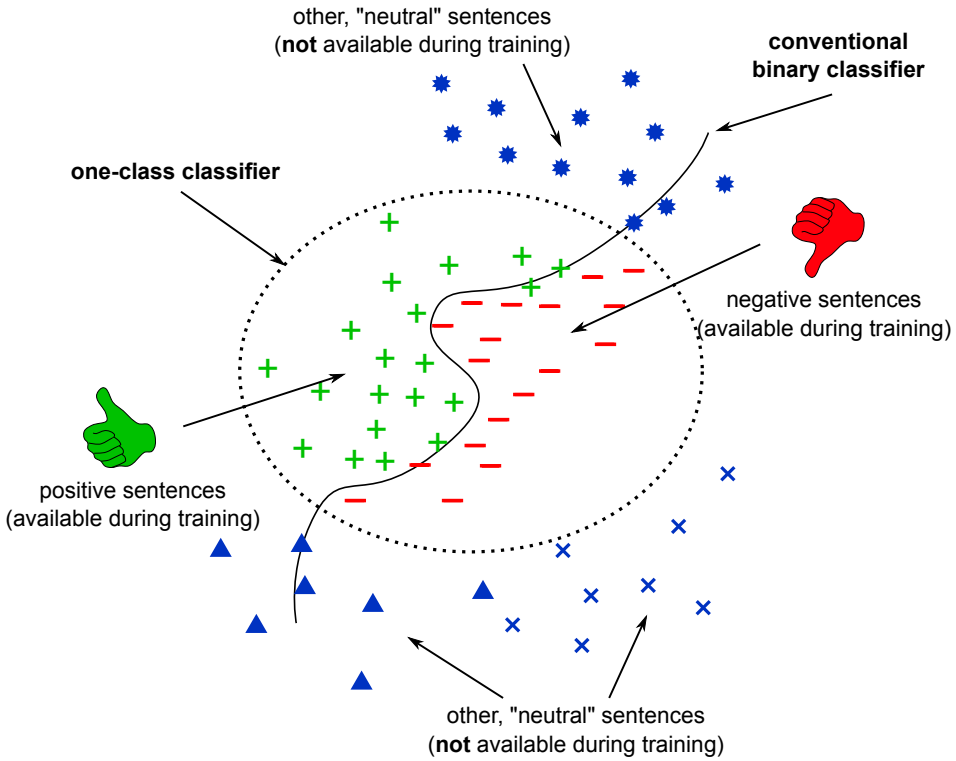


Figure 10.7.: Conventional, binary classification vs. one-class classification for sentence level polarity classification. The binary classifier distinguishes positive from negative sentences and is trained on samples for both target classes. The one-class classifier distinguishes subjective/polar sentences from other (objective) sentences and is trained on polar sentences only. (illustration inspired by Tax [379, fig. 1.1])

One-class Classification

We use one-class classification methods simply as a tool for polarity classification with weakly labeled data. Thus, we only point out the main concepts of this methodology. A much more comprehensive coverage of one-class classification can be found in the PhD thesis of Tax [379]. He outlines the underlying problem of one-class classification as making "a description of a target set of objects and to detect which (new) objects resemble this training set." The name "one-class classification" stems from the fact that the classifier is trained on a single class (in the following denoted as *target class*) only. Samples for the other class (*outliers*) may not be available or may be extremely costly to obtain. Thus, in contrast to a conventional setting, a boundary which separates the target class and outliers has to be learned from samples of the target class only. Tax [379] describes the resulting task as "to define a boundary around the target class, such that it accepts as much of the target objects as possible, while it minimizes the chance of accepting outlier objects."

Fig. 10.7 illustrates the relationship between one-class and conventional, binary classification in the context of our polarity detection task. Using the weakly labeled data extracted from pros/cons summaries, positive and negative sentences are available in the training set, but objective sentences are missing. We can easily train a binary classifier to distinguish positive and negative polarity (indicated by the solid boundary). For the one-class classifier (indicated by the dashed ellipse) we regard the entire set of polar sentences (positive and negative) as the target class. Objective sentences are considered as outliers. If the boundary that is learned by the one-class classifier is "tight", we can detect these outliers and thus can distinguish between subjective/polar and objective sentences. Besides our polarity classification scenario, other application domains for one-class classification are for example

the detection of machine faults [346], authorship verification [218], or intrusion detection [142]. Also, depending on the concrete domain of application, different terms are used to describe the basic problem addressed by one-class classification. The most common other names are *novelty detection*, *outlier detection*, or *concept learning*.

For our experiments with one-class classification, we will use the implementation provided as part of the LIBSVM library[20][69]. This implementation is based on a methodology introduced by Schölkopf et al. [335]. With this approach, one-class classification is formulated as learning a function that is positive for samples from the target class and negative for other samples. It can be regarded as an extension of the support vector machine (SVM) methodology to one-class classification. More detailed information concerning the concrete implementation is provided by Chang and Lin [69][21].

Besides one-class classification with SVMs, other "boundary methods" (e.g., based on k-centers or nearest neighbor approaches) have been suggested. Also non-boundary methods (e.g., based on density estimation or prototype reconstruction) are considered [379]. Manevitz and Yousef [248] examine these different methods in the context of text categorization, which is similar to our setting. They compare the SVM based approach to other approaches such as (adaptations of) the Rocchio algorithm, the nearest neighbor algorithm, the naive Bayes algorithm, or neural networks. They report that the SVM approach is superior to all other methods, except the neural network, which shows comparable performance. We thus conclude that using the LIBSVM implementation of one-class classification is a reasonable choice for our experiments.

To complete the picture, we note that a similar problem to one-class classification is to learn classifiers from only positive (i.e., the target class) and unlabeled data. For example, in our case this would mean to learn from the weakly labeled polar sentences and additionally consider sentences extracted from associated review texts as unlabeled data (containing a mix of polar and non-polar sentences). As can be seen, such a semi-supervised setting also fits our problem description (samples from the target class, as well as relevant unlabeled data is easily available). However, in this work we do not examine these semi-supervised methods and leave that for future work. Approaches to learn from only positive and unlabeled data are for instance proposed by Elkan and Noto [117], Blanchard et al. [41], Liu et al. [235], or Yu et al. [456].

10.4. Experiments and Results

In this section we discuss our experiments and results with sentence level polarity classification. The remainder of this section is organized as follows: We first describe our basic experimental setup in Section 10.4.1. Subsequently, and to take up the discussion of Section 10.2, we experiment with varying document representations (actually: sentence representations). In particular, we examine the effectiveness of various preprocessing steps (Section 10.4.2) and consider the effects of different feature sets, testing lexical, knowledge-based, and linguistic features (Section 10.4.3). In Section 10.4.4 we compare the use of a cascaded approach to ternary polarity classification, instead of using a conventional multi-class classification strategy. We are further interested in whether the addition of weakly labeled data helps in improving classification performance. We first consider the utility of weakly labeled data for the task of binary polarity classification in Section 10.4.5. We experiment with different amounts of manually and weakly labeled data. Additionally, we examine the interplay of varying amounts of training data and varying order of n-gram features (Section 10.4.6). Effectiveness of our heuristics for cleaning the weakly labeled data are analyzed in Section 10.4.7. Subsequently, we consider the task of subjectivity detection in the context of training with weakly labeled data. Section 10.4.8 covers our experiments in this direction, including the one-class classification approach.

[20] available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[21] A regularly updated version of the article is provided at http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf.

10.4.1. Experimental Setup

Datasets

- Gold standard: We evaluate the different classification models by means of our sentence level, manually labeled gold standard corpora (see Section 5.2). We use the handcrafted sentiment polarity annotation as label for each sentence. Clearly, sentences which are annotated with positive (negative) polarity are labeled as "positive" ("negative"). Non-polar sentences (empty sentiment label) are added as samples for the "objective" class. We remove all sentences that are labeled with "neutral" or "mixed" polarity. In other words, we disregard the more fine-grained distinctions introduced with the "mixed polarity" and "neutral" classes. As noted earlier, this decision simplifies the problem setting. Reported results will be slightly better compared to solving the more difficult problem with the original five classes. However the number of samples in the "neutral" and "mixed" classes are too low to build adequate classifiers. Table 10.2 gives an overview of the class distribution in the simplified gold standard corpora.

	hotel		camera	
class label	samples (proportion)	polar facts	samples (proportion)	polar facts
positive	1370 (46.3%)	9.6%	1201 (40.8%)	6.7%
negative	761 (25.7%)	37.1%	647 (22.0%)	32.3%
objective	829 (28.0%)	—	1094 (37.2%)	—
total	2960	14.0%	2942	9.9%

Table 10.2.: Basic statistics of the evaluation datasets for polarity classification. The column "polar facts" refers to the proportion of polar fact sentences for a particular class label.

- Training datasets: Depending on the experiment, we train classifiers on the gold standard corpus, the corpus of weakly labeled data, or on some combination of both corpora. We use cross validation to guarantee generalizable results.

- Weakly labeled data: As described earlier, we extract weakly labeled data for the "positive" and "negative" classes from pros and cons summaries of customer reviews. For the hotel review domain, we extract the relevant data from Priceline.com. For the digital camera review domain, we crawl data from Reevoo.com and Epinions.com. After preprocessing, we obtain more than half a million sentences for the hotel domain and slightly more than 150,000 sentences for the camera review domain. From the hotel dataset we randomly sample 200,000 sentences, where positive and negative sentences are equally distributed. From the camera review dataset, we sample 100,000 sentences, also equally distributed among the positive and negative class.

Cross Validation and Statistical Significance

We use ten times repeated 10-fold cross validation for all experiments if not otherwise stated. To guarantee comparability of results, we ensure that for all experiments the same (ten) partitionings are used. For each 10-fold cross validation instance, we compute the means of the results for each fold. These mean values serve as the basis for deriving the statistical significance of deviations between different experiments. We use a paired t-test and if not otherwise stated, we report significance at the 99% confidence level. In addition, we ensure that the test folds only contain data from the gold standard corpora and never contain weakly labeled data (see also Fig. 10.9 on page 239).

component	parameter	value	description
corpus	min-sentence-length	4	the minimum length of sentences in tokens to be considered during training and evaluation
corpus	min-feature-count	2	the minimum support of a feature within the training data (basic feature selection)
corpus	k	10	the number of folds for cross validation
SVM	C	0.1	the cost parameter used during training of SVM models
SVM	ν	0.1	the offset parameter used during training of one-class SVM models
lexicon classifier	$\tau_{\textit{sentiment-score}}$	0	the threshold parameter for lexicon-based subjectivity detection

Table 10.3.: Parameter settings for the experiments with sentence level polarity classification.

Classification with Support Vector Machines

We use support vector machines (SVM) for classification. More specifically, we use the LIBLINEAR implementation [126] provided by the LIBSVM authors [69]. Regarding our use of SVMs, the following aspects are important to note: As suggested by the LIBSVM authors[22], if necessary, we scale the input data (the feature values) prior to learning or applying a classification model. In particular, we linearly scale the lexicon, shifter, rating, and context feature values to the interval $[0, 1]$ (all mentioned feature types are introduced in the following section). The unigram, n-gram, and pattern features are all binary (i.e., values are zero or one) and thus do not need to be scaled. We further follow the LIBSVM authors' suggestion by using a linear kernel. Our setting is comparable to document classification in the sense that the dimensionality of the feature space is extremely large. For example, when using the weakly labeled data, we easily find more than 10^5 distinct n-grams (order = 3) or about 10^3 unigrams. For the case with weakly labeled data the number of training instances and the number of features is very large. Using the manually labeled data only, the number of features is much larger than the number of training samples. In both cases, simple linear kernels are generally sufficient due to the high dimensional feature space. The LIBLINEAR library is particularly tuned for such settings and allows for much more efficient training of large-scale, sparse datasets [126]. When using a linear kernel, only a single hyperparameter — namely, the cost (or soft margin) parameter C — needs to be set. If not otherwise stated, we set $C = 0.1$ for all experiments.

Parameter Values

A set of different parameters need to be considered throughout the various experiments. Table 10.3 summarizes the most relevant parameters and their values.

Evaluation Metrics

In most experiments, we perform ternary polarity classification — that is, we distinguish between positive, negative, and objective sentences. In our context this classification task is embedded as a tool for customer review mining. We are thus primarily interested in the performance with regard to detecting positive and negative sentences. The performance with regard to objective sentences is rather irrelevant. In most cases, we therefore report results for the positive and negative class only and neglect the objective class. Most commonly, we report the macro and micro-averaged f-measures, computed over the positive and negative class. These measures adequately describe the systems accuracy in detecting sentences belonging to one of the target classes.

[22] http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

	hotel		digital camera	
setting	**macro-F1**	**micro-F1**	**macro-F1**	**micro-F1**
baseline	70.2	73.3	66.8	70.2
lemma	71.9 (+1.7)*	74.7 (+1.5)*	65.2 (-1.6)*	68.9 (-1.4)*
pos	71.2 (+1.0)*	74.3 (+1.0)*	66.2 (-0.6)*	69.5 (-0.7)*
downcase	71.8 (+1.5)*	74.7 (+1.5)*	66.6 (-0.2)	70.0 (-0.3)*
downcase+pos	71.2 (+1.0)*	74.3 (+1.0)*	66.2 (-0.6)*	69.5 (-0.7)*
aspect-remove	71.4 (+1.2)*	74.5 (+1.2)*	67.0 (+0.2)	70.4 (+0.1)
aspect-mask-same	71.1 (+0.9)*	74.2 (+0.9)*	66.7 (-0.1)	70.2 (+0.0)
aspect-mask-canonical	71.2 (+1.0)*	74.4 (+1.1)*	66.7 (-0.1)	70.0 (-0.2)*
downcase-remove	71.4 (+1.2)*	74.5 (+1.2)*	67.0 (+0.2)	70.4 (+0.1)
downcase+mask-canonical	71.2 (+1.0)*	74.4 (+1.1)*	66.7 (-0.1)	70.0 (-0.2)*

Table 10.4.: Effectiveness of varying linguistic preprocessing steps. The star symbol indicates statistical significance at the 99% confidence level (with regard to the deviation from the baseline, paired t-test).

10.4.2. Unigram Baseline and Shallow Linguistic Preprocessing

The goal of our first experiment is to set a baseline and to learn about the influence of different linguistic preprocessing steps. For each different setting, we train an SVM model, using unigram features only. With regard to linguistic preprocessing, we consider the following options:

- Term normalization: We experiment with lemmatization and simple downcasing.
- Word sense disambiguation: For a shallow form of word sense disambiguation, we append each token with its corresponding part-of-speech tag. We use only the first two characters of the tag (e.g., "VBZ" becomes "VB").
- Modify aspect mentions: We experiment with three different forms of modifying recognized product aspect mentions. We use a lexicon to detect the aspect mentions. Then, we either remove all aspects from the token sequence in a sentence ("aspect-remove"), we replace each aspect mention with a common symbol ("aspect-mask-same"), or we mask each aspect with its canonical form as defined in the lexicon ("aspect-mask-canonical"). The basic intuition is again to normalize different terms so that the individual features allow for better generalization.

The results in Table 10.4 show the macro and micro-averaged f-measure for the positive and negative class. For the baseline setting we do not perform any other preprocessing than tokenization. With this basic unigram model, we achieve micro-averaged f-measures of 73.3% for the hotel dataset and 70.2% for the digital camera dataset, respectively. Looking at the data for the other settings and comparing the two different corpora, we find that the results are rather inconclusive. Whereas most of the shallow linguistic preprocessing steps are helpful concerning the hotel dataset, we observe no statistically significant improvements for the digital camera corpus. For example, while modifying aspect mentions in the hotel dataset shows a significant improvement of 0.9 to 1.2 percentage points in micro-averaged f-measure, we see no significant effects for the camera dataset. Considering lemmatization, we even observe that results are significantly worse for the camera corpus, while they are significantly better for the hotel dataset. Our inconclusive results are in line with related work in document level sentiment classification (see Section 10.2). Also in the literature we find positive and negative results regarding the utility of different preprocessing steps. We simply conclude that selecting appropriate preprocessing steps is dependent on the dataset and its effects should be evaluated for each individual case. In general, no huge improvements are to be expected. For all following experiments, we decide to lowercase the terms contained in unigram or n-gram features.

feature name	description	type
lexicon label	two binary features "label-pos", "label-neg" that indicate the sentence polarity (score >1 or <-1)	binary
raw polarity score	the polarity score as calculated by Algorithm 10.1	continuous
raw pos/neg score	two features representing the sum of all positive/negative polarity scores in a sentence	continuous
purity	purity $= \frac{\text{raw-score}}{\sum_{p \in S} \lvert p \rvert}$, where p is a polarity score calculated for a phrase in sentence S, see also [40]	continuous
polar phrases count	two features representing the number of positive and negative polar phrases	integer
polarity count by POS tag	ten features representing the number of positive/negative adjective/noun/verb/adverb/other polar phrases	integer
polarity score by POS tag	ten features representing the polarity score of positive/negative adjective/noun/verb/adverb/other polar phrases	continuous

Table 10.5.: List of lexicon-based sentiment polarity features.

10.4.3. Effectiveness of Different Feature Types

We now experiment with different features types. Based on the literature analysis conducted in Section 10.2, we consider the following different types:

- **N-grams**: For each sentence, we extract all n-grams up to an order of 3 (i.e., all lower order n-grams, bigrams and unigrams in this case, are also extracted as features). As mentioned earlier, we lowercase each token, but do not perform any lemmatization or stemming, nor do we distinguish different parts of speech.

- **Sentiment lexicon**: For each sentence, we calculate different scores based on the (target-specific) prior polarity information encoded in a sentiment lexicon. Table 10.5 lists all different scores we compute. Each score is used as a separate feature. Most of the proposed polarity scores are inspired by the works of Wiegand and Klakow [422] and Blair-Goldensohn et al. [40]. Algorithm 10.1 describes how we apply a sentiment lexicon to calculate the polarity of a sentence. The algorithm performs a basic negation detection and, in case, simply flips the sign of the estimated polarity score.

- **Sentiment shifters**: We compile different dictionaries that contain typical amplifiers (intensifier and downtoner), negators, and neutralizers. Using the lexicons, we match relevant tokens/phrases in a sentence. As features we extract the number of amplifiers, the number of amplified expressions[23], the number of negators, and the number of neutralizers.

- **Sentiment pattern**: We build patterns based on part-of-speech tag sequences and sentiment related information. Based on lexicon matches, we replace each sentiment expression, each sentiment shifter, and each product aspect mention with a specific symbol. Depending on the polarity score, each sentiment expression is masked with a symbol which follows the pattern `sentiment-[pos|neg]-[0|1|5]`. The suffix represents whether the absolute score is less or equal one, less or equal five, or greater than five. For instance, a phrase with polarity score -3.4 is masked as `sentiment-neg-1` and a token with score 0.7 is replaced by `sentiment-pos-0`. Sentiment shifters are simply masked by a symbol representing the shifter type. For instance,

[23] multiple amplifiers may be associated with a single expression

a negator such as "never" is masked by `negation`, an amplifier such as "very" is masked by `amplifier`, and a neutralizer such as "if" is replaced by the symbol `neutralizer`. All product aspect mentions are masked by the single symbol `aspect`. All remaining tokens (i.e., all lexical information) are replaced by the corresponding part-of-speech tags. Again, only the two character prefixes of the original Penn Treebank tags are used. Putting it all together, for example the sentence "I really did not like the noisy air conditioning system.", is mapped to the sequence

```
PR amplifier VB negation sentiment-pos-1 DT sentiment-neg-1 aspect.
```

We extract sentiment patterns by constructing all n-grams from this sentence representation (with $2 \leq n \leq 6$). Each such n-gram is extracted as a feature.

- **Review rating**: We map the overall, user-provided rating of a review to an ordinal five-point scale and use this value as a feature. The intuition is that the overall rating correlates with the class labels of individual sentences (Blair-Goldensohn et al. [40] report improved results with this feature). Naturally, the feature can only be used if we train (and apply) a model on texts which provide such information. For instance, when classifying sentences extracted from blogs or user forums, the feature is simply not available. Also, if we train a model on weakly labeled data from the pros/cons summaries, we cannot use this feature type. There exists no correlation between the review rating and the polarity of a sentence in the pros or cons (our assumption is that "pros sentences" are always positive, independent of the rating and vice versa for "cons sentences").

- **Context**: Using a prefix and suffix window, we add the total raw polarity score, the total purity score (see Table 10.5), and the total number of positive (negative) expressions in the context of a sentence as features. The intuition is that having information on the polarity of preceding and following sentences may help to classify the sentence under consideration. The size of context windows is defined by the number of preceding/following sentences. This feature is inspired by the contextual features used by Blair-Goldensohn et al. [40]. For the same reasons as before, this feature cannot be used in conjunction with weakly labeled data from the pros/cons summaries. We experimented with different window sizes and found that for our datasets a size of length 2 is optimal. We only report results for this setting.

Results — Feature Engineering

We summarize our results of experimenting with the various feature types in Tables 10.6 and 10.7. The first row in each table represents the unigram baseline (with lowercasing). Differences to this baseline are indicated by the numbers in parentheses. We first evaluate the usefulness of the different feature types separately by including the particular feature information together with the basic unigram feature. For those feature types that individually show improved results, we also test feature combinations.

Regarding the individual features, the most significant improvements are obtained for the sentiment lexicon and review rating features. We observe a consistent, statistically significant improvement for both evaluation corpora. Considering the macro-averaged f-measure when adding the lexicon feature (setting "uni+lexicon"), we find an increase of 4.0 percentage points with regard to the hotel review dataset and an increase of 5.0 percentage points concerning the digital camera corpus. While both, the positive and negative class benefit from adding the lexicon feature, the more significant improvement is for the negative class. The same observation is true with regard to the review rating feature. The overall improvement in macro-averaged f-measure with this feature is slightly less compared to the sentiment lexicon features. The increase is 3.8 percentage points for the hotel dataset and 3.5 percentage points for the camera corpus. When combining both feature types (setting

	positive			negative			average	
feature	**precision**	**recall**	**f1**	**precision**	**recall**	**f1**	**macro-f1**	**micro-f1**
unigram	79.8	83.5	81.6	62.9	61.2	61.9	71.8	74.7
ngram	79.3 (-0.5)	81.5 (-2.0)*	80.4 (-1.2)*	63.4 (+0.5)	63.3 (+2.1)*	63.3 (+1.3)*	71.8 (+0.1)	74.3 (-0.4)
lexicon	74.3 (-5.5)*	81.6 (-1.9)*	77.8 (-3.8)*	61.4 (-1.4)*	64.8 (+3.6)*	63.0 (+1.1)*	70.4 (-1.4)*	72.6 (-2.2)*
uni+lexicon	82.3 (+2.5)*	84.7 (+1.2)*	83.4 (+1.8)*	70.3 (+7.4)*	66.3 (+5.2)*	68.2 (+6.2)*	75.8 (+4.0)*	78.1 (+3.4)*
uni+rating	81.3 (+1.4)*	86.6 (+3.1)*	83.8 (+2.2)*	68.1 (+5.2)*	66.7 (+5.5)*	67.3 (+5.4)*	75.5 (+3.8)*	78.1 (+3.3)*
uni+ctxt	80.1 (+0.3)	85.1 (+1.6)*	82.5 (+0.9)*	66.3 (+3.5)*	62.4 (+1.2)*	64.1 (+2.2)*	73.3 (+1.6)*	76.2 (+1.5)*
uni+goldctxt	80.8 (+1.0)*	85.6 (+2.0)*	83.1 (+1.5)*	68.7 (+5.8)*	65.3 (+4.1)*	66.8 (+4.9)*	75.0 (+3.2)*	77.5 (+2.7)*
uni+shifter	80.0 (+0.1)	83.5 (+0.0)	81.7 (+0.1)	62.9 (+0.0)	61.4 (+0.3)	62.1 (+0.2)	71.9 (+0.1)	74.8 (+0.1)
uni+pattern	81.3 (+1.4)*	83.2 (-0.4)	82.2 (+0.6)*	66.9 (+4.0)*	64.1 (+2.9)*	65.3 (+3.4)*	73.7 (+2.0)*	76.3 (+1.5)*
uni+l+p	81.6 (+1.7)*	84.0 (+0.5)	82.7 (+1.1)*	68.0 (+5.1)*	65.4 (+4.2)*	66.5 (+4.6)*	74.6 (+2.9)*	77.1 (+2.4)*
uni+l+r	84.3 (+4.5)*	87.6 (+4.1)*	85.9 (+4.3)*	72.4 (+9.5)*	69.9 (+8.7)*	71.0 (+9.1)*	78.5 (+6.7)*	80.7 (+6.0)*
uni+l+ctxt	82.9 (+3.1)*	86.1 (+2.5)*	84.4 (+2.8)*	71.4 (+8.6)*	68.4 (+7.2)*	69.7 (+7.8)*	77.1 (+5.3)*	79.3 (+4.6)*
uni+l+r+ctxt	84.4 (+4.5)*	87.8 (+4.3)*	86.0 (+4.4)*	72.9 (+10.0)*	70.9 (+9.7)*	71.8 (+9.9)*	78.9 (+7.1)*	81.1 (+6.3)*

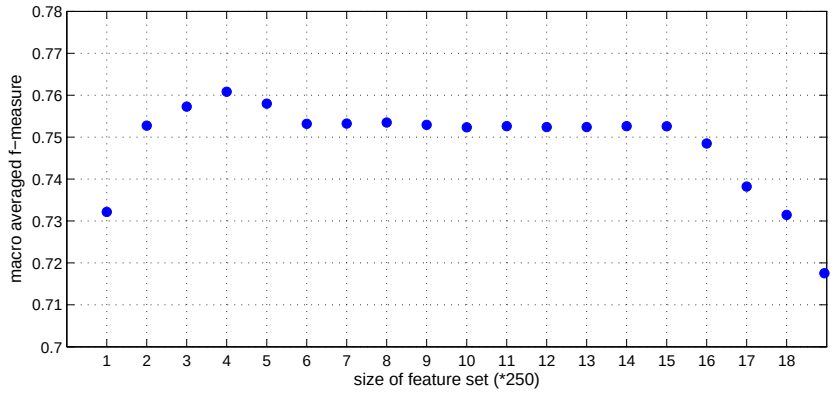
Table 10.6.: Hotel corpus: Effectiveness of different feature types for ternary polarity classification. The table shows the precision, recall, and f1-scores for the positive and negative class. Macro and micro-averages are computed with regard to these two classes. Numbers in parentheses show the difference to the unigram baseline and a star symbol indicates a statistically significant deviance (99% confidence level).

	positive			negative			average	
feature	**precision**	**recall**	**f1**	**precision**	**recall**	**f1**	**macro-f1**	**micro-f1**
unigram	75.2	78.5	76.8	58.7	54.7	56.5	66.6	70.0
ngram	77.4 (+2.1)*	79.7 (+1.2)*	78.5 (+1.7)*	62.2 (+3.5)*	53.7 (-1.0)	57.5 (+1.0)*	68.0 (+1.3)*	71.5 (+1.6)*
lexicon	68.1 (-7.1)*	78.5 (-0.0)	72.9 (-3.9)*	57.3 (-1.4)*	39.1 (-15.6)*	46.2 (-10.3)*	59.6 (-7.1)*	65.0 (-4.9)*
uni+lexicon	79.2 (+4.0)*	80.4 (+1.8)*	79.7 (+2.9)*	67.5 (+8.8)*	60.3 (+5.6)*	63.6 (+7.1)*	71.6 (+5.0)*	74.3 (+4.3)*
uni+rating	77.6 (+2.4)*	79.3 (+0.8)*	78.4 (+1.6)*	66.0 (+7.3)*	58.5 (+3.8)*	61.8 (+5.3)*	70.1 (+3.5)*	72.9 (+2.9)*
uni+ctxt	76.8 (+1.6)*	78.7 (+0.2)	77.7 (+0.9)*	62.3 (+3.5)*	52.9 (-1.8)*	57.1 (+0.6)	67.4 (+0.7)*	70.9 (+0.9)*
uni+goldctxt	78.2 (+3.0)*	80.0 (+1.4)*	79.0 (+2.3)*	65.2 (+6.4)*	56.9 (+2.2)*	60.6 (+4.1)*	69.8 (+3.2)*	72.9 (+2.9)*
uni+shifter	75.1 (-0.1)	78.5 (-0.0)	76.7 (-0.1)	58.7 (-0.0)	54.7 (-0.0)	56.5 (-0.0)	66.6 (-0.1)	69.9 (-0.1)
uni+pattern	78.3 (+3.1)*	77.8 (-0.8)*	78.0 (+1.2)*	63.1 (+4.4)*	56.4 (+1.7)*	59.4 (+2.9)*	68.7 (+2.1)*	71.7 (+1.7)*
uni+l+p	79.0 (+3.8)*	77.9 (-0.6)*	78.4 (+1.6)*	64.7 (+6.0)*	59.0 (+4.3)*	61.6 (+5.1)*	70.0 (+3.3)*	72.7 (+2.7)*
uni+l+r	81.0 (+5.8)*	82.0 (+3.4)*	81.4 (+4.6)*	70.3 (+11.5)*	65.7 (+11.0)*	67.7 (+11.2)*	74.6 (+7.9)*	76.8 (+6.8)*
uni+l+ctxt	79.8 (+4.6)*	81.1 (+2.6)*	80.4 (+3.6)*	67.9 (+9.2)*	61.5 (+6.9)*	64.4 (+7.9)*	72.4 (+5.8)*	75.0 (+5.1)*
uni+l+r+ctxt	81.1 (+5.9)*	81.9 (+3.4)*	81.5 (+4.7)*	70.3 (+11.6)*	65.5 (+10.9)*	67.7 (+11.2)*	74.6 (+7.9)*	76.8 (+6.8)*

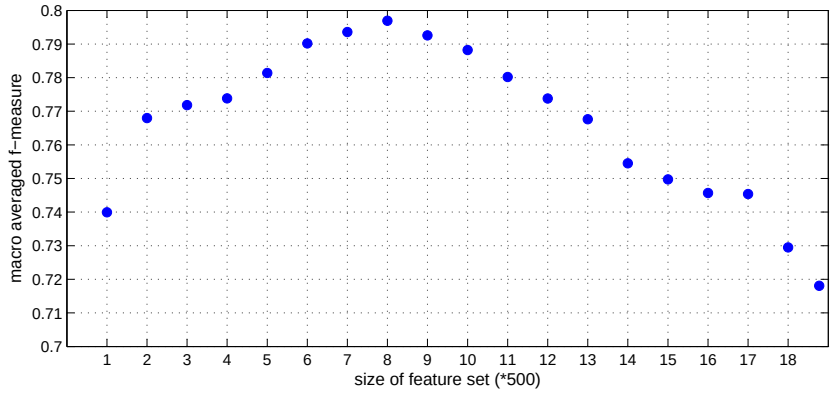
Table 10.7.: Camera corpus: Effectiveness of different feature types for ternary polarity classification.

"uni+l+r"), we find that the conveyed information complements each other. The increase in micro-averaged f-measure with the combination is nearly the sum of the individual improvements. For both datasets we observe a micro-averaged f-measure that is at least six percentage points above the unigram baseline. This additional increase is statistically significant compared to the individual results (not shown in the table). We also experimented with using the lexicon feature alone — that is, without adding unigram features. However, the results are significantly worse, especially for the digital camera review corpus. The recall for the negative class drops by nearly 16 percentage points.

Our results of using n-gram instead of the unigram features are at a first sight not conclusive. Whereas we measure a slight improvement for the camera dataset (+1.6 percentage points), the performance regarding the hotel corpus slightly deteriorates by 0.4 percentage points (not statistically significant). As we have noted earlier, the benefit of higher order n-grams comes into play if enough training data is available. The size of our manually labeled corpora is apparently too small for observing consistent, beneficial effects. To underpin this hypothesis and to outline the potential of higher order n-grams, we conducted experiments with including a feature selection step. Prior to training a classification model, we select the k most significant unigram or n-gram features. We use the χ^2-



(a) unigram features



(b) n-gram features

Figure 10.8.: Hotel corpus: The effects of feature selection for the unigram and n-gram feature types. The results show the macro-averaged f-measure (positive and negative class) for varying feature set sizes. Important: The reported results overestimate the true performance, as the feature selection procedure has access to the test folds.

test to calculate the most discriminating features with regard to the three different class labels (see also Manning et al. [250, chap. 13.5]). A global ranking is obtained by computing the mean of the per-class-scores for each feature. We only report results for the hotel corpus (observations are similar for the camera dataset). Figs. 10.8a and 10.8b summarize our experiments with feature selection. It is important to note that we did not perform feature selection within the cross validation loop, but obtained the feature rankings by considering the complete evaluation corpus. In consequence, the reported results overestimate the true performance. However, the experiments solely serve for showing the potential of higher order n-gram features. Comparing both figures, we see that choosing the optimal set of n-gram features (approx. the top-4000 features) leads to a macro-averaged f-measure that is 4 percentage points higher than the performance with the optimal set of unigram features (approx. the top-1000 features). We will also see in subsequent experiments, where huge amounts of weakly labeled data are available, that the use of higher order n-grams leads to improved results compared to an unigram model.

To estimate the utility of the context feature, we introduce a synthetic feature "gold context". This feature type has access to the class labels as defined in the gold standard. For each target sentence, the feature encodes the true labels of the previous and following sentence. It thus serves as an upper bound for the "real" context feature which does not have gold standard access, but calculates polarity information based on a sentiment lexicon. We can observe that including context information is generally helpful. Using the gold context feature, we find significant improvements in micro-

1	great	2	nice
3	very	4	good
5	staff	6	helpful
7	location	8	clean
9	excellent	10	friendly
11	was great	12	comfortable
13	beautiful	14	clean and
15	not	16	friendly and
17	were very	18	stayed
19	only	20	great location

(a) hotel dataset

1	great	2	easy
3	easy to	4	easy to use
5	is	6	very
7	love	8	is great
9	amazing	10	excellent
11	good	12	to use
13	not	14	is easy
15	great camera	16	great pictures
17	only	18	quality
19	is easy to	20	takes great

(b) digital camera dataset

Table 10.8.: The top-20 n-gram features according to a feature ranking based on the χ^2-test. Apparently, aspect mentions are also good indicators (presumably, to separate objective from subjective sentences).

averaged f-measure for both evaluation corpora (hotel/camera: +2.7/+2.9 percentage points). For the "real" context feature, we also obtain improved results (hotel/camera: +1.5/+0.9 percentage points in micro-average). For both datasets the increase is statistically significant. We also consider the combination of the lexicon and context features (setting "uni+l+ctxt"). As both rely on the same information source (the sentiment lexicon), we are interested in finding out whether both features complement each other. Our results show that the combination of both features further increases the micro and macro-averaged f-measure. Compared to the "uni+lexicon" setting, the increase of 1.3 (hotel) and 0.8 (camera) percentage points in micro-f1 is statistically significant. When testing the context feature in conjunction with the review rating and lexicon feature (setting "uni+l+r+ctxt"), we observe no significant increase compared to the "uni+l+r" setting for the camera corpus. For the hotel dataset, a very small, but statistically significant increase is measurable (+0.3 percentage points). In summary, the results show that there is truly potential for incorporating information from the context. Even when relying on inaccurate sentiment lexicon information, the performance improves consistently. However, if review rating information is available, additional benefits from the context feature are marginal.

Adding information about the occurrence of sentiment shifters (setting "uni+shifter") has only minimal effects on the micro/macro-averaged classification performance (at maximum 0.1 percentage points). For both datasets the difference is not statistically significant. We conclude that the sentiment shifter features do not help for our polarity classification task.

Adding the pattern features to the unigram baseline significantly increases performance. We observe improvements in micro-averaged f1-scores of 1.5 (hotel) and 1.7 (camera) percentage points. Nonetheless, the more complex pattern features do not seem to add complementary information compared to using the sentiment lexicon features. Combining lexicon and pattern features shows even worse results compared to using only lexicon features. Classification performance (micro f1) for the setting "uni+l+p" is 1.0 (hotel) and 1.6 (camera) percentage points lower than with the "uni+lexicon" setting (difference is statistically significant; not shown in the table). Thus, although we obtain improved results with the pattern feature, we conclude that is of no additional value. The lexicon features are apparently a superior way of encoding the sentiment lexicon information.

General Results and Mistake Analysis

In the following we discuss some more general results and consider the most frequent types of mistakes. If not otherwise stated, all subsequent comments refer to the "unigram+lexicon" setting. For both datasets we observe that results for the positive class are significantly better than for the negative

	positive	negative	objective
positive	—	47.8% (47.9%)	52.2% (52.1%)
negative	35.3% (62.3%)	—	64.7% (37.7%)
objective	41.4% (64.3%)	58.6% (35.7%)	—

(a) unigram (hotel)

	positive	negative	objective
positive	—	42.1% (47.9%)	57.9% (52.1%)
negative	37.8% (62.3%)	—	62.2% (37.7%)
objective	45.8% (64.3%)	54.2% (35.7%)	—

(b) unigram + lexicon + rating (hotel)

	positive	negative	objective
positive	—	37.4% (37.2%)	62.6% (62.8%)
negative	32.2% (52.3%)	—	67.8% (47.7%)
objective	49.2% (65.0%)	50.8% (35.0%)	—

(c) unigram (camera)

	positive	negative	objective
positive	—	31.8% (37.2%)	68.2% (62.8%)
negative	35.9% (52.3%)	—	64.1% (47.7%)
objective	46.4% (65.0%)	53.6% (35.0%)	—

(d) unigram + lexicon + rating (camera)

Table 10.9.: The misclassification rate by class label. For the different target classes, each row of the table shows the observed proportion of false positives with regard to the two control classes. The numbers in parentheses represent the true sample distribution with respect to the two control classes.

class. The difference is more than 10 percentage points in f-measure. In related work (e.g., [40, 422] such a difference is not observed. We first hypothesized that inferior results for the negative class may stem from the imbalanced datasets. The number of positive samples is about twice as large as the number of samples for the negative class. To test this hypothesis, we artificially balanced the evaluation corpora by randomly removing samples for the positive class. While the difference is slightly decreased for these artificial datasets, it is still over 8 percentage points for the hotel corpus and 6 percentage points for the camera corpus. Imbalanced data thus only partially explains our observation.

We therefore take a closer look at the actual misclassifications. Table 10.9 summarizes the misclassifications for different settings on a per class basis. For each target class, we report the percentage of false predictions with regard to the two control classes. For example, considering the positive class, the table shows that 47.8% of the false positives actually belong to the negative class and 52.2% belong to the objective class (hotel, unigram setting). The numbers in parentheses represent the false positive distribution that we would expect, based on the true distribution of the control classes' labels within the corpus. We find that, for the positive class, the observed distribution of false positive nearly perfectly fits the expected distribution in both corpora. In other words, errors in predicting the positive class are neither biased towards the negative, nor towards the objective class. For the negative class the picture is much different. The classifier has significantly more problems in distinguishing negative from objective samples than distinguishing negative from positive samples. Whereas we would expect that about 37.7% (hotel) and 47.7% (camera) of the errors would stem from falsely predicting the objective class, the observed numbers are around 65% in both corpora. Thus, a further reason for the lower performance concerning the negative class is explained by the observation that the distinction between the negative and objective class is more difficult than the distinction between the positive and objective class. We know that adding lexicon features to the unigram model raises the overall performance for both classes. In this context Tables 10.9b and 10.9d indicate the following: While the improvement for the positive class predominantly stems from a better distinction between both polar classes, the improvement for the negative class is mostly due to a better separation of the objective class.

To further analyze the reasons for misclassification, we look at the actual samples that led to false predictions. We especially consider errors in distinguishing the objective from the negative class. Our analysis shows that the classifier has difficulties in correctly classifying polar facts. Obviously, this distinction is harder as explicit clues are missing and to detect a polar fact, contextual knowledge is often necessary. For instance, consider the sentence "It took two hours to get a maid to clean the

room." No sentiment word is contained in the sentence. However, our (human) experience tells us that it is a sign of bad service if it takes two ours to find someone for cleaning a hotel room. We can easily interpret the sentence as negatively connoted. Now the problem with polar facts is as follows: Most polar sentences in the training data are not polar facts. These non-polar-fact samples contain explicit sentiment expressions and there exist a few expressions (e.g., "great", "nice", "helpful", "rude", "dirty") which can be found in a majority of polar sentences. A supervised classifier can easily learn such clue words or n-grams. Polar facts not only occur with lower frequency, but also the lexical variability of those token sequences that may serve as polarity indicators (e.g., "took two hours") is much higher. For instance, in this particular case, the sequence may also read "took three hours", "took four hours", or "took many hours". So, unless the classifier is provided with huge amounts of training data, it has difficulties in learning to correctly classify a great share of the polar facts. For the hotel dataset and the setting "unigram+lexicon", only 53% of the polar facts are correctly classified, whereas the correct rate for non-polar-fact samples is 78%. Regarding the digital camera dataset the numbers are 51% vs. 75%. The difficulties with polar facts are also a reason why the classification performance for the negative class is worse than for the positive class. The proportion of polar facts in the negative class is much higher than in the positive class. In the hotel dataset the proportion of polar facts in the positive class is about 10% compared to 37% in the negative class. Concerning the camera corpus the numbers are 7% compared to 32% (see also Table 10.2).

Another observation is that the classification performance with regard to the hotel corpus is significantly better compared to the digital camera dataset. Whereas we can measure a micro-averaged f-measure of 81.1% for the hotel dataset, the best setting with the camera corpus achieves a value of 76.8%. For both target classes, the positive and negative category, we find inferior results. The main reason for the reduced classification performance is due to the higher proportion of objective sentences within the digital camera corpus. We have 37% objective sentences in the camera corpus and only 28% in the hotel corpus. As distinguishing polar sentences from objective sentences is generally more difficult than distinguishing the two polar classes (see also the following section), the error rate in the digital camera corpus is comparably higher.

10.4.4. One vs. Rest Classification against the Cascaded Approach

In this section, we compare results with two different strategies of tackling the ternary polarity classification problem. For the previous experiments we used the built-in one-vs.-rest strategy of the LIBLINEAR SVM classification package. We inherently built three binary SVM classification models: "positive vs. negative/objective", "negative vs. positive/objective", and "objective vs. positive/negative". The strategy then compares the predictions of the individual classifiers and chooses the class which classifies a test sample with the greatest margin. A cascaded approach to ternary polarity classification may seem more natural. We only build two classifiers. A first classifier for subjectivity detection separates objective from subjective sentences and a second classifier determines the polarity of subjective sentences. Compared to the one-vs.-rest strategy, this setting defines more natural "rest" classes. Mixing negative and objective or positive and objective samples, as done in the one-vs.-rest setting, seems unintuitive. The binary classifiers in the cascaded approach are provided with more clear-cut sample sets — subjective vs. polar and positive vs. negative. We thus hypothesized that a cascaded strategy allows to create more accurate classifiers. To implement the cascaded approach, we perform the following steps for each training fold within a cross validation instance: To train the subjectivity detector, we relabel all positive and negative samples in the fold as "polar". To learn the polarity classifier, we remove all objective sentences from the training fold. For testing, we first apply the subjectivity detector. Those samples which are classified as subjective become input for the polarity classifier (note that this input may already contain false positives, i.e., objective sentences). We recombine the predictions of the two classifiers to create the final prediction vector.

Table 10.10 summarizes the results of our comparison of both strategies. The main result is that

	positive			negative			average	
features	**precision**	**recall**	**F1**	**precision**	**recall**	**F1**	**macro-F1**	**micro-F1**
one-vs.-rest	84.5	87.4	85.9	72.4	70.2	71.2	78.5	80.7
cascaded	82.0 (-2.5)*	88.1 (+0.7)	84.9 (-1.0)	68.7 (-3.7)*	71.0 (+0.8)	69.7 (-1.5)	77.3 (-1.2)*	79.6 (-1.2)*

(a) hotel corpus

	positive			negative			average	
features	**precision**	**recall**	**F1**	**precision**	**recall**	**F1**	**macro-F1**	**micro-F1**
one-vs.-rest	80.8	81.6	81.2	69.7	66.1	67.8	74.5	76.6
cascaded	78.2 (-2.6)*	82.1 (+0.5)	80.1 (-1.1)	67.3 (-2.4)*	66.0 (-0.2)	66.5 (-1.3)	73.3 (-1.2)*	75.4 (-1.1)*

(b) digital camera corpus

Table 10.10.: Comparison of the one-vs.-rest and the cascaded strategy for ternary polarity classification.

our hypothesis is not confirmed. We consistently observe inferior results with the cascaded approach on both evaluation corpora. Compared to the original one-vs.-rest approach, the macro-averaged f-measure is 1.2 percentage points lower for both datasets. The difference is statistically significant. We also experimented with different settings for the soft margin parameter C of the linear SVM classifier. Using the cascaded approach, we can apply different values C_{subj} and C_{pol} for the two classifiers[24]. We hypothesized that reasonable parameters for the subjectivity and polarity classifiers may deviate strongly, so that setting them independently could improve the overall performance. However, it turned out that, even with parameter optimization, results for the cascaded approach are consistently worse than for the one-vs.-rest strategy.

The cascaded approach explicitly subdivides the ternary classification task into the two subtasks "subjectivity detection" and "polarity classification". Table 10.11 provides detailed results for the these two subtasks. The subjectivity detection results are based on relabeling the evaluation corpus. Results with polarity detection are obtained by evaluating on the polar subset of the corpus (i.e., ignoring the objective part). The table reports the f-measure values for the relevant classes in each task. The main observation is that separating objective from subjective sentences is generally more difficult than separating positive from negative sentences. The macro-averaged f-measure for subjectivity detection is around 10 percentage points lower in both corpora compared to the polarity classification task. The f1-score for the objective class is lower than 70% in both corpora. As mentioned earlier, in customer review mining, we are more interested in accurately identifying the subjective sentences. For this class we observe f-measure values of 86.9% (hotel) and 82.6% (camera). On the other hand, the f1-score obtained with the supervised subjectivity classifier is not much higher than the f1-score obtainable with a simple classifier that predicts each sample as subjective. For the hotel dataset the distribution of subjective vs. objective samples is 72% vs. 28%. Predicting each sample as subjective

[24] Note that this is also possible for the one-vs.-rest approach by tweaking the "weight" parameters of the LIBLINEAR SVM training module.

	subjectivity				polarity			
dataset	f1-polar	f1-objective	macro f1	micro f1	f1-positive	f1-negative	macro f1	micro f1
hotel	0.869	0.625	0.747	0.806	0.898	0.813	0.855	0.868
camera	0.840	0.722	0.781	0.797	0.900	0.811	0.856	0.870

Table 10.11.: Classification performance for subjectivity detection and binary polarity classification.

leads to a precision of 72% at a recall level of 100% (subjective class). The f1-score for such a classifier is thus 83.7%. For the camera dataset precision would be 62.8%, resulting in an f-measure of 77.1%. The random accuracy[25] for the datasets is 59.7% and 53.3%. The gain in overall accuracy is thus approximately 20 percentage points for the hotel dataset and roughly 25 percentage points for the camera corpus. In summary, comparing the supervised subjectivity classifier with a random classifier, we observe a huge gain in overall accuracy, but only a small gain in accuracy with respect to the target class ("subjective"). Random accuracy values for the polarity classifiers are 54.1% (hotel) and 54.5% (camera), so the gains are over 30 percentage points for both classes in this task.

10.4.5. Binary Polarity Classification with Weakly Labeled Data

In the following sections we evaluate the utility of weakly labeled data for polarity classification. We first consider the task of binary polarity classification — that is, the separation of positive and negative sentences. The weakly labeled data we extracted from the pros/cons summaries of customer review fits well for this task as we are provided with huge amounts of training data for both classes. In the following, we mainly experiment with two settings. We either use the weakly labeled data as a complement to the manually labeled data, or we rely on the weakly labeled data as the only source for training data (see also Section 10.3.4). In particular, we consider the following experimental setups:

- Weakly labeled data as complement: As with previous experiments, we use ten times repeated 10-fold cross validation (in fact we use the same ten partitionings as before). Let W be the set of weakly labeled data and M be the set of manually labeled data. For each of the ten training/test splits of a single cross validation instance, we proceed as follows: We construct a training set $TR_k = W \cup M_{k\text{-}train}$, where $M_{k\text{-}train} \subset M$ is the k-th training partition of M. We train a binary SVM on TR_k and evaluate the resulting classifier on the test fold $M_{k\text{-}test} \subset M$ (we thus ensure that only manually labeled samples serve as test data). This setup is depicted in Fig. 10.9. In addition, we vary both, the amount of weakly labeled samples as well as the amount of manually labeled instances that are available for training. The test folds are always the same, independent of the amounts of weakly or manually labeled training data. To vary the amount of manually labeled training data, we artificially reduce the corpus size by simply removing some samples. We do so in a stratified way to retain the original class label distribution.

- Only weakly labeled data: In this setting we train a classifier on weakly labeled data only. Again, we vary the amount of data available during training. For each such experiment, we train a single SVM classifier. This classifier is then applied to the same 10*10 test folds as described before. The final reported results are calculated as the mean of the results for the 100 different test folds. Statistical significance is based on a paired t-test that compares the results (means) for each of the 10 cross validation instances.

As feature types, we use 3-grams (including lower order n-grams) and the sentiment lexicon features. The review rating or context features cannot be applied in the settings with weakly labeled data. We do not perform any feature selection.

Figs. 10.10 and 10.11 summarize our results with both experimental setups. The figures are interpreted as follows: Different colors represent the varying sizes of manually labeled data available during training. For our experiments, we examined five different sizes. The first setting (size = 0) represents the setting where we only use weakly labeled data. In the last setting we use the complete training fold of manually labeled data $M_{k\text{-}train}$. Naturally, for this setting the size is dependent on the concrete corpus and thus varies between different experiments. The x-axis is log-scaled and

[25] For the binary classification task, accuracy equals the micro-averaged f-measure. Random accuracy refers to the accuracy obtained with a classifier that makes random predictions based on the distribution of class labels.

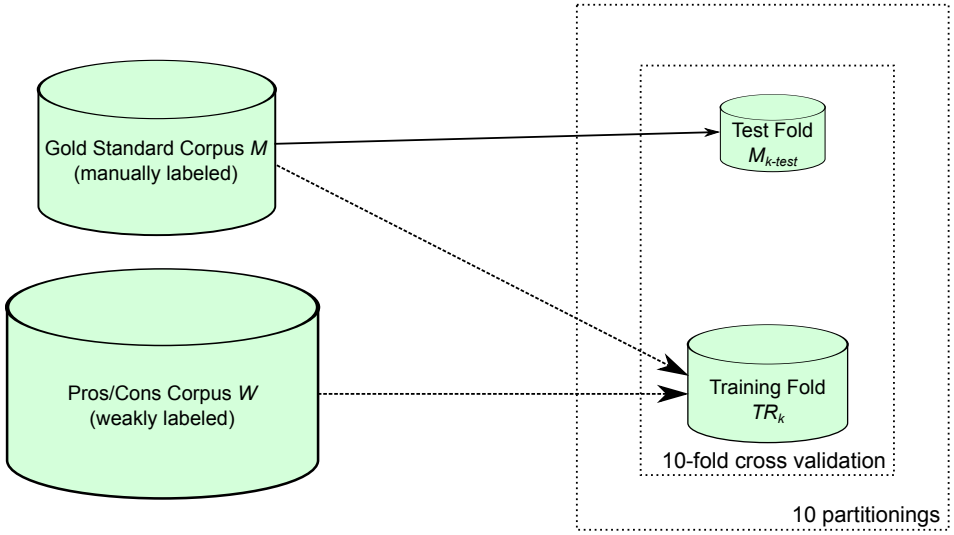


Figure 10.9.: The interplay of manually and weakly labeled data for polarity classification.

represents the amount of weakly labeled data used for training the classifiers. Take note that the maximum size of the weakly labeled training datasets vary for the two different domains (hotel: 200,000 sentences; camera: 100,000 sentences). Obviously, no measurement for *manually labeled size = weakly labeled size* = 0 is available. For each experiment, the y-axis reports the macro-averaged f-measure. Our main results are as follows:

Results — Amount of Manually Labeled Data

Considering all but the blue colored measurements (size = 0) at x = 0 (no weakly labeled data) gives us insight about the classification performance with different sizes of manually labeled data. Using only 100 manually labeled training instances, we achieve an already good macro-averaged f-measure of around 80% for the hotel dataset. However, for the camera dataset this particular setting performs much worse with only around 71% f-measure. Using the original corpora (i.e., without artificially reducing the size), we observe f-measures of 86.2% (hotel) and 83.9% (camera). Take note that these measures slightly deviate from the results in Table 10.11. This is due to the different feature types used in the experiments. As can be expected, by increasing the amount of manually labeled data, we can improve the overall classification performance. For both datasets, using only 1,000 samples, we can achieve a performance that is only 2 percentage points lower compared to using the complete corpus.

Results — Weakly Labeled Data as Complement

Again, all but the blue colored measurements are relevant for this analysis. The main result is that the weakly labeled data is indeed helpful for binary polarity classification at the sentence level. With increasing amount of weakly labeled data, the classification performance improves significantly. For the most conservative setting, when using the complete manually labeled corpus, we can increase the macro-averaged performance by 4 percentage points for the hotel dataset and by 3 percentage points for the camera review corpus. Both improvements are statistically significant. For the other settings with less manually labeled data, improvements are significantly higher. For example, when only 100 manually labeled samples are available, the increase in macro-averaged f-measure is more than 10 percentage points for the hotel dataset and over 13 percentage points for the digital camera corpus. With increasing amount of weakly labeled data, the size of the initially available manually

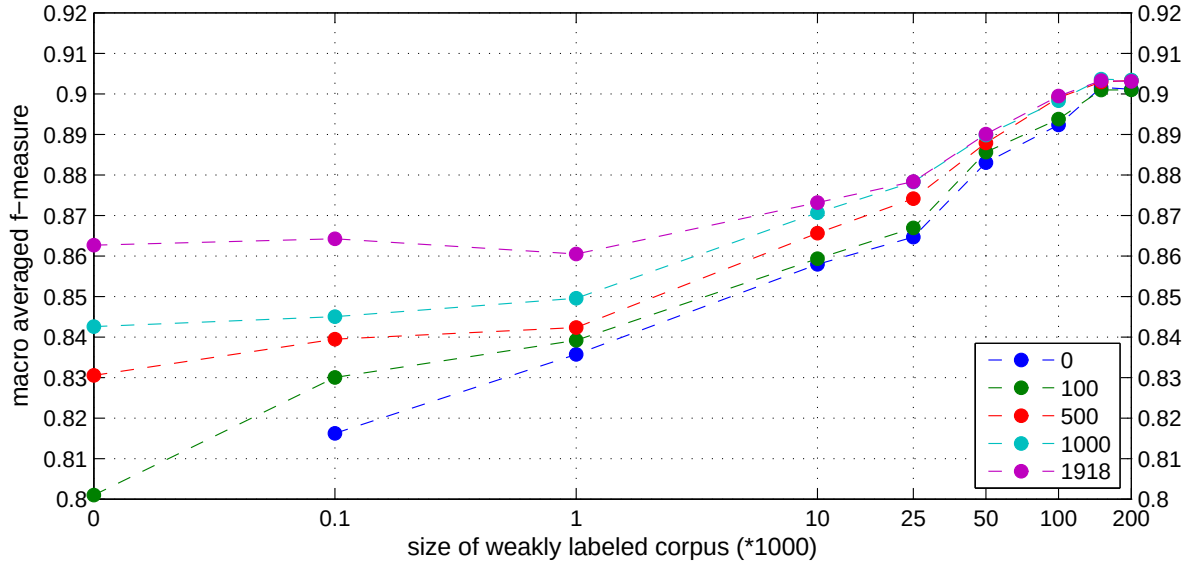


Figure 10.10.: Hotel review corpus: Results for binary polarity classification with varying amounts of weakly and manually labeled training data. The different colors represent the varying amounts of manually labeled data. Individual measurements are indicated by thick dots, the dashed lines are only for reasons of visualization.

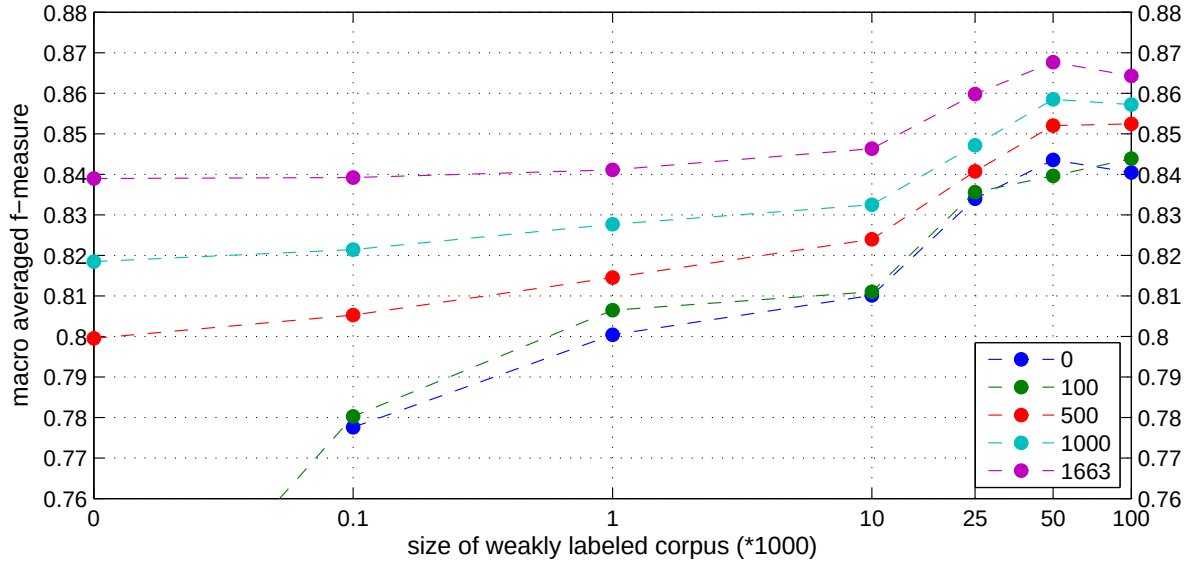


Figure 10.11.: Digital camera review corpus: Results for binary polarity classification with varying amounts of weakly and manually labeled training data.

labeled training data becomes less relevant. At least for the hotel review dataset, we can observe that classification performance for all settings is roughly equal at around 90% f-measure. This culminates in the following observation:

Results — Weakly Labeled Data Only

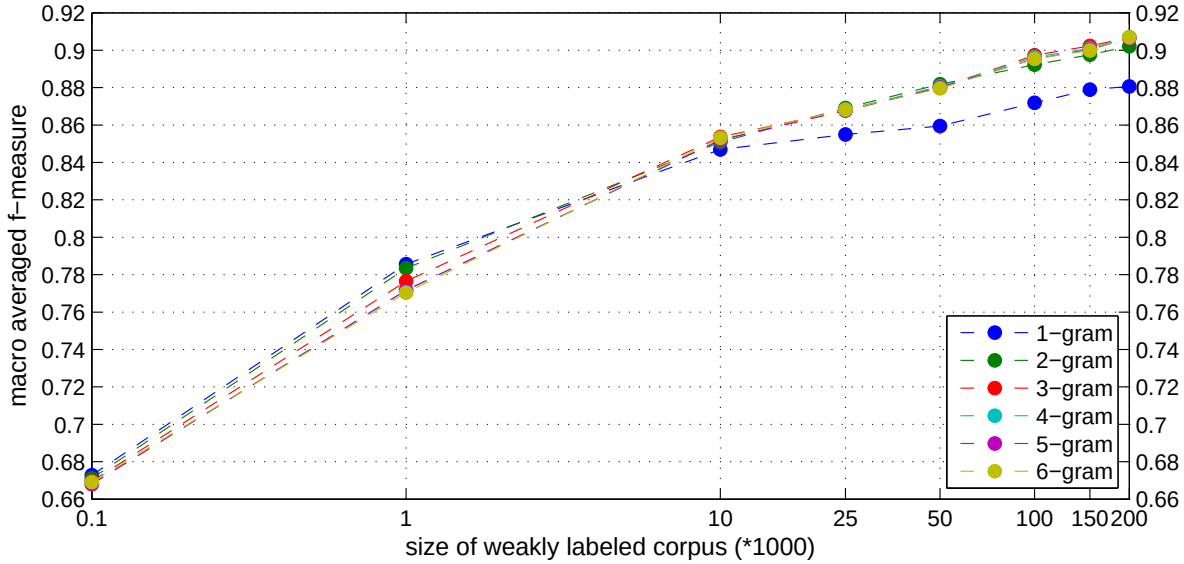
Given that enough weakly labeled data is available, we can completely go without manually labeled data and achieve the same results as if hand labeled data were available. For the hotel dataset, we achieve f-measures of slightly more than 90%, which is an improvement of 4 percentage points over the setting with only manually labeled data (complete corpus). With respect to the digital camera dataset, we cannot measure an improvement, but observe equally good results at around 84% f-measure. Here, the setting with only weakly labeled data is around 2.5 percentage points worse compared to the setting where all weakly and manually labeled data is used. For both datasets we also observe that classification performance with 100 weakly labeled samples is better than with 100 manually labeled sentences. This is astonishing at first sight, but is explained by the experimental setup. As we use stratified sampling to artificially reduce the amount of manually labeled data, the bias towards positively labeled samples is retained in the dataset. Too few negatively labeled sentences are available. On the other hand, the weakly labeled data contains an equal amount of positive and negative samples. With only 100 samples this setup affects classification performance. With regard to the hotel corpus we roughly need 25,000 weakly labeled samples to achieve the same performance as with a conventional, manually labeled corpus. For the digital camera review dataset, the break even is at around 50,000 samples. Considering that much more weakly labeled is available at roughly no costs, we can conclude that for binary polarity classification, weakly labeled data from pros/cons summaries is totally sufficient.

10.4.6. Varying N-gram Order

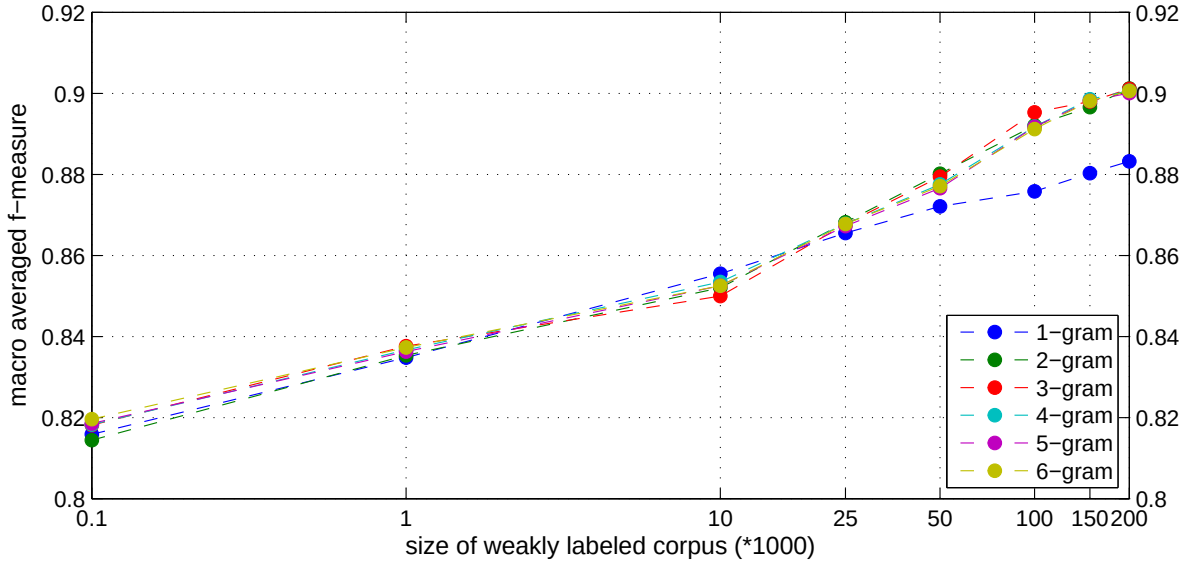
Earlier we pointed out that benefits with higher order n-gram features come into play if "sufficient" training data is available. With the following experiments we examine classification performance with varying n-gram order at varying amounts of training data. The experimental setup is similar to the previous one. We consider binary polarity classification. Training data is exclusively composed of weakly labeled samples. Evaluation is based on ten times repeated 10-fold cross validation with the same test folds as before. We further distinguish two different setups. In the first setup we only use n-gram features, in the second setup we combine n-gram and sentiment lexicon features. In our experiments n-gram features also include all lower order k-grams with $1 \leq k < n$. Fig. 10.12 depicts our results for the hotel corpus and Fig. 10.13 illustrates the results we obtained for the digital camera dataset. Again, the x-axis represents the amount of weakly labeled training data on a logarithmic scale. The differently colored dashed lines correspond to the maximum order of the n-gram features.

Our experiments show mainly three important results: First, the results confirm our hypothesis that higher order n-grams outperform simple unigram features if sufficient training data is available. For the hotel dataset, the best performance with higher order n-grams is 2.8 percentage points better compared to the setting with only unigrams. For the camera datasets the difference is even 3.3 percentage points. Both differences are statistically significant. The figures further show that for both, the hotel and the camera dataset, the break even point is at around 10,000 weakly labeled samples. With less amounts of training data, the unigram model exhibits equally good or even better classification performance. With increasing amounts of training data, we observe better results for the higher order n-gram features.

The second main result is that the difference in classification performance for different higher order n-gram models is rather marginal. For example, we do not observe much better results with 6-gram models compared to 3-gram models. On the other hand, we can measure a statistical significant

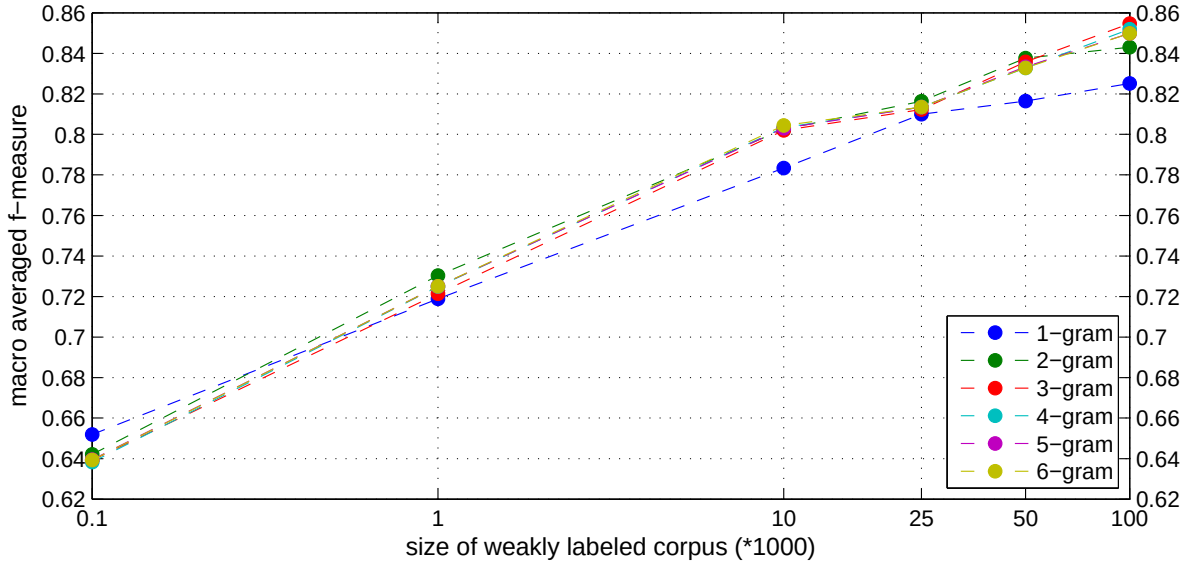


(a) excluding sentiment lexicon features

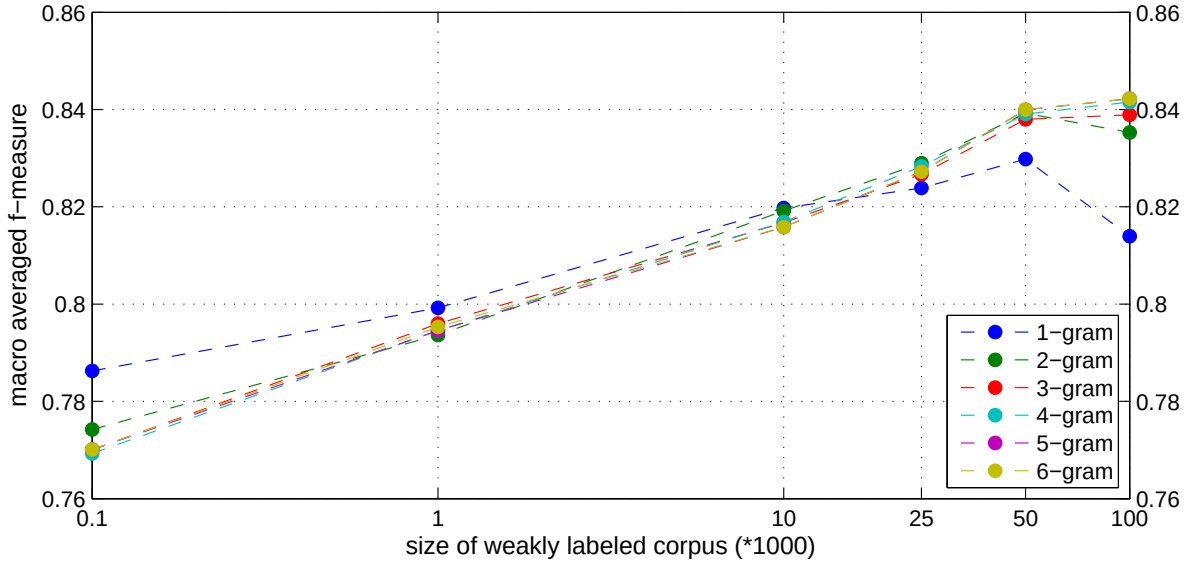


(b) including sentiment lexicon features

Figure 10.12.: Hotel corpus: Results for binary polarity classification with varying n-gram order and varying amount of weakly labeled training data.



(a) excluding sentiment lexicon features



(b) including sentiment lexicon features

Figure 10.13.: Digital camera corpus: Results for binary polarity classification with varying n-gram order and varying amount of weakly labeled training data.

deviance between bigram models and n-gram models with n > 2. For instance, for the camera corpus the difference between the bigram and trigram model is 1.7 percentage points. For the hotel corpus the difference is much lower (0.4 percentage points), but still significant. We may assume (without proof) that with even more training data (e.g., > 10^6 samples) we could also observe differences between n-gram models with n $\geq$ 3.

The third main results refers to the utility of sentiment lexicon features in the context of large amounts of training data. We examine the effects by comparing Figs. 10.12b and 10.13b with Figs. 10.12a and 10.13a. The main observation is that sentiment lexicon features can compensate for too small amounts of training data. For instance, if only 100 weakly labeled samples are available, using the lexicon features increases the classification performance by over 15 (hotel) and over 13 (camera) percentage points. Also with 1,000 samples, the difference is still relatively high with more than 5 (hotel) and 8 (camera) percentage points. With increasing amounts of training data, the availability of the lexicon features becomes irrelevant or even affects classification performance negatively. Given enough training data, the higher order n-gram models achieve better results without including the lexicon features. For instance, for the camera corpus the difference is 1.8 percentage points with the trigram model. Differences are lower with the hotel corpus. A further note: We find no easy explanation for the decreased classification performance from step 50,000 to step 100,000 for the unigram model in Fig. 10.13b. Such a behavior is not observed for the other, similar experiments.

10.4.7. Effectiveness of Data Cleansing Heuristics

With the following experiments we examine the effectiveness of the cleansing heuristics that we have introduced in Section 10.3.3. Recall that the heuristics allowed to raise the precision for the weak label decision to 97% (camera) and 95% (hotel), respectively. We are now interested in analyzing whether this *intrinsic* increase in performance has positive effects with regard to *extrinsic* evaluation. In other words, does classification performance improve with "cleaner" weakly labeled data compared to "dirty", unfiltered weakly labeled data? To examine the effectiveness, we simply switch off the cleaning heuristics and compare this "dirty" dataset with the filtered datasets (which was used in the previous experiments). In particular, the dirty dataset is based on the following setup:

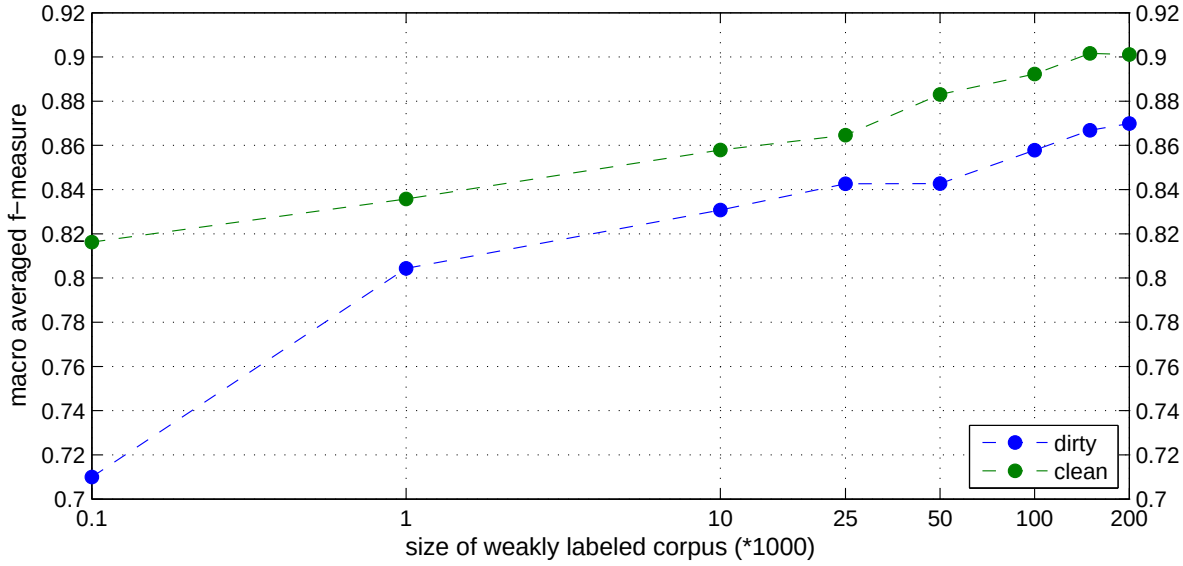
- Disabled heuristics: "absence of pros/cons", "missing verb", "contradiction", "sentiment lexicon"
- Enabled heuristics: "minimum length", "enumeration"

We use the same feature types as in Section 10.4.5, namely trigram and sentiment lexicon features.

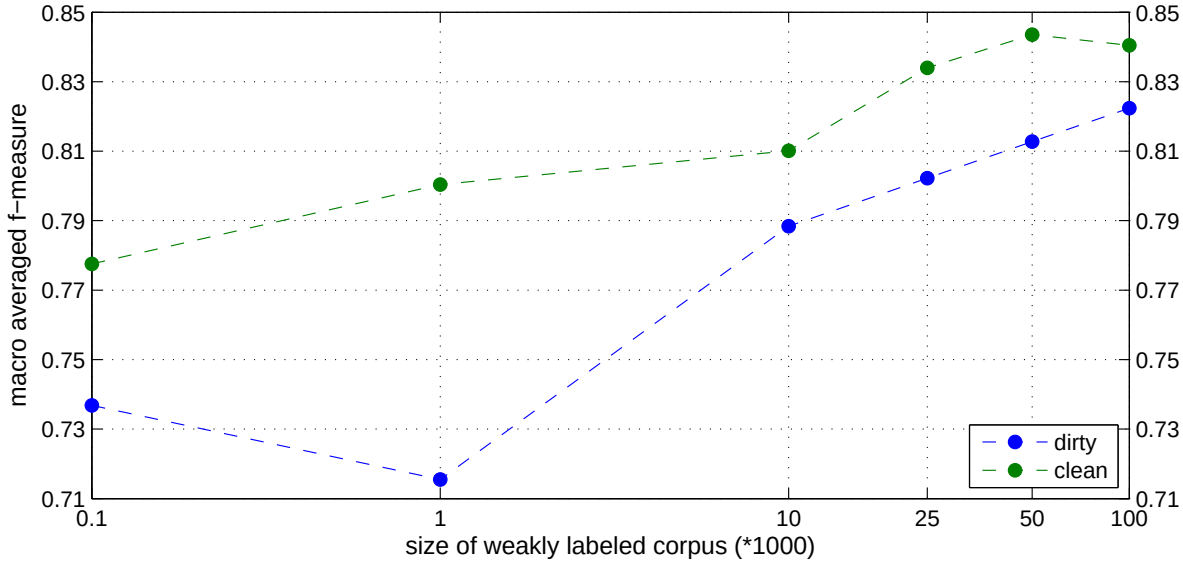
We summarize the results of our experiments in Fig. 10.14. The results obtained with this extrinsic evaluation underline our previous results with intrinsic evaluation. We observe that for both datasets and for varying amount of training data, classification performance is improved by the proposed filtering techniques. For the hotel dataset, we find increases in macro-averaged f-measure by around 3 percentage points. The numbers for the camera dataset are similar with improvements from 2-3 percentage points. Differences in classification performance seem to be more distinct with small amounts of training data. We conclude that data cleansing is an important step when working with weakly labeled data from pros/cons summaries and further that our proposed heuristics successfully improve the data quality.

10.4.8. Subjectivity Detection with Weakly Labeled Data

In the previous section we focused on binary polarity classification and disregarded the separation of subjective from objective sentences. We now examine the utility of our weakly labeled data for the task of subjectivity detection. In particular, we conduct the following two experiments:



(a) hotel review corpus



(b) digital camera review corpus

Figure 10.14.: Effectiveness of the cleansing heuristics for weakly labeled data.

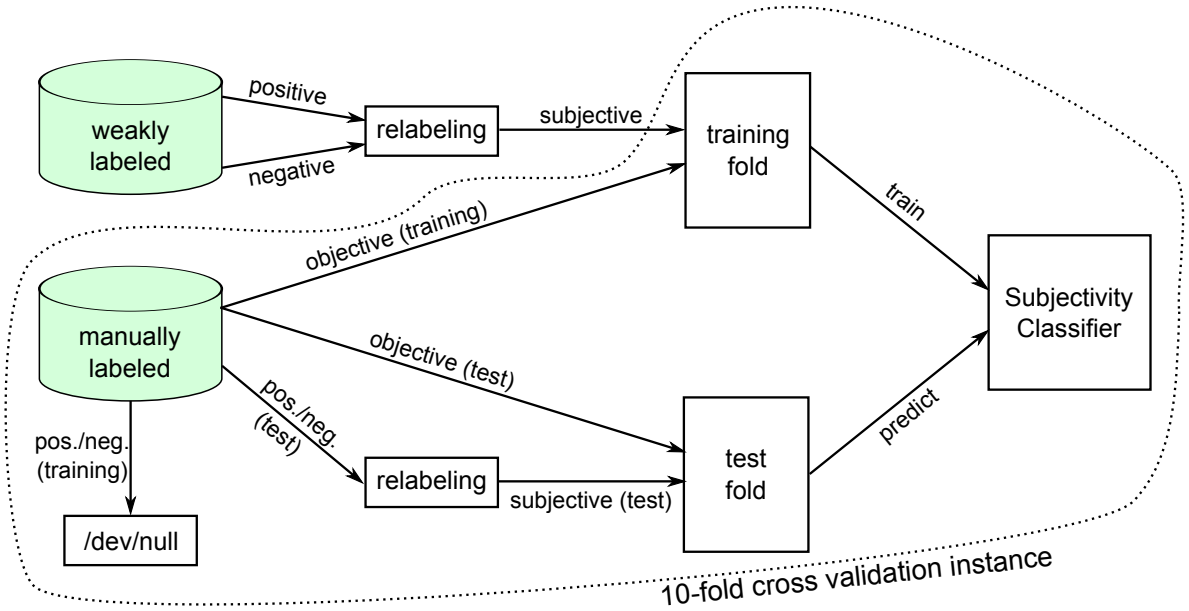


Figure 10.15.: The experimental setup for evaluating the performance of subjectivity classification with weakly labeled data. The arrows indicate which type of samples stem from which data source.

1. Subjectivity detection with manually labeled objective samples and weakly labeled subjective/polar samples: This setting corresponds to the second strategy as described in Section 10.3.4. We are interested in how well the weakly labeled data can compensate for the lacking polar samples from the gold standard.

2. Subjectivity detection with only weakly labeled data and one-class classification: This setting corresponds to the third strategy as described in Section 10.3.4. We learn a one-class subjectivity classifier from the weakly labeled, polar data. We compare our results with this approach to the results from the first experiment and, of course, to the performance of a subjectivity classifier that is trained on the complete gold standard corpus. For further comparison, we evaluate an unsupervised, lexicon-based subjectivity classifier.

The following subsections describe each experiment in more detail and discuss the obtained results.

Subjectivity Detection with Manually and Weakly Labeled Data

The concrete experimental setup is as follows: Again, we conduct ten times repeated 10-fold cross validation. The samples used in the 100 test folds are the same as in all previous experiments. For the subjectivity detection task, we relabel all samples from the positive and negative classes as belonging to the single class "subjective". We do so for the samples of the gold standard as well as for the samples in the weakly labeled corpus. For each fold of each cross validation instance, we train a single subjectivity classifier. Samples for the objective class are taken from the corresponding training fold of the gold standard corpus. Subjective training samples exclusively stem from the weakly labeled data. In fact, the weakly labeled subjective samples are the same for each fold. (We can do so, as the test folds contain only data from the manually labeled gold standard corpus.) Fig. 10.15 illustrates this experimental setup.

Figs. 10.16 and 10.17 present the results of our experiments. As evaluation metrics, we consider the f-measure for the target class "subjective" and the classifier's total accuracy (which equals the micro-averaged f-measure in this case). For comparison, we indicate the accuracy values for two other

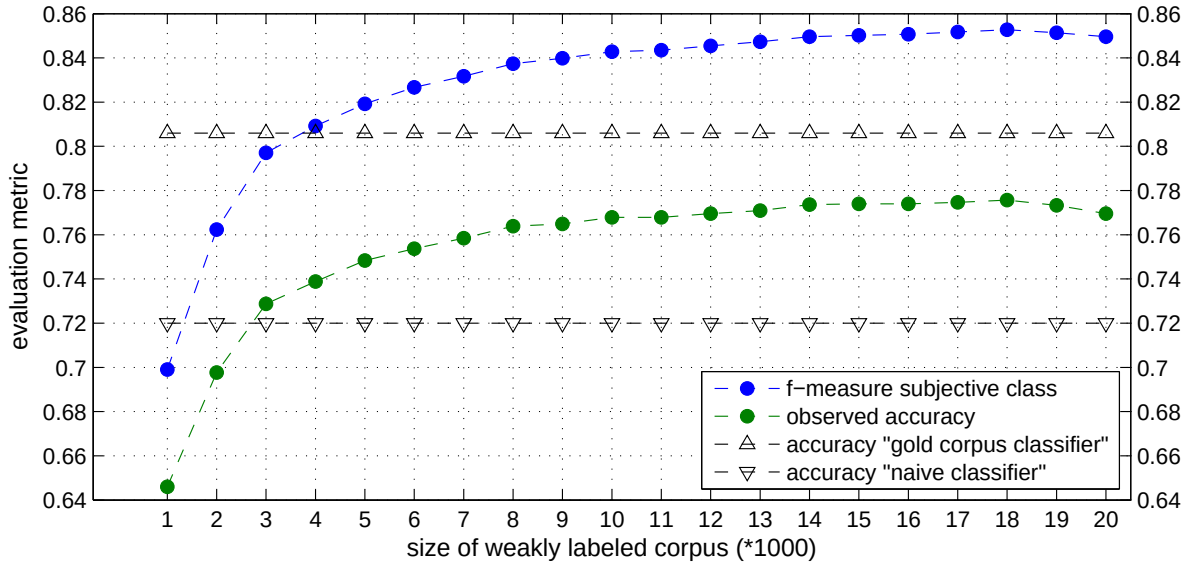


Figure 10.16.: Hotel corpus: Subjectivity detection with weakly labeled data.

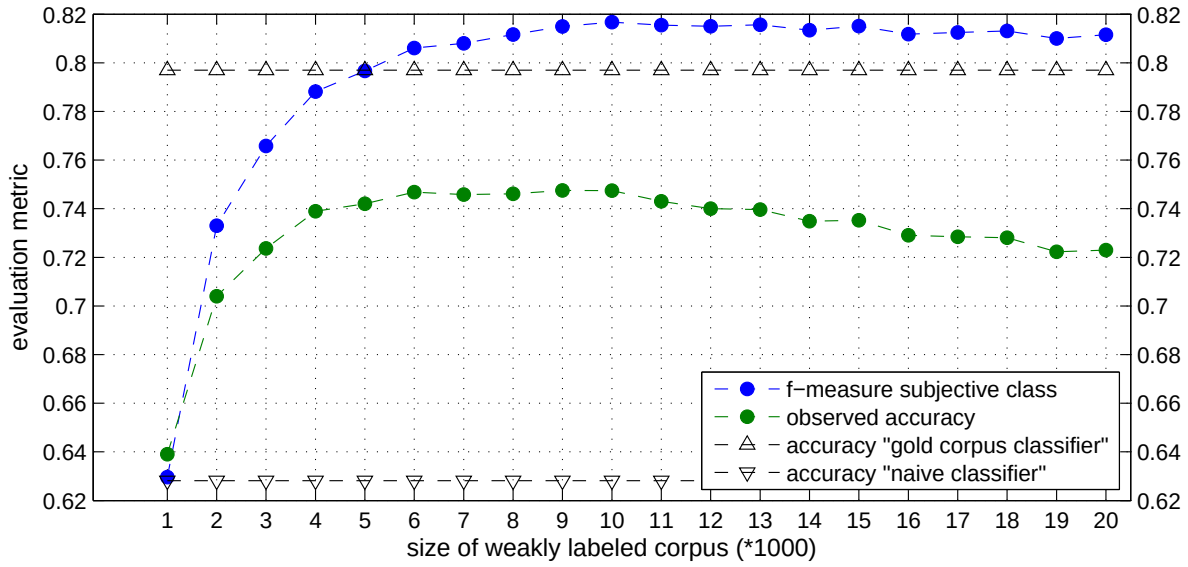


Figure 10.17.: Digital camera corpus: Subjectivity detection with weakly labeled data.

classifiers. The measured accuracy for the original "gold corpus classifier" that was trained exclusively on manually labeled data serves as an upper reference (see Table 10.11). The "naive classifier" that predicts each sample as belonging to the majority class "subjective" serves as a lower bound[26]. Considering the f-measure for the target class, we observe values of 85.3% for the hotel dataset and 81.7% for the digital camera corpus. These numbers are only around 2 percentage points lower than the results we achieved with the gold corpus classifier (not shown in the figure). With respect to the overall accuracy, we find maximum values of 77.6% (hotel) and 74.8% (camera). Here, the difference to the results with gold corpus classifier (indicated by the triangle markers) is more pronounced. For the hotel corpus, the accuracy is 3.0 percentage points lower, for the camera corpus the deviance is 4.9 percentage points.

Looking at the curve shapes of the evaluation metrics, we find that the classification performance first increases with larger amounts of weakly labeled training data and then remains on the same level. Our figures only show results for sizes in the interval from 1,000 to 20,000. With even larger amounts of weakly labeled data, we obtain worse results and the curves actually fall. The reason for the negative effects of more and more weakly labeled data is due to the increasing imbalance of the training data. Recall that by increasing the amount of weakly labeled data, we only increase the amount of samples available for the subjective class. As samples for the objective class stem exclusively from the manually labeled corpus, this part of the training set always remains the same, independent of the the amounts of weakly labeled data. The imbalance in the training set shifts more and more to the subjective class and thus more and more deviates from the sample distribution in the test set. The negative effects may be compensated by tuning the weight parameters offered by the LIBLINEAR SVM training module, but we did not extend our experiments in this direction.

We summarize and conclude as follows: Weakly labeled data from pros/cons summaries can also be used for subjectivity detection if additional, manually labeled samples for the objective class are provided. Due to the nature of the data, it does not fit as well as for binary polarity detection. Classification performance with the weakly labeled data is lower than using only manually labeled data, but is comparable and reasonably high. The utility of larger amounts of weakly labeled data is restricted by problems with increasing imbalance in the training data. Future work may consider more sophisticated tuning of the SVM classifiers to compensate for this problem.

Subjectivity Detection with Weakly Labeled Data and One-Class Classification

The concrete setup for our experiments in this section is as follows: We train a single one-class classifier by means of the weakly labeled data. To do so, we relabel positive and negative samples as belonging to the single class "subjective". This data is input to the one-class training algorithm. As features, we use trigrams and the sentiment lexicon scores. Since we do not use any of the data from the gold standard corpus for training, we can go without any cross validation. We simply apply the learned classifier on the complete evaluation corpus (after relabeling the positive and negative class) and report results for this single test set. With regard to the one-class training algorithm, a single parameter ν needs to be defined. We set $\nu = 0.1$ for both datasets. For reasons of comparison, we also consider the use of a simple, unsupervised, lexicon-based classifier. We use Algorithm 10.2 to compute sentiment scores for the individual words/phrases in a sample sentence. We sum up the absolute scores to obtain a single score for the whole sentence. If this absolute sentence score is greater than a threshold value $\tau_{\textit{sentiment-score}}$, we predict the sentence as being subjective. Otherwise it is assumed to be objective. In our experiments, we set $\tau_{\textit{sentiment-score}} = 0$. In other words, each sentence that contains a sentiment word is predicted as being subjective.

Figs. 10.18 and 10.19 present the results of our experiments. As before, we report the f-measure for the target class "subjective" and the total accuracy when both classes are considered. Dots represent

[26] Although we indicate different measurements over varying amount of weakly labeled data, the reference values are based on a single measurement (or calculation, respectively) and are independent of the actual corpus size.

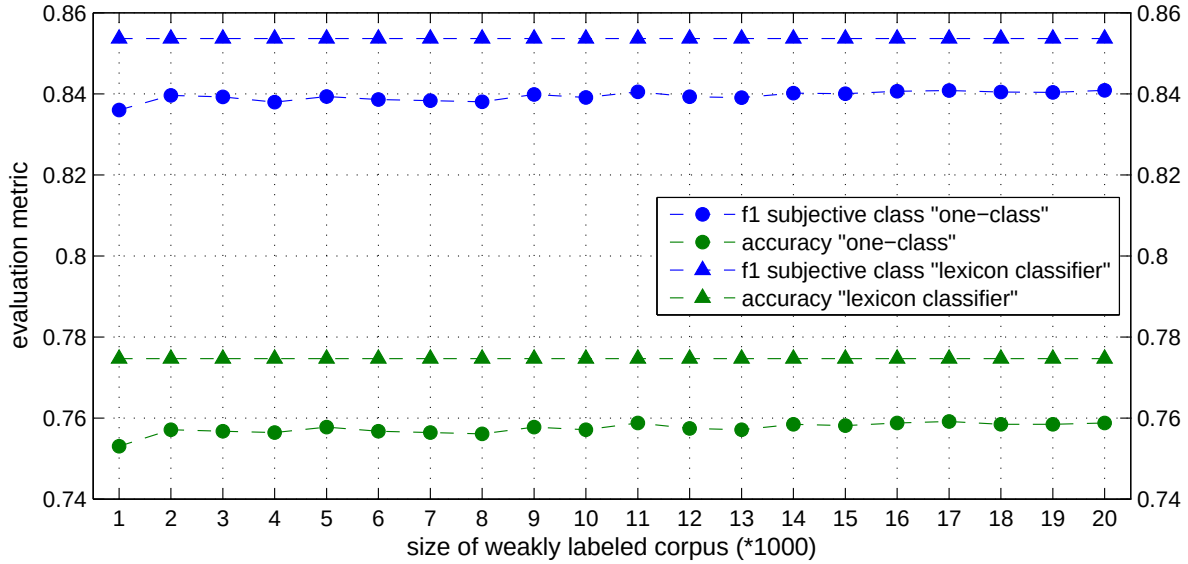


Figure 10.18.: Hotel corpus: Subjectivity classification with a one-class classifier vs. a lexicon-based classifier.

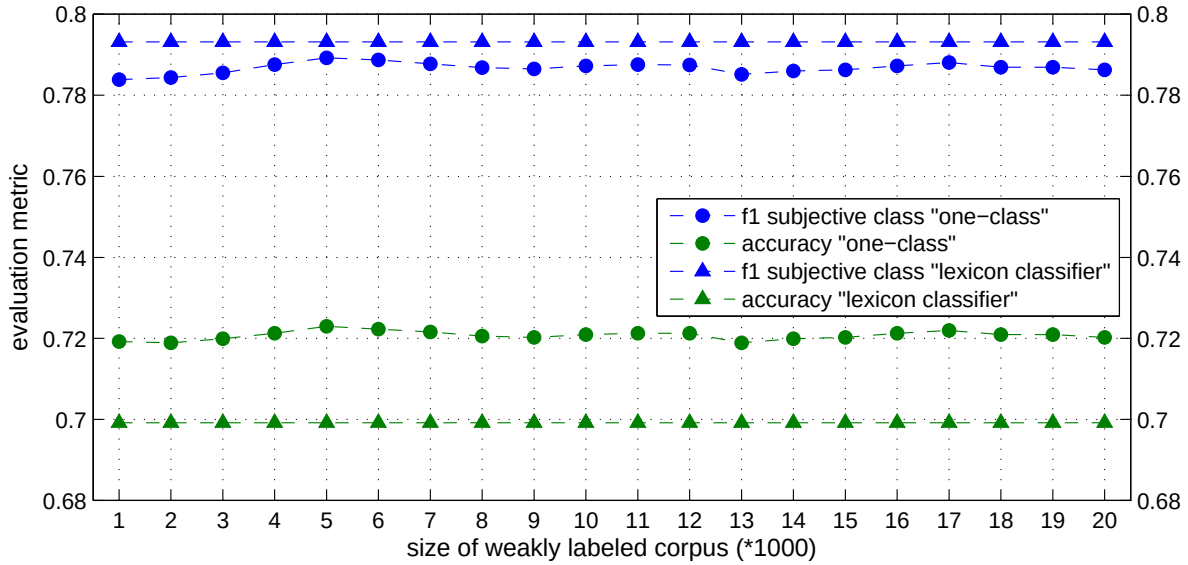


Figure 10.19.: Digital review corpus: Subjectivity classification with a one-class classifier vs. a lexicon-based classifier.

the measurements with the one-class classification algorithm for varying amounts of weakly labeled data. Triangles represent results with the lexicon-based classifier. Considering the classification performance with regard to the target class, the main result is that one-class classification is worse than any other reference we have tested before. Also the simple lexicon-based classifier achieves superior results. Further, the observed f-measure for the subjective class is only marginally higher than the f-measure of the hypothetical "naive" classifier. Considering the total accuracy, the picture is slightly better. For the camera dataset, total accuracy with the one-class classifier is about 2 percentage points higher compared to the lexicon-based classifier. Nonetheless, the absolute values of around 76% (hotel) and 72% (camera) are significantly lower than the results obtained with the gold corpus classifier or the results presented in the previous subsection. Varying the amount of available (weakly labeled) training data does not seem to have large effects on the classification performance. We do not observe significant deviations for both datasets.

To analyze whether the inferior performance with one-class classification is due the data being *weakly* labeled, we conducted a similar experiment using the manually labeled data only. In particular, we perform a 10-fold cross validation, where we train a one-class classifier on the manually labeled, subjective samples. Our results are even worse with this setup. Thus, the reason for inferior results is rather due to the inferiority of the approach than due to the different data quality.

Based on the results, we conclude as follows: Our approach with one-class classification to overcome the problem with missing (weakly labeled) training data for the objective class does not adequately compensate the lack. Regarding the target class, we obtain better results with a simple, unsupervised, lexicon-based classifier. In consequence, if the goal is to reduce the effort entailed with creating labeled datasets for subjectivity detection, using a lexicon-based classifier is a better choice.

10.5. Summary and Conclusions

In this chapter, we provided a detailed analysis of supervised methods for the task of sentence level polarity classification in customer reviews. Besides discussing (and experimenting with) varying feature sets and machine learning configurations, we were particularly interested in the utility of weakly labeled data for this task. More precisely, we proposed to exploit pros/cons summaries as training data for a supervised polarity classifier. Our goal was to answer the question whether weakly labeled data can be used to reduce or even eliminate the need for manually labeled training data. Our experiments show that for the task of binary polarity classification (i.e., positive vs. negative), classifiers trained on huge amounts of weakly labeled data achieve even better results than classifiers that had access to manually labeled data. For this task, we can thus completely go without the effort of manually creating training data. For the task of supervised subjectivity detection (subjective vs. objective), we can partly substitute manually labeled data with the weakly labeled pros/cons data. In particular, we can use the weakly labeled data as samples for the subjective class, however still rely on manually labeled data for the objective class. We were unable to find an appropriate source for extracting weakly labeled samples for the objective class. To overcome this lack of data, we experimented with one-class classification techniques. Unfortunately, we could not create adequate subjectivity classifiers with such an approach. In fact, we achieved better results for this task with simple, unsupervised, lexicon-based classifiers. In the following we summarize our main insights and contributions of this chapter in more detail:

In the introductory part of this chapter (Section 10.1), we provided a general overview of supervised techniques for sentiment polarity classification. Whereas being one of the most well studied subtasks in sentiment analysis, no single, clear-cut definition can be provided. We proposed to distinguish the various manifestations by categorizing along the two dimensions "unit of analysis" and "scale of outcome variable". With regard to the first dimension, we distinguished the document, sentence, and sub-sentence level as unit of analysis. Concerning the second dimension, we consider

binary polarity classification (positive vs. negative), ternary classification (positive vs. negative vs. objective), and ordinal regression (classification by means of a rating scale or by sentiment strength). We described the different tasks in more detail and discussed the relevant related work. As training data is conveniently available, most work has been published for the task of binary, document level polarity classification. For the same reason, but vice versa, much less work can be found for the sentence level or even sub-sentence level tasks. We further pointed out the similarity between supervised text categorization (i.e., classification by topic) and supervised polarity classification. We identified linguistic phenomenons such as sentiment shifters as being the primary factor that renders polarity classification more difficult.

For textual data, the classification performance with machine learning methods much depends on the chosen document representation. We were thus interested which types of features have been proven to be helpful for polarity classification tasks. In Section 10.2, we presented our results of a critical literature review regarding the utility of different feature types. We distinguished lexical, knowledge-based, linguistic, and sentiment shifter features. Many different document representations have been proposed, but unfortunately we also found many inconclusive results, especially with respect to complex, linguistically inspired feature types. Table 10.1 summarized the results of our analysis and Section 10.2.5 already pointed out our main conclusions, which we will therefore not repeat here.

In Section 10.3 we set the agenda for sentence level polarity classification with weakly labeled data. After revisiting the concrete problem setting, we then reasoned about the availability of weakly labeled data for this task. In particular, we pointed out that sentence level data is not directly available, but can be extracted from pros/cons summaries of customer reviews. We discussed the main properties of this kind of data and identified the main reasons for errors with regard to the weak label decision. Main problems were that pros/cons summaries are often composed of simple enumerations of aspects, may simply indicate the absence of any (dis)advantage, or may express rather mixed sentiments. All these issues can lead to wrong label decisions. To alleviate these problems, we proposed several filtering heuristics that were able to increase the precision of the label decision up to 97%. Later experiments also showed that this higher quality of (weakly labeled) training data truly leads to better classification results. We further reasoned about the availability of weakly labeled data for the objective class, presented some approaches, but eventually concluded that we could not find any appropriate source. Our idea to compensate for this lack of data was to use one-class classification techniques. We introduced the main concepts of this technique in Section 10.3.4.

Section 10.4 presented the results of our own experiments with sentence level polarity classification. The main focus of the experiments was on feature engineering and the different aspects with incorporating weakly labeled data within the classification task. We used support vector machines to learn classification models. In the following, we point out the main results of our experiments and recall our main conclusions:

Feature Engineering

- The first experiment evaluated the benefit of different linguistic preprocessing steps for generating term-based (in this case unigram) features. We found inconclusive results and concluded that the choice of appropriate preprocessing is dependent on the dataset. For our datasets, we decided to use simple case-folding of terms. Also part-of-speech tagging or lemmatization did not prove to be useful (see Table 10.4).

- N-gram features are generally superior to unigram features, but only if sufficient training data is available or feature selection techniques are applied (see Tables 10.6 and 10.7, Fig. 10.8, and Section 10.4.6).

- Knowledge-based features derived from sentiment lexicons significantly increase classification

performance. However, the added value with this feature type decreases and eventually becomes marginal, the more training data is available. Incorporating sentiment lexicon features did not improve n-gram-based classifiers that were built with huge amounts of training data. As it is typically difficult to decide how much training data is sufficient, we conclude that it is generally helpful to include sentiment lexicon features.

- If the training data and the target data contains user provided signals such as an overall review rating, including this information as feature type has proven to be useful. We observed significant increases in overall classification performance. Furthermore, the effects of adding this feature type were complemental to other feature sets. The utility of the feature is restricted by its availability in the training and target data (for instance, blog or forum data normally does not contain user ratings).
- Incorporating features based on context information about the sentiment status of neighboring sentences has shown to be beneficial. We used a sentiment lexicon to calculate the sentiment status of neighboring sentences. To include context information, one might also think of using other classification techniques, for instance conditional random fields. Such an approach allows to directly model dependencies between individual sentences.
- We also tested features derived from the occurrence of sentiment shifter words. Such words were identified by means of a lexicon-based method. The features were of no help.
- We built pattern features by masking individual tokens by their part-of-speech tag or symbols based on the occurrence of a sentiment word, sentiment shifter, or product aspect. We could measure significant improvements with this feature type compared to an unigram model. However, in conjunction with the more simple lexicon-based features, we did not observe any increase in performance. We thus concluded that this type of feature is not helpful.

General Results

- Subjectivity detection is more difficult than binary polarity classification, at least for our datasets. The error rate for separating positive/negative sentences from objective sentences was generally higher than for distinguishing positive from negative sentences. Especially the distinction between the negative and objective class turned out to be difficult.
- The error rate with regard to polar facts is significantly higher compared to sentences with explicit sentiment expressions. The two main reasons are that less training data for polar facts is available and that the lexical variability is generally higher with polar facts.
- For both corpora, we found that the share of polar facts is higher for the negative class. This is one reason why classification performance with respect to this class is worse for our datasets.
- Our hypothesis that a cascaded approach to ternary polarity classification is superior to a conventional one-vs.-rest strategy could not be confirmed. We observed significantly better results with the one-vs.-rest strategy.

Weakly Labeled Data

- Weakly labeled data derived from pros/cons summaries of customer reviews is generally helpful for binary polarity classification. We achieved significantly better results when incorporating weakly labeled data. If enough weakly labeled training data is available, we could completely go without manually labeled data. The benefit is thus twofold: We can raise classification performance while simultaneously reducing the costs of obtaining training data. With such an

approach, we can thus easily build polarity classifiers for a variety of application domains (of course constrained to customer review data).

- Our filtering heuristics for increasing the quality of weakly labeled data have proven to be effective. We observed improvements in classification performance (binary polarity classification) of 2-3 percentage points.
- For the task of supervised subjectivity detection we found that we can partly substitute manually labeled data with the weakly labeled pros/cons data. When substituting samples for the subjective class with weakly labeled data, we obtained reasonably good results. Classification performance was lower in comparison to exclusively using manually labeled data, but differences for the target class were relatively small.
- To completely eliminate the need for manually labeled data also for the subjectivity detection task, we examined the use of one-class classification techniques. Our experiments revealed that this approach does not lead to the desired results. We could not create adequate subjectivity classifiers and obtained even worse results in comparison to a simple, unsupervised, lexicon-based classifier. In conclusion: If no weakly or manually labeled data for the objective class is available, it is more promising to rely on an unsupervised, lexicon-based approach than on a one-class classification technique.

Part IV.

Discussion

Alles Gescheite ist schon gedacht worden, man muß nur versuchen, es noch einmal zu denken.

Johann Wolfgang Goethe

11. Summary and Conclusion

With the emergence of the Internet as a social and interactive platform, an increasing share of public discourse and opinion making is taking place on the Web. Customer reviews represent a very prominent example where people share their opinions and experiences online. For companies and consumers such genuine customer voices represent extremely valuable information and they would like to have tools that automatically analyze and summarize this textual data.

In this work, we presented an in-depth analysis of *aspect-oriented customer review mining*. Part I introduced the concrete problem setting, provided the necessary background information, and gave an overview of related research areas. As reviews are composed of natural language text, we started our study with a linguistic analysis of the problem setting in Part II. Our primary goals were (i) to characterize the ways reviewers express their opinions towards a product and (ii) how to operationalize these characteristics within a formal model. We proposed two separate models that differ in the granularity of analysis (phrase level vs. sentence level). We implemented both models by appropriate annotation schemes and hand-labeled a large set of customer review texts. The resulting text corpora served mainly two purposes: to quantify and characterize the problem setting (corpus analysis) and to create a ground truth for the evaluation of algorithmic approaches (gold standard). Part III of the thesis addressed the problem of computationally treating the expression of opinions in customer reviews. We focused on the two main subtasks of aspect-oriented sentiment summarization: (i) identifying relevant product aspects and their mentions in review texts; (ii) detecting expressions of sentiment and determining their polarity. For both subtasks, aspect detection and sentiment analysis, we experimented with unsupervised (lexicon-based) and supervised approaches. As an overarching question, we examined how we can apply *distant supervision techniques* to reduce the costs of creating lexicons or labeled training corpora. In the following section, we summarize our contributions in detail, point out our major findings, and draw conclusions from the obtained results.

11.1. Summary of Contributions

11.1.1. Models, Datasets, and Corpus Analysis

To understand the problem setting and to provide a basis for computational treatment, we first needed to devise formal models that describe the constituents of sentiment expressions in natural language text (Chapter 4). Since a general model can be arbitrarily complex, we restricted the models' functional requirements to the concrete application domain of customer review texts. As a first step, we elaborated on the main subject of a review, namely the product. We proposed to model products and related product aspects by means of a hierarchically organized *product type taxonomy*. We argued that the hierarchical structure helps to summarize extracted information in a comprehensive form. Regarding the taxonomy, we differentiated between coarse-grained, *concept level* aspects and fine-grained, *mention level* aspects. Concept level aspects subsume and semantically group mention level aspects. As a second step, we examined how sentiment is expressed towards the product aspects. For analysis at the concept level, we proposed to dissect a review text along three main dimensions, each reflecting a unique information need. The resulting *discourse oriented model* splits a review into individual text segments that are coherent with respect to (i) discourse functions, (ii) aspect-related topics, and (iii) sentiment polarity. We further proposed the *expression level model*, which captures the linguistic constituents of sentiment expressions and allows for more fine-grained analysis. Borrow-

ing from prior work such as *appraisal theory* and the concept of *private states*, we "disassembled" the textual manifestation of an opinion into three functional components: (i) sentiment expressions, (ii) sentiment targets, and (iii) sentiment shifters.

In reference to both models, we developed annotation schemes that we used to manually label customer reviews for two exemplary application domains — namely, hotel and digital camera reviews (Chapter 5). We provided detailed annotation guidelines that may serve as a basis to extend or adapt our corpora (Appendix A). Our basic decision to create our own evaluation corpora from scratch was motivated by the lack of adequate, widely accepted corpora that met our requirements. Based on the annotated corpora, we could analyze the problem setting in detail, quantify particular (linguistic) phenomenons, and decide on the relevance of individual subproblems (Chapter 6). Our corpus analysis revealed the following findings:

- Customer reviews are highly focused documents. About 60% of all sentences express sentiment towards a relevant product aspect. If a reviewer expresses any sentiment, it is most likely that he evaluates the product. 93% of polar sentences mention a product aspect. We concluded that a review mining system can benefit from *jointly* considering sentiment expressions and product aspects.

- Polar facts constitute a significant phenomenon, especially with regard to negative evaluations, where ~30% can be ascribed to polar facts. It is worthwhile to recognize polar facts, but we also found that automatic approaches have difficulties.

- The content of customer reviews can be attributed to a small set of discourse functions. Our set of 16 predefined functions covered 97% of all sentences. Some discourse functions (e.g., "conclusion", "advice", "lack") reflect information needs (e.g., "What is missing, what are my customers' wishes?") that can be relevant in specific application scenarios.

- Nominal mentions of product aspects are by far the most frequent. They account for more than 90% of all occurrences of product aspects (> 80% for sentiment targets). It is therefore reasonable to put most effort on detecting this mention type. Only 6-10% of sentiment targets refer to pronominal mentions of aspects. Thus, additional coreference resolution could increase the recall by at maximum 10%.

- Sentiment targets are distributed according to a power law. The 50 most frequent types (~10% in our corpora) represent about 70% of all (nominal) occurrences. This partly favors the use of simple lexicons; a small dictionary of 50 entries already covers the majority of occurrences. On the other hand, low-frequency types (the *long tail*) will be hard to detect with lexicons.

- More than 20% of all sentiment expressions exhibit a target-specific polarity, where 90% of these are adjectives. It is thus reasonable to account for this phenomenon.

11.1.2. Automatic Acquisition of Product Aspect Lexicons

As a necessary requirement to constructing a product type taxonomy, we need to know which product aspects are relevant with respect to a particular product type. Depending on the application scenario, the aspects may either be predefined or they may need to be derived from relevant text corpora. Our contribution in Chapter 7 was to examine unsupervised methods to automatically derive relevant product aspects from large collections of review documents. In particular, we proposed to cast the task as a *terminology extraction problem*. Our implementation followed a pipeline architecture, which covered components such as linguistic pre-processing, candidate acquisition, candidate filtering, variant aggregation, and candidate ranking. We experimented with several approaches for each of these components. Our main findings in this chapter were:

- By choosing appropriate acquisition, filtering, and ranking techniques we could increase the precision of the extracted lexicons by around 15 percentage points compared to a baseline approach. Depending on the application domain, 73% - 85% of the extracted lexicon entries referred to valid product aspects. Errors are mainly due to the frequency based ranking methods. They tend to extract non-aspect terms that are either closely related to (1) the product type (e.g., "subway", "Manhattan") or (2) the domain of reviews (e.g., "problem", "disadvantage").

- The automatically constructed lexicons were composed of around 1,000 terms. Due to the high accuracy, we were able to manually revise each lexicon in around 3 hours. The revised lexicons covered around 87% (hotel) and 94% (camera) of all nominal product aspect mentions in our evaluation corpora (lenient metric). With these lexicons we were able to achieve f-measure values of around 80% for the aspect detection task. For the detection of sentiment targets, we could achieve f-measures of up to 86% in the best scenario. False positive were mainly produced by the missing context awareness of lexicon-based extraction (e.g., lexical ambiguity). Not recognized law-frequency terms ("Zipf's law") were responsible for most false negatives.

- Candidate ranking algorithms based on contrastive relevance measures such as the LRT- or MRRF-score performed best. Our experiments with combinations of different measures in a weighted-rank scheme did not reveal any improvements.

- The choice of acquisition patterns and heuristics is important. We observed differences in f-measure of around 15 percentage points. We found that the bBNP heuristic consistently outperformed all other methods. Incorporating sentiment information in the acquisition process was not beneficial.

- Comparison to related work, showed that utilizing a separate, large foreground corpus for lexicon acquisition substantially improves the recall. We achieved the best f-measure with foreground corpus sizes between 1000 and 5000 documents. Larger corpora lowered the precision.

11.1.3. Detection of Product Aspect Mentions at the Sentence Level

In Chapter 8, we considered the task of detecting aspect-related topics in customer reviews. A first step was to identify which topics are relevant for a given product type. Instead of relying on expert knowledge, we proposed to follow a data-driven approach. In particular, we described a semi-automatic framework: We used *probabilistic topic models* to automatically gather the main themes discussed in a large collection of reviews (Appendix D). A manual post-processing step refined the results by filtering out domain irrelevant or incoherent topics. To create a product type taxonomy, we combined the results of topic extraction with the results obtained via the terminology extraction component. More precisely, we manually organized the topics in a two level hierarchy (concept level) and associated each extracted term (mention level) to one of the topics. Having identified the set of relevant aspect-related topics, we then compared different methods to automatically detect mentions of the topics in natural language text. We cast the task as a *multi-label, multi-class text categorization problem* at the sentence level: The goal was to attribute each sentence to zero (off-topic) or more (on-topic) of the predefined topics. To this end, we examined a lexicon-based method and two supervised settings. For the lexicon-based method, we utilized the information encoded in the product type taxonomy to associate sentences with aspect-related topics. For the supervised approaches, we first mapped the multi-label, multi-class problem with *binary relevance transformation* and then trained multiple binary maximum entropy classifiers. The two supervised settings differed in the way training was conducted. In the first setting, we used our manually labeled training corpora to learn classification models, in the second setting we experimented with a distant supervision approach. Our distant supervision assumption was that we can map section headings of reviews to the aspect-related topics,

so that we can extract the headed paragraphs as training data. The main results of our experiments in Chapter 8 were:

- The topic modeling approach is an effective tool for discovering the main themes in a set of customer reviews. We found that a sentence-oriented document representation was better suited in our context and setting the number of topics between 50 and 70 was most reasonable.
- Even with the relative simple lexicon-based approach, we could achieve micro averaged f-measures between 70% and 75%. The precision was generally higher than the recall. The major reason for reduced recall was the failure to recognize non-nominal mentions of aspects (> 80% of all false negatives). The major reason for reduced precision was the lack of context-awareness of the simple string matching approach (> 80% of all false positives).
- The supervised approach outperformed the lexicon-based method by more than 10 percentage points in f-measure. In contrast to the lexicon-based method, the classification models also recognized implicit aspect mentions. The best results were achieved for a setting where we used the lexicon information as additional features for the supervised predictors (up to 5.5 percentage points improvement). Mistake analysis revealed four important sources for failures: data sparsity, imbalanced data, information loss due to the binary relevance transformation, and weak or missing context information.
- The distant supervision approach generated highly accurate training corpora (91% of the weak labels were correct). Classifiers trained on the weakly labeled data showed nearly the same performance as the baseline classifiers, which were trained on the manually labeled data. When combining weakly and manually labeled corpora, we could even improve over the baseline (+3 percentage points).

11.1.4. Automatic Acquisition of Domain-Specific Sentiment Lexicons

In Chapter 9, we considered the task of automatically constructing sentiment lexicons. In this context, we put special emphasis on integrating domain-specific knowledge. We pointed out that the prior sentiment polarity of many words and phrases is dependent on the application domain or may even be dependent on the addressed product aspect (e.g., "*long* battery life" vs. "*long* shutter lag time"). Whereas most previous works do not consider this aspect, we proposed an indirect crowdsourcing approach that allows to automatically derive a domain and context-aware sentiment lexicon. In particular, we devised a method that leverages the short, semi-structured pros/cons summaries of customer reviews. We designed high-precision heuristics that extracted tuples of sentiment word candidates and product aspects from the pros/cons texts. Using a hypothesis test, we selected those tuples that occur with significantly higher probability either in the pros or in the cons. Based on this analysis, we defined the sentiment polarity of the tuples and adapted a general purpose sentiment lexicon accordingly. In our experiments, we compared our approach to other state-of-the-art approaches. Our main findings were:

- Our high-precision extraction patterns were indeed very accurate. Around 95% of the extracted target-specific lexicon entries were correct. For the domain-specific lexicons the precision was slightly over 80%. The recall of the extraction patterns was strongly dependent on the actual format of the pros/cons summaries. In general we found that the vast majority of extractions can be obtained with very few, simple patterns.
- Including the automatically extracted domain and target-specific lexicon entries led to major improvements in sentiment polarity detection. We observed increases in f-measure of up to 14 percentage points compared to a baseline approach.

- We tried to use the extracted sentiment words as additional seeds for a label propagation approach. This method led to worse results than simply adding the terms to a general purpose sentiment lexicon. We further tried to use label propagation to expand a domain/target-specific lexicon. Also this approach was not successful, we could not observe any improvements.

11.1.5. Polarity Classification at the Sentence Level

Chapter 10 provided a detailed analysis of supervised methods for the task of sentence level polarity classification in customer reviews. We cast the task as a binary (positive vs. negative) or ternary (positive vs. negative vs. factual) text categorization problem. We pointed out that, compared to traditional document level categorization, classification at the sentence level is likely to suffer from data sparsity. We thus hypothesized that feature engineering is very important and as a first step reviewed the relevant literature. We found that reported results were quite inconsistent and therefore conducted our own experiments with different feature sets. As a main contribution, we studied the utility of a distant supervision approach. We devised heuristics that exploit the information contained in pros/cons summaries to automatically generate labeled training corpora. As we were unable to find an appropriate source for extracting weakly labeled samples for the objective class, we tried to overcome this problem by means of *one-class classification techniques*. The main results or our experiments in this chapter were:

- N-gram features were generally superior to unigram features if sufficient training data was available or feature selection techniques were applied. Including sentiment lexicon features was helpful for smaller training corpora, but for large training sets we could not observe any improvements over an n-gram baseline. Encoding the sentiment status of adjacent sentences as a feature was also beneficial. Other types, such as the sentiment shifter features or part-of-speech pattern features, did not lead to better results. Also linguistic pre-processing (e.g., lemmatization) showed only marginal (and inconsistent) effects.

- Separating positive/negative sentences from objective sentences was more difficult than distinguishing positive from negative sentences. Especially the distinction between the negative and objective class turned out to be difficult for the classifiers. This was mainly due to the significantly higher rate of polar facts in the negative class. Polar facts were generally harder to detect because explicit clues are missing, contextual knowledge may be necessary, and lexical diversity is higher. Only slightly more than 50% of the polar facts were correctly classified.

- The distant supervision approach generated highly accurate training corpora (95%-97% of the weak labels were correct). For the task of binary polarity classification, the weakly labeled data perfectly substituted the manually labeled corpora. We achieved comparable or even better results with the weakly labeled data.

- For the task of supervised subjectivity detection we found that we can partly substitute the manually labeled data with the weakly labeled pros/cons data. The classification performance was lower in comparison to using the complete manually labeled corpus, but differences were relatively small.

- The proposed one-class classification approach for subjectivity detection did not lead to the desired results. The obtained classifiers performed even worse than simple lexicon-based classifiers and were thus not useful.

11.2. Discussion of Results

We have seen that the expression of opinion is in general a complex linguistic phenomenon. Opinions may be expressed implicitly (e.g., polar facts), sentiment shifters may strengthen, weaken, or neutralize opinions, irony or sarcasm may flip the polarity, and multiple, nested sentiment sources may exist. Also sentiment targets may be implicit and typically appear in different forms (nominal, named, pronominal). All these phenomenons can be observed in customer reviews too. However, our results show that many of the more complex constructs occur with rather low frequency in our review corpora. For instance, less than 3% of all sentiment expressions exhibit neutral polarity, less than 4% address multiple targets, and less 5% are affected by the shifter types "downtoner", "softener", and "solidifier". Also the vast majority of sentiment expressions is explicit (> 80%) and most sentiment targets appear as nominal mentions (> 80%). For a proper linguistic theory of opinion expression all the specific phenomenons and subtleties are important, but when designing a review mining system, it is reasonable to put emphasis on correctly recognizing the most frequent patterns. In other words, the Pareto principle is also true for sentiment analysis of customer reviews. The major share of sentiment expressions and targets can be correctly recognized with comparably simple methods — for example, with appropriate lexicons and nearest-neighbor heuristics. A smaller share requires more complex linguistic/syntactic analysis. In consequence, we believe that lexicon-based approaches represent a good starting point in developing an aspect-oriented review mining system. They promise reasonably good results, require relatively low effort, and are open to extension — for example, by integrating the lexicons into a rule-based system. In addition, we have seen that lexicon information is also beneficial in supervised settings.

Due to the importance of lexicons, we extensively discussed approaches to automatically create such lexical resources. We proposed methods to generate product aspect lexicons and we considered techniques to build sentiment lexicons. These automatic approaches will never be free of errors and will never generate "complete" dictionaries. We thus regard the presented techniques as part of a semi-automatic framework, rather than as part of a fully automatic setting. In such a semi-automatic setting, the automatic extraction only supports the human supervisor. In a productive environment it would be most important to provide convenient tools to create and maintain the knowledge bases. For example, whereas we needed to manually edit XML files or revisit the results of topic modeling in a text editor to create the product type taxonomy, it would be much more convenient to provide an integrated, graphical environment.

With regard to the automatic construction of product aspect lexicons, low-frequency terms were most problematic for our terminology extraction methods. This was to be expected as frequency-based approaches inherently have difficulties in detecting terms in the "long tail". Incorporating this long tail in a dictionary obviously involves much effort and other techniques need to be applied to recognize such terms. Some researchers propose to use relationship detection techniques to find low-frequency terms that are targeted by a sentiment expression [177, 183]. However, this requires that sentiment words are correctly identified and relationship detection is accurate. We find that many of the low-frequency terms are actually variants (e.g., misspellings, near-synonyms, compositional variants, or specializations) of more frequent terms. Although our variant aggregation experiments only showed marginal improvements, we believe that more sophisticated variant detection algorithms in combination with convenient tool support (for grouping and ranking variant candidates) represent a good alternative.

With regard to sentiment lexicons, we pointed out that domain adaptability is an important factor. We achieved improvements of up to 14 percentage points with our adapted lexicons compared to a general purpose sentiment lexicon. In general, we think that adaptability is an important property of a review mining system. This primarily refers to domain adaptability, but also to other aspects. For instance, the relevance of terms (e.g., product aspects) evolves over time. Some product aspects may be relevant only for a certain period of time. Also adaptability to other languages plays a major role.

We believe that our proposed, unsupervised indirect crowdsourcing techniques to lexicon creation are very helpful in this direction (for example, they are language agnostic).

We considered aspect-oriented review mining at two levels of granularity. The expression level model allowed to recognize fine-grained product aspects and individual sentiment expressions. The discourse oriented model ignored the linguistic constituents and considered the functional components of an opinion expression at a coarser-grained level of analysis. We believe that both models complement each other. For instance, since at the sentence or paragraph level more context information is available, we can better cope with implicit aspect mentions or polar facts. We may also combine both models to provide a product centric and a product type oriented perspective at the same time (see Chapter 7). For example, similar to the system proposed by Blair-Goldensohn et al. [40], the topics of the discourse oriented model may serve as "static" aspects, which represent a whole product class, and distinct sets of fine-grained aspects would be collected for each individual product.

As an overarching topic of the thesis, we were interested in methods to reduce the costs involved with creating labeled training corpora or lexical resources. We proposed and examined several new indirect crowdsourcing (distant supervision) methods that exploited the metadata and structure of customer reviews. For example, the information encoded in pros/cons summaries helped us to create huge corpora for sentence level polarity classification and to automatically derive domain/target-specific sentiment lexicons. We were able to devise high-precision heuristics that allowed to extract very accurate datasets. With the presented methods we could successfully reduce the required amount of human supervision for several important tasks. For some tasks (e.g., polarity classification or topic detection) the weakly labeled data perfectly substituted the hand-labeled corpora. However, whether we can benefit from automatically extracted datasets depends on the considered task, the application domain, and the availability of adequate data. For example, we did not find an adequate data source for extracting samples of factual sentences for the subjectivity detection task. It is also questionable whether our approaches are transferable to other application domains, such as sentiment analysis of newswire text. In general, with indirect crowdsourcing methods, we often find samples for the "positive" class only. This was the case for the subjectivity detection task, but to some extent also for topic detection: Exploiting the section headings, we could derive samples for all predefined topics. However, the method did not allow to find samples for off-topic sentences (i.e., sentences that do not refer to one of the predefined topics). For the subjectivity detection task, we experimented with a one-class classification approach to alleviate this lack of data. Unfortunately, our attempt was not successful. In such situations, if data for the negative class is missing, indirect crowdsourcing techniques can at least partly substitute the need for human supervision. In any case, distant supervision generally allows to create very large amounts of labeled data — typically some orders of magnitude more data than possible with traditional annotation. We observed that this aspect was also relevant for the design of some concrete approaches. For instance, with thousands or millions of training samples, simple n-gram features can be superior to more complex linguistic features (e.g., syntax features such as dependency paths in a parse tree) for the task of polarity classification.

11.3. Outlook

11.3.1. Distant Supervision

We used distant supervision techniques to derive weakly labeled training corpora. Whereas our extraction heuristics were quite accurate, they were not perfect; five to ten percent of the label decisions were false. We simply ignored this fact. Instead, we hypothesized that the large amount of extracted data and the robustness of the applied machine learning algorithms can compensate these errors. While we already achieved good results, an open question is whether the results can be improved by explicitly recognizing that the labels are weak. How to learn in the presence of noisy labels is primary

a research problem in the machine learning community [15]. For example, some approaches try to remove or correct samples with erroneous labels in a pre-processing step [54, 268, 317, 467]. Others adapt supervised learning algorithms to cope with noisy labels [28, 225, 447]. For future work, we would like to examine whether these more sophisticated machine learning techniques are beneficial in our context.

Closely related to learning from noisy labels is the setting of semi-supervised learning (i.e., learning from labeled and unlabeled data). In fact, we can think of combining distant supervision techniques with semi-supervised learning approaches. For instance, we could use the weakly labeled data in a bootstrapping setting [196] or combine weakly labeled and hand-labeled data with an expectation maximization approach [283]. Instead of directly training on weakly labeled data, we may also use the extracted information to pre-label [25, 93] a text corpus. In this case human supervision would be reduced to the revision of incorrect samples, which lowers the per-annotation costs.

Earlier we pointed out that we experimented with one-class classification approaches to compensate the lack of labeled training data. Our particular approach with one-class SVMs was unsuccessful. However, other methods that address the problem of learning in the absence of counter examples have been proposed in the literature [235, 379, 456]. We think it is worth to experiment with these approaches in the context of distant supervision approaches.

11.3.2. Domain Adaptability and Cross Domain Settings

In this work, we explicitly set focus on sentiment analysis of customer review documents. The presented models and approaches were tailored towards this application domain. However, similar information (i.e., feedback on products and services) exists in many other formats. For instance, people share their opinions and experiences in blogs, microblogs, message boards, or social networks. It is not clear how well the developed models and approaches fit to other application scenarios. When applying or adapting our approaches to other domains, we primary need to cope with varying text genres. In particular, we need to consider the different styles of writing, the availability/lack of meta data, the varying length of documents (e.g., microblog posts vs. customer reviews), or the degree of structuring (e.g., message boards vs. blogs). For example, can we train polarity classification models on review documents to predict the polarity of blog or microblog posts? Similar to our work, the major share of research on sentiment analysis considers a single and often quite restricted application domain. Domain adaptability of approaches and cross domain settings only recently gained more attention [14, 20, 44, 260, 410]. We believe that this problem setting deserves a closer look. For example, it is highly relevant for practical sentiment analysis systems; information typically needs to be obtained from very different data sources. As a first step, we think that it is important to acquire a better understanding of the characteristics of different text genres with regard to sentiment analysis. Future research should elaborate on qualitative and quantitative criteria to describe these differences. Such an analysis would help to examine domain adaptation approaches more systematically. Closely related is the question whether and how distant supervision techniques can be beneficial in cross domain sentiment analysis settings.

11.3.3. Corpora

In Chapter 5, we already indicated the lack of well-known and widely accepted evaluation corpora for many sentiment analysis tasks. Whereas the MPQA Opinion corpus [413] may be regarded as a reference corpus for sentiment analysis of newswire text, similarly established corpora do not exist for other analysis tasks. For instance, we find many customer review corpora, but most of them are heuristically extracted and they differ widely with regard to the underlying annotation schemes or the granularity of analysis. Subtasks in sentiment analysis are often not clearly defined (or interpreted differently). No standardized annotation schemes exist that could serve as a basis for creating

reference corpora for a specific sentiment analysis task. In consequence, comparisons of different approaches for the same task are difficult and often questionable. As indicated in the previous section, the application domain and text genre also play a major role with regard to the exact definition of tasks and annotation schemes. We think that for future research it would be beneficial to establish widely accepted annotation schemes for the most relevant subtasks and application domains. This would help to create larger evaluation corpora, guarantee better comparability, and thus foster the development of new approaches.

11.3.4. Discourse Functions

We introduced the notion of discourse functions to describe the function of individual text segments within a review document (e.g., to express an expectation, to describe a problem, to conclude the review, or to express a wish). Although we postulated an information need with regard to these functions, we did not examine any algorithmic approaches to automatically recognize the information. As we are convinced that automatic analysis of discourse functions is relevant for certain application scenarios, we think that future research should cover this dimension more closely. For example, a review mining system that addresses discourse functions might answer queries such as "What do customers *expect* from my products?", "Which aspects are mentioned most frequently in the context of *problem descriptions*?", "What are the *wishes* of my customers?", or "How do customers *use* the product?". Future research may develop a more fine-grained model of discourse functions and should experiment with computational approaches to recognize the functions in customer review texts.

Appendix

A. Annotation Guidelines

In this chapter, we provide comprehensive annotation guidelines for the two annotation schemed presented in Chapter 5. The guidelines primarily serve to enforce the creation of **reliable and consistent annotations**. The inherent ambiguity and variety of natural language in general, and in particular the fact that perceptions of opinions are dependent on the annotator's individual background and knowledge, render the annotation process a difficult task. Therefore, with the guidelines we present a reference to the annotator, giving concrete instructions and hints on how to identify, interpret, and mark the different types of annotations defined in the scheme. Instructions and hints are described in the form of examples, associated with reference annotations and a detailed explanation of the specific case. All examples represent excerpts from the actual datasets (opposed to being artificially constructed). The examples are unaltered, that is, grammatical mistakes and misspellings are not corrected.

Besides serving as a reference to the annotator, the guidelines also represent an ideal source to gain deeper insight into the shape of the corpora without being forced to examine their raw data. In other words, the well structured and carefully chosen list of examples can be interpreted as a **commented and explained excerpt of the corpora**. We hope that the guidelines are that well written and informative so that others can more easily use and interpret our datasets for their own studies or for reproducing our results. Thus, they may serve as a **starting point for the adaptation of our annotation schemes**, for instance in order to create a further labeled dataset of another product domain or text genre.

The remainder of this chapter is structured as follows: We first provide some general remarks in Appendix A.1. The following two sections describe the guidelines for the discourse oriented annotation scheme (Appendix A.2) and for the expression level scheme (Appendix A.3).

A.1. General Remarks

- The annotation guidelines provide example sentences with associated reference annotations. For the sake of simplicity and clarity, our representation of annotations uses short names of attributes. For examples the attribute *DiscourseFunction* is referred to as *function* or *SentimentPolarity* is referred to as *polarity*. It is always clear from the context which attribute is meant.

- With respect to the attributes of annotation types, we use the terms *unset* and *empty* interchangeably. Depending on the tools used for annotation, the annotator indicates that an attribute has no value by either not setting the attribute at all or leaving it empty. In the former case the attribute is not existent as part of an annotation, in the latter case the attribute exists, but contains an empty string. In the following the special value `EMPTY` refers to an unset or empty attribute.

- During the annotation process the annotator needs to select values from different predefined lists of valid values. Such lists are for example the set of discourse functions or the set of coarse-grained product aspects. These lists have been compiled in a preliminary, data-driven study as explained in Appendix D.

- Independent of the type of the entity that is being reviewed (e.g., digital camera or hotel), we generally refer to the entity as a *product*. Although the term *service* would better fit in the context of hotel reviews, we do not make such a distinction. Observe that this may have implications

with regard to the use of language: For example, if we speak of "purchasing a product", you have to reinterpret it in the context of service reviews (e.g., hotel) to a formulation similar to "booking a service" or "paying for a service".

A.2. Annotation Guidelines for the Discourse Oriented Model

A.2.1. Basic Instructions

For the sentence level annotation project, we represent each corpus as a single, huge XML document. The associated XML Schema structures the document into a set of reviews and each review consists of a set of discourse segments. The corpus was preprocessed with an automatic sentence splitter[1], so that each discourse segment element comprises exactly one sentence. More detailed information regarding this setup is provided in Appendix A.2.8.

The annotator treats each sentence as an isolated unit of annotation — that is, the context of a sentence (preceding or following sentences) is irrelevant for the annotation currently in focus. For each sentence in the corpus, the annotator proceeds according to the following instructions:

1. Read the sentence with care. Read the sentence again. Decide which kind of review specific discourse function the sentence exhibits. Classify the sentence into one of the provided functions (Table 5.4) by setting the discourse function attribute. If the predefined list of functions does not contain a matching entry, set the discourse function to `OTHER`.
2. Consider whether the sentence expresses an explicit opinion or is rather objective. Take this decision irrespective of whether a sentence is *on-topic* or *off-topic* (see step 4.). If you can identify an opinion expression, decide on the polarity of the expression by setting the sentiment polarity attribute.
3. If the sentence is objective, but expresses a fact that induces an apparently positive or negative impression, set the polar fact attribute to `true`. Decide on the polarity of the expression by setting the sentiment polarity attribute accordingly.
4. Consider whether the sentence addresses one of the topics predefined for the relevant product domain. Table 5.5 provides valid topics and subtopics[2] for the domain of hotel and digital camera reviews. If applicable, always prefer choosing the more specialized subtopic instead of its parent topic. If a sentence addresses multiple topics, enumerate all relevant, unique topics in a comma separated list.
5. If you are unsure about any of the attributes, first consult the annotation guidelines. Many standard cases and also the most common borderline cases are documented. If you are still unsure, choose the annotation you are most confident with, but mark the whole sentence by setting the confidence value to *low*.

For each attribute defined in the annotation scheme we now present guidelines that serve as reference to the annotator during the annotation process.

A.2.2. Guidelines for the Sentiment Polarity Attribute

Distinguishing Facts and Opinions

To set the sentiment polarity attribute correctly and consistently, you basically need to be able to identify expressions of opinions and to distinguish them from purely factual information.

[1] `http://www-nlp.stanford.edu/software/`

[2] Recall that a subtopic represents a specialization of its parent (generalized) topic — for example, `bathroom` is considered a subtopic of the more general `room` topic.

We distinguish two types of evaluative expressions: Reviewers may either convey their opinion explicitly, by use of subjective language, or may do so implicitly by referring to (polar) facts that imply a positive (desired) or negative (undesired) state. In the following, we put emphasis on the expression of explicit opinions and discuss the latter case in the guidelines for the polar fact attribute in Appendix A.2.3.

How can you identify opinions and distinguish them from facts? Most basically and following Wilson and Wiebe [435], an opinion can be described as a *private state* which Quirk et al. [309] define as a state "that is not open to objective observation or verification". Such a state reflects a subjective belief that is composed of for instance the attitudes, evaluations, judgments, ideas, and thoughts an author has about a certain topic. Consequently, and as the main distinguishing feature to factual information, an opinion is neither falsifiable nor verifiable: Typically, opinions are supported by arguments that are based on certain facts. Although you can reason about these facts being true or false, you cannot falsify the derived opinion as it only reflects an interpretation of the facts. Different individuals may derive different opinions from the same basic facts.

In case you are unsure whether a sentence expresses an explicit opinion or not, ask yourself the following questions:

1. Does the sentence represent a subjective belief of the reviewer?
2. Does the reviewer evaluate, assess or judge any subject matter?
3. Does the reviewer express any feelings or thoughts about a topic?
4. Could another person have different views or feelings with respect to the mentioned topic?
5. Can you grade the provided information?

In case you can answer one or more of the questions in the affirmative, this is a strong indicator that you need to set a value for the sentiment polarity attribute. The following examples highlight the distinction between factual information and opinion expression:

(A.1) The beds were soft and cozy and I had a wide selection of pillows to choose from.
[function=sentiment, topic=bed; polarity=positive]

(A.2) The room was dark and shabby and smelled of old smoke.
[function=sentiment, topic=room; polarity=negative]

(A.3) We received the room we requested with two double beds.
[function=fact, topic=room]

The first two examples (A.1) and (A.2) reflect a personal evaluation of the reviewer. In the first sentence, it is a subjective belief that the beds were soft an cozy. Observe that you cannot verify the information. You could measure "softness", for instance in terms of the mattress' compression, given a fixed pressure. However, which "compression factor" is interpreted as soft, medium, or hard is subjective. Another person that is more used to harder mattresses or has a higher body weight could have the impression that beds are much too soft and therefore anything else than cozy. Also the second sentence represents a subjective belief. The impression of a room's brightness and its general condition is subject to gradation. The reviewer might have the impression of shabbiness because he is used to luxury hotels. Both sentences are annotated with the sentiment polarity attribute set to an appropriate value.

In contrast, the third sentence (A.3) exemplifies a pure fact. The information could (at least hypothetically) be verified or falsified by examining the hotel's booking records. There is no room for interpretation, either the reviewer received the requested room or not.

(A.4) I was very happy with the hotel and would definitely come to the Kimberly again.
[function=conclusion, topic=PRODUCT; polarity=positive]

(A.5) One thing that irks me is the battery.
[function=sentiment, topic=battery; polarity=negative]

(A.6) The rated number of shots in the User Manual is based on using alkaline batteries.
[function=fact, topic=battery]

The preceding examples (A.4) to (A.5) point out the situation when reviewers provide a personal evaluation by expressing their *feelings* towards the product or an aspect — you would answer the third question in the affirmative. For instance, in sentence (A.4) the reviewer expresses a feeling of happiness with the stay and thus positively summarizes his overall impression. In example (A.5), the reviewer mentions a feeling of irksomeness with regard to the battery. In customer reviews, and as shown in the previous two examples, the expression of a feeling typically implies a positive or negative opinion with respect to the product. If this is the case, mark the sentence as containing an opinion by setting the sentiment polarity attribute appropriately. Sentence (A.6) also refers to the topic battery (in particular the battery lifetime), but does not express any opinion. It represents a fact that could be verified by reading the manual or checking with the manufacturer.

The following examples point out some more typical cases that you should annotate as opinions:

(A.7) No item is perfect and people are complaining about the battery life with the camera.
[function=other reviews; topic=battery; polarity=negative]

Also annotate a sentence as containing an opinion if the opinion holder is different from the author of the review. Example (A.7) points out such a case.

(A.8) I think it's obvious that any camera in this class, with a fairly high mega-pixel count (10.1 in this case) and a small image sensor is going to have some compromises.
[function=general remark; topic=picture quality; polarity=negative]

Reviewers may also evaluate the product by expressing an opinion which refers to the whole product class. By recognizing the product as a member of this class, the opinion is indirectly expressed on the concrete product itself. Annotate such sentences as containing an opinion. Example (A.8) highlights this case.

Typical cases when reviewers provide just factual information is when they cite the specification of a product. Examples (A.9) to (A.11) illustrate this situation:

(A.9) The pool level also had a basketball court, bowling alley, pool tables and more all available for rent.
[function=fact; topic=recreation]

(A.10) The camera accepts SDHC cards, which have a capacity of up to 16 GB.
[function=fact; topic=memory]

(A.11) This camera is very small, about 3 3/4 inches wide by 2 inches tall by 3/4 inch deep, and light.
[function=fact; topic=dimensions]

Setting the Value of the Sentiment Polarity Attribute

If you identified an explicit opinion or a polar fact, you must provide a value for the sentiment polarity attribute. It takes four valid values: `positive`, `negative`, `neutral`, and `both`. We do not capture a degree of positivity or negativity, that is, we do not distinguish between utterances such as "*outstanding* video quality" and "*good* video quality". Whenever a reviewer expresses an opinion that is neither clearly positive nor clearly negative, set the polarity value to `neutral`. The following examples clarify our instructions:

(A.12) Battery life is OK.
[function=sentiment; topic=battery; polarity=neutral]

(A.13) The battery life is about what I expected it to be, not bad, but not amazing either.
[function=sentiment; topic=battery; polarity=neutral]

(A.14) The batteries last quite long.
[function=sentiment; topic=battery; polarity=positive]

(A.15) Battery life is great.
[function=sentiment; topic=battery; polarity=positive]

(A.16) The battery life is outstanding.
[function=sentiment; topic=battery; polarity=positive]

(A.17) The battery life on this thing is absolutely incredible.
[function=sentiment; topic=battery; polarity=positive]

All examples (A.12) to (A.17) represent an evaluation of the battery life of a camera. However, they differ with respect to the polarity and intensity of the expressed opinion. In examples (A.12) and (A.13), it is clear that the reviewer's impression of battery life is rather neutral. It does not deviate from his expectation. From the reviewer's perspective, the characteristic of the aspect is neither an advantage nor a disadvantage. Annotate sentences with a similar types of sentiment with the polarity attribute set to `neutral`. In contrast, examples (A.14) and (A.15) reflect a clearly positive evaluation of the aspect battery life. It is very obvious that the reviewer regards the (long/great) battery life as an advantage of the product. We therefore mark the sentences with polarity attribute set to `positive`. Examples (A.16) and (A.17) show an even increased degree of sentiment intensity (outstanding/absolutely incredible battery life). However, we do not make a further distinction in degree of intensity and also annotate with polarity set to `positive`.

If you are unsure about setting the sentiment intensity to `neutral` consider the following criterion:

- Can you deduce from the reviewer's evaluation that the mentioned aspect represents a definite advantage or disadvantage of the product? And, if the product in its entirety is targeted: Can you deduce from the evaluation whether the reviewer would recommend or not recommend the product?

If you can answer the question with yes, then use `positive` or `negative` as value. If you answer with no, set the sentiment polarity attribute to `neutral`.

A reviewer may express both, positive and negative opinions in a single sentence. However, since the scope of annotation comprises a whole sentence, we cannot further distinguish these contrary opinions. If you identify such a situation, use the value `both` for the sentiment polarity attribute. Sentences (A.18) and (A.19) provide an example of this case:

(A.18) The LCD screen is large so no squinting to view pics however it is very fragile.
[function=sentiment; topic=screen; polarity=both]

(A.19) Rooms are small but with awesome views on times square.
[function=sentiment; topic=room,view; polarity=both]

A.2.3. Guidelines for the Polar Fact Attribute

In the previous section, we pointed out how to distinguish opinions from factual information. But we already know that mentioning certain facts may also reflect an opinion — that is, reviewers can evoke positive or negative impressions by means of providing factual information. Typically, such facts are concerned with desirable or undesirable properties of the product. The reader infers a positive or negative attitude by applying commonsense knowledge. In other words, the opinion is implicit and must be derived. We denote these kind of factual information as polar facts since we can attribute a positive or negative valuation to them. Thus, if a sentence is objective, but expresses a fact that induces an apparently positive or negative impression, set the polar fact attribute to `true` and decide on the polarity of the expression by setting the sentiment polarity attribute accordingly.

The following examples help you to distinguish between polar facts and "plain" facts:

(A.20) All our clothes and other belongings as well as our baby reeked of cigarette smoke.
[function=fact; topic=room; polarity=negative; polarFact=true]

(A.21) When we arrived we were told we would have the same room for both weeks.
[function=fact; topic=room]

Sentence (A.20) exemplifies the expression of a negative polar fact. It provides only factual information and does not include any subjective belief. However, common sense tells us that clothes (and even a baby) reeking of smoke are generally regarded as undesirable. In the context of hotel reviews it implies that a room smells too much of smoke. The sentence thus represents a negative polar fact that addresses the topic room. In contrast, sentence (A.21) contains plain factual information about the room. We cannot deduce any positive or negative appraisal from the given information.

(A.22) You can do the most common operations without even reading the manual.
[function=fact; topic=ease of use; polarity=positive; polarFact=true]

(A.23) The concierge offered me a complimentary paper for my breakfast and told me to enjoy my meal.
[function=fact; topic=service; polarity=positive; polarFact=true]

Sentences (A.22) and (A.23) illustrate cases where the polar fact evokes a positive impression of the topic. In the first sentence the reviewer implies that the product can be used in an intuitive way. From the second sentence, we can infer that the service staff is helpful and obliging.

Distinguishing Polar Facts from Opinions

Sometimes it is very clear that a sentence conveys a positive or negative appraisal, but you are unsure whether it stems from the expression of an explicit opinion or a polar fact. The following examples provide a helpful reference for this decision:

(A.24) Our room only had the view of other rooms and the lobby roof.
[function=sentiment; topic=view; polarity=negative]

(A.25) The view of the room directly faced a wall.
[function=fact; topic=view; polarity=negative; polarFact=true]

(A.26) The rooms provide just minimal amenities.
[function=sentiment; topic=room amenities; polarity=negative]

(A.27) The rooms do not offer any amenities.
[function=lack; topic=room amenities; polarity=negative; polarFact=true]

In both examples (A.24) and (A.25) reviewers comment negatively about the view from their room. In the first sentence a personal evaluation is expressed. It is not inherently obvious that a view on other rooms or the lobby is undesired, it might well be acceptable. However, by using the word "only", the reviewer makes clear that he or she disliked the view. The second sentence represents a polar fact. Here, it is obvious that the fact "view is facing a wall" is undesirable. A similar situation is highlighted in examples (A.26) and (A.27), which both address the amenities provided in a room. It is very subjective which and how many amenities a room should offer, thus the first sentence is annotated as an explicit opinion. On the other hand, offering no amenities at all is a fact and by common sense not desired.

A.2.4. Guidelines for the Topic Attribute

Topics refer to coarse-grained aspects of a product. In Table 5.5, we defined valid topics for the domains of hotel and digital camera reviews. For correct and consistent annotation, you should make yourself familiar with the topics.

Setting the topic attribute is optional. Leaving the attribute empty implies that the sentence is off-topic — that is, the reviewer does not address one of the predefined topics. Furthermore, the topic attribute is set independent of any other attribute. For instance, irrespective of whether the sentence expresses an opinion or not, you must consider whether the reviewer addresses a valid topic, and in case, set the attribute appropriately.

Deducing Topics

In the majority of cases, topics become manifest at the surface text of a sentence very clearly and explicitly. However, topics may also be implied and only indirectly observable. You as an annotator must deduce the correct topic, potentially applying your common sense knowledge. Consider the following examples:

(A.28) Something else that we appreciated was how happy and nice everyone was.
[function=sentiment; topic=service; polarity=positive]

(A.29) We did have to bother the desk twice more for extra towels and to find out the location of the hair dryer and they were quick and courteous.
[function=sentiment; topic=service; polarity=positive]

(A.30) The staff was extremely efficient, attentive and gracious.
[function=sentiment; topic=service; polarity=positive]

The first two sentences (A.28) and (A.29) highlight the case when a topic is expressed implicitly. Both address the topic "service", here in terms of evaluating the hotel staff. The reviewer does not directly mention the staff, but you can infer from the description that he comments on it. You know that the phrases "how happy and nice everyone was" or "they were quick and courteous" refer to the hotel staff. Also in example (A.30) the reviewer comments on the staff. But on the contrary, here the topic is named explicitly.

In any case, that is, irrespective of whether the topic becomes manifest directly or indirectly as part of the surface text, you must set the topic attribute appropriately. The annotation scheme does not distinguish between explicit and implicit aspect mentions.

Subtopics

Topics can be structured hierarchically. If reviewers comment on a subtopic, they implicitly evaluate the parent topic. For example, expressing a negative opinion on the topic "breakfast", also implies a negative impression with regard to the more abstract topic "dining". We only consider a two-level

hierarchy and denote the more specific child-topics as subtopics. Observe that not every top-level topic is further subdivided into subtopics (see Figs. 8.1 and 8.2).

How should you annotate with regard to this hierarchical structuring? Simply ignore the hierarchy and regard the set of top-level topics and subtopics as a plain list and always set the topic attribute to the most specific topic you can identify in the sentence. For instance, if the reviewer comments on the "huge and wonderful bath tub", set the topic to "bathroom" (instead of the more abstract topic "room"). The following examples illustrate the situation:

(A.31) Another annoyance is the Face Detection function is turned on each time you power up the camera.
[function=problem; topic=face detection; polarity=negative]

(A.32) I tried most of this camera's features, the result was disappointing.
[function=sentiment; topic=features; polarity=negative]

We know that the face detection is a feature of a digital camera, but it has also been predefined as a topic on its own right. So in sentence (A.31) the topic is set to "face detection", whereas in sentence (A.32) the reviewer refers to the features in general.

Multiple Topics

Due to the fact that annotations are attributed to a whole sentence, a single unit of annotation may refer to more than one topic. If you identify multiple different topics, annotate them all by providing a comma-separated list for the topic attribute (e.g., example (A.33)).

(A.33) The price was good, and the room was clean.
[function=sentiment; topic=price, cleanliness; polarity=positive]

A.2.5. Guidelines for the Non-Focus-Entity Attribute

With the non-focus-entity attribute we capture cases when an opinion is expressed, but the target is not the entity that is primarily in the reviewer's focus. Opposed to expert reviews, a customer review addresses (quasi per definitionem) a single entity, namely the product a reviewer has bought. However, reviewers may have possessed or tested similar other products, which they might mention or compare in their review.

Whenever a reviewer refers to a product or an aspect of a product that is not the focus entity, set the non-focus-entity attribute to `true`. Most often, this is the case in comparisons, as shown in examples (A.34) and (A.35), but can also occur in other contexts, for instance, as in sentence (A.36).

(A.34) Jummeirah essex house is a much better option.
[function=comparison; topic=PRODUCT; polarity=positive; nonFocusEntity=true]

(A.35) Panasonic TZ3 owned for 3 years had a great stabilizer system which can prevent the had movement very well but had very very bad indoor photos and low light pictures.
[function=comparison; topic=features, picture quality, low-light performance; polarity=both; nonFocusEntity=true]

(A.36) My wife owns a Canon point and shoot that is beautifully compact and really light-weight.
[function=sentiment; topic=dimensions; polarity=positive; nonFocusEntity=true]

A.2.6. Guidelines for the Irrealis Attribute

Polanyi and Zaenen [303] point out that the presence of modal operators may have a great influence on the perception of an opinion:

> Language makes a distinction between events or situations which are asserted to have happened, are happening or will happen (realis events) and those which might, could, should, ought to, or possibly occurred or will occur (irrealis events). Modal operators set up a context of possibility or necessity and in texts they initiate a context in which valenced terms express an attitude towards entities which do not necessarily reflect the author's attitude towards those entities in an actual situation under discussion. Therefore, in computing an evaluation of the author's attitude, terms in a modal context should not be treated precisely as terms in a realis context.

In summary, the polarity of opinion indicating words or phrases may shift or be neutralized in the context of irrealis events. The subsequent examples put emphasis on this effect:

(A.37) I really would have liked the room.
[function=sentiment; topic=room; polarity=positive; irrealis=true]

(A.38) Pictures seemed to look very nice on the LCD.
[function=sentiment; topic=picture quality; polarity=positive; irrealis=true]

(A.39) If you are not a novice like me, you would probably appreciate the wealth of settings.
[function=sentiment; topic=settings; polarity=positive; irrealis=true]

All three examples contain polar expressions ("liked", "nice", "appreciate", and "wealth") that a priori convey a positive impression. However, due to the use of modal operators, the evaluation is set into an irrealis context and the opinion is neutralized. Our goal is to capture these cases so that we can make a distinction between realis and irrealis comments. Annotate all such sentences with the *irrealis attribute* set to `true`. Only consider setting the attribute for polar sentences; for all other sentences you should leave the attribute empty.

A.2.7. Guidelines for the Discourse Function Attribute

The discourse function attribute accepts the $16 + 1$ valid values defined in Table 5.4. In this section we provide guidelines on how to identify the correct discourse function and how to distinguish the different categories from each other.

Advice

Reviewers often describe their personal experiences with a product or related subjects. In this context it is quite common that they offer some advice on the usage or how to circumvent any problems they encountered. The following examples show excerpts from reviews which should be annotated as `Advice`.

(A.40) I've been using the high ISO setting with the flash turned off to get nice indoor shots.
[function=advice; topic=picture quality]

(A.41) What I have found that works great is to push the shutter button halfway down then take the picture.
[function=advice; topic=user interface]

(A.42) I would highly recommend a "bay front" room or suite.
[function=advice; topic=room]

(A.43) The manual suggests you use the "natural" color mode setting to minimize the effects of the noise.
[function=advice; topic=settings]

Mark a sentence as `Advice`, if the reviewer describes a best practice with respect to the product or one of its aspects as shown in examples (A.40) to (A.43). Often a precise recommendation is given by the reviewer. Also classify a sentence as `Advice` if the reviewer just cites a best practice (in this case from the user manual), as shown in example (A.43).

(A.44) You must see Grand Central station a fantastic building.
[function=advice]

(A.45) Tip: go to the local starbucks for star sightings.
[function=advice]

Also mark sentences as `Advice` if the recommendation does not directly refer to the product or one of its aspects, such as illustrated in examples (A.44) to (A.45).

(A.46) I would recommend buying a larger size, I have a 64 mb which holds about 55 photos.
[function=advice; topic=memory]

(A.47) BUY at least a few more battery back ups!
[function=advice; topic=battery]

(A.48) It gets EXTREMELY loud between 2am and 4am so if you can, avoid the street front rooms.
[function=problem; topic=noise; polarity=negative; polarFact=true]

Typically, when describing a problem of the product, reviewers give an advice how to circumvent it. The specific problem might be expressed explicitly, as in example (A.48), or is implied by the context, as in examples (A.46) and (A.47). Mark such sentences as `Advice`.

(A.49) You need or must have a good understanding of basic photography, Understand ISO and other simple settings.
[function=advice]

Also annotate as sentence as `Advice`, if the recommendation addresses a requirement with regard to the user's abilities, skills or expertise.

Helpful Criteria to Identify the Expression of Advices In a great share of cases an advice is expressed in form of an imperative sentence. The absence of an overt subject, the verb being in its base form and the use of an exclamation mark as punctuation are good indicators for an imperative clause. In addition, a helpful indicator is when the reviewer directly speaks to the reader by making use of the pronoun *you* — for instance, as in examples (A.44) and (A.48) to (A.50). Also look out for verbs such as "suggest", "propose", "recommend", "advise", etc. or modals such as "must" and "should".

Advice versus Problem The discourse functions `Advice` and `Problem` are closely related. Reviewers tend to describe a problem and subsequently give advice how to solve it. If a problem description and the related advice appear in the same sentence, as occurs in example (A.48), it is unclear how to classify the sentence. Our simple rule is to give the problem a higher weight. Annotate a sentence as `Problem` if it contains both, a problem description and the related advice.

Advice versus Sentiment If an advice is based on a positive or negative evaluation of the product or one of its aspects, such as in example (A.50), mark the sentence as `Sentiment`.

(A.50) Second don't eat at the restaurant on site unless you like slow service, over priced poor quality food.
[function=sentiment; topic=Food; polarity=negative]

Comparison

We consult product reviews to find out which product best fits our needs and compare products by reading different reviews. However, comparisons between different products also occur within a single review. Reviewers highlight advantages and disadvantages of the reviewed product by comparing it to other ones. It is a typical stylistic device, which we recognize as the discourse function `Comparison`. The following examples help you to identify sentences that exhibit a comparison.

(A.51) The picture quality at 12M is inferior to that of my 4 1/2-year old Sony DSC-W5 at 1M.
[function=comparison; topic=picture quality; polarity=negative]

(A.52) We've previously stayed at the Omni Hotel, and both thought that Hotel Indigo was better.
[function=comparison; topic=PRODUCT; polarity=positive]

(A.53) This camera is cheaper than comparable models w/ similar features.
[function=comparison; topic=price; polarity=positive]

(A.54) The room size was definitely larger than most European hotels I've stayed in.
[function=comparison; topic=room; polarity=positive]

(A.55) This camera far surpasses any other digital point and shoot camera I have handled.
[function=comparison; topic=PRODUCT; polarity=positive]

Annotate a sentence with discourse function `Comparison` if reviewers compare a product aspect (example (A.51)) or the product itself (example (A.52)) to another entity. Instead of addressing a specific product, they also might compare to a whole group of entities, such as illustrated in examples (A.53) to (A.55).

(A.56) The room was definitely the best Sheraton I've stayed in - I think they were renovated within the last few years.
[function=sentiment; topic=room; polarity=positive]

Do not mark sentences as `Comparison` which contain a form of the superlative like in example (A.56). Here the discourse function `Sentiment` is prevalent.

(A.57) The 4x6 prints are comparable to that from a disposable 35mm film camera.
[function=comparison; topic=picture quality]

(A.58) Even though it's only 8.0 megapixels, the pictures are as crisp as my friend's 10.0 mp camera.
[function=comparison; topic=picture quality; polarity=positive]

(A.59) The service just didn't feel like the other IC Hotels that we have stayed at in the past.
[function=comparison; topic=service; polarity=negative]

Another common type of comparative forms are comparisons of similarity, for example as contained in examples (A.57) to (A.59). They should also be annotated with the discourse function attribute set to `Comparison`.

(A.60) Panasonic TZ3 owned for 3 years had a great stabilizer system which can prevent the had movement very well but had very very bad indoor photos and low light pictures.
[function=comparison; topic=image stabilization, low-light performance; polarity=both; nonFocusEntity=true]

(A.61) The Sony DSC7, pictures looked very well and nice on the Camera's 3.0 LCD, bright and crisp, but viewing these photos on computer looked bad, pictures look kinda washed out or cant put on word, may be had noise or looked like paint.
[function=comparison; topic=picture quality; polarity=both; nonFocusEntity=true]

(A.62) I fiddled with other brand cameras in a few stores and some of these have considerable lag between shots.
[function=comparison; topic=speed; polarity=negative; nonFocusEntity=true]

Although none of the preceding examples (A.60) to (A.62) includes an explicit comparative form, we mark them as `Comparison` since they all refer to another product than that primarily addressed in the review. The reviewer may name the other product explicitly, such as in examples (A.60) and (A.61), or leave it unspecified, such as in example (A.62). In example (A.60) it is clear from the sentence alone that another product is referred to, whereas in example (A.61) this is deducible only from the context.

Remember to set the *non focus entity* flag to `true` if the reviewer's comments exclusively refer to other products. Observe that in examples (A.51) to (A.55) we do not set the flag. Although other products are mentioned, the reviewer's comments primarily address aspects of the entity which is in focus of the review.

Helpful Criteria to Identify the Expression of Comparisons For detecting a comparison, indicators are quite obvious. Look out for comparative forms of adjectives as well as constructions with "more" or "less". Words such as "same", "similar", or "like" often indicate a comparison of similarity. To detect comparisons such as the ones introduced in examples (A.60) to (A.62), search for occurrences of product names or mentions of the product class in its plural form — like "cameras" or "hotels".

Setting the Sentiment Polarity Attribute in Comparisons In comparisons, one or more products are compared to the entity in focus of the review. Set the polarity attribute according to which sentiment the sentence evokes with regard to the focus entity. For instance, if the comparison lets the focus entity appear in a favorable light, set it to `positive`. In comparisons of similarity both products are typically evaluated equally positive or negative. Or no sentiment is expressed at all, such as in example (A.57).

If a sentence exclusively assesses a product that is not the focus entity, set the polarity attribute in accordance with the sentiment conveyed towards this non-focus entity.

Conclusion

Generally, a conclusion is defined as "the summing-up of an argument or text"[3]. In customer reviews it typically represents a summarizing statement of opinion and underlines the overall judgment that has been reached. Reviewers may put such a discourse segment in front of their review or, more commonly, express their final assessment in the last part of the review. In the following, we present a set of sentences that exemplify when to set the discourse function attribute to `Conclusion`.

(A.63) every thing is great i recomend this camera
[function=conclusion; topic=PRODUCT; polarity=positive]

(A.64) Overall this hotel is a gem and well worth staying in.
[function=conclusion; topic=PRODUCT; polarity=positive]

[3]Definition is taken from the Oxford Dictionary of English [349].

(A.65) Blanket statement: I am in love with this camera!
[function=conclusion; topic=PRODUCT; polarity=positive]

(A.66) I really would not recommend this camera to anyone, and i will be returning it promptly.
[function=conclusion; topic=PRODUCT; polarity=negative]

(A.67) Our whole experience at this hotel can be described as TERRIBLE.
[function=conclusion; topic=PRODUCT; polarity=negative]

Examples (A.63) to (A.67) represent the most common form of conclusive statements in customer reviews. Reviewers sum up their overall impression towards the product as a whole. As in examples (A.63) and (A.66), such a summary is often accompanied by a general recommendation or disapproval of the product.

(A.68) Everything from our large room and the hotel staff were terrific and we honestly didn't have a single complaint.
[function=conclusion; topic=room, service; polarity=positive]

(A.69) The location is small and the desk people are not the friendliest but overall a great experience.
[function=conclusion; topic=service; polarity=both]

Another common form reviewers make use of, is to substantiate their conclusive judgment by revisiting their major arguments. They stress the most critical advantages or disadvantages. The preceding examples (A.68) and (A.69) point out this manner.

Helpful Criteria to Identify the Expression of Conclusions Conclusive statements express the overall sentiment towards the product. A good hint is to look out for words and phrases similar to "overall", "all in all", "to sum up", "final note", "bottom line", "everything", "whole", etc. Other helpful indicators are the position in the review (mostly the last or the first sentences) and the formulation of a recommendation. As a conclusion reflects a decision that has been reached by the reviewer, it is closely related to the reviewer's future behavior. Look out for expressions where reviewers reason about their future behavior.

Expectation

In addition to describing their true experience, reviewers may express what they expect from a product. The information presented in such sentences does not need to reflect the real value of a product, but the individual presuppositions the reviewer has or had in mind.

(A.70) I showed up to this hotel expecting the worst, expecting horrible.
[function=expectation; topic=PRODUCT;polarity=negative; irrealis=true]

Example (A.70) illustrates the use of this discourse function. In many cases a clearly evaluative language ("worst", "horrible") is in effect neutralized by the fact that an expectation is described. If the sentence contains polar language, set the sentiment polarity attribute accordingly. Furthermore, expressing an expectation often implies that you set the *irrealis* attribute to `true`.

It is important to distinguish between cases when the reviewer mentions an expectation, but what has been expected already has become real (or not) and cases when it is unclear whether the expectation has become real or will become real. In the first case, do not mark the sentence as `Expectation`. For example in sentence (A.71) the reviewer describes a real, negatively connoted situation, he or she only adds that this situation was expected beforehand. In sentence (A.72) it is clear that the expectations have not become real as they have been "exceeded". However, in example (A.73) it is still unclear whether the expectation has become true or not, therefore it is annotated as `Expectation`.

(A.71) Plus, there were some sketchy looking people hanging around (we were ok with the punk rock kids, we expected that).
[function=fact; topic=security; polarity=negative; polarFact=true]

(A.72) Epic hotel however far exceeded all our expectations.
[function=sentiment; topic=PRODUCT; polarity=positive]

(A.73) I wasn't expecting a lot from this camera based on the price and some reviews, but because of some of the shared photos, I decided to give it a try.
[function=expectation; topic=PRODUCT; polarity=negative]

Helpful Criteria to Identify the Expression of Expectations As expectations describe things which are unclear to happen or occur, look out for phrases and grammatical constructs that imply the irrealis case. Think about whether the sentence contains modal verbs that are used in subjunctive form. Watch out for modals such as "would", "could", "should", or "might". Other words and phrases that indicate an expectation are for instance "assume", "expect", "hope", "look forward to", "presume", or "suspect".

Fact

In case a reviewer mentions a plain fact and no other discourse function is applicable, mark a sentence as `Fact`. Facts may be uttered with regard to the product, one of its aspects, or any other concept. Oftentimes a reviewer enumerates facts when he cites the specifications of the product. Take note that facts may also imply a positive or negative sentiment. In that case, mark the sentence as polar fact. The following sentences provide some examples:

(A.74) We had a few beers in the lobby bar.
[function=fact; topic=dining]

(A.75) They were able to check me in early and took care of all of my needs.
[function=fact; topic=check-in/out; polarity=positive; polarFact=true]

(A.76) This camera is essentially a Leica D-Lux 3 with a little bit different shell.
[function=fact; topic=PRODUCT]

(A.77) Partially because the zoom moves so slowly, the camera takes quite a while to power up and stand ready for use.
[function=fact; topic=zoom, speed; polarity=negative; polarFact=true]

General Remark

Reviewers may express their thoughts and ideas with regard to a general issue of the product class or a related topic without addressing the product in particular. In such cases the focus entity is referred to only implicitly, namely as a member of the product class. Annotate sentences which contain such very general comments as `GeneralRemark`.

(A.78) Many electronic items now come with cds to read the manual.
[function=general remark; topic=user manual]

(A.79) I think it's obvious that any camera in this class, with a fairly high mega-pixel count (10.1 in this case) and a small image sensor is going to have some compromises.
[function=general remark; topic=picture quality; polarity=negative]

(A.80) Remember, this is a point and shoot, not a professional grade camera, so do not expect the world from it.
[function=general remark]

In sentences (A.78) to (A.80) we exemplify the situation when a remark refers to a general property of the product class. The first two examples indirectly address a specific aspect of the product, whereas in the latter sentence a property of the product class is discussed. In the first case, set the topic attribute appropriately, in the second case, leave the topic attribute empty.

(A.81) Sitting around your monitor looking at photos just doesn't have the appeal of sitting down with the family & pouring over old photo albums.
[function=general remark]

(A.82) Of course, it's also a lot of work to get your film developed & then scan each shot to be put on the web, so it really depends on your primary goal.
[function=general remark]

The preceding examples (A.81) and (A.82) highlight the situation when a reviewer gives a general remark on a topic that is only related to the product class under consideration, implying that the topic attribute should be left empty.

(A.83) With so many options out there with electronics it is very important to go by others experience using the item and really read their opinions to see what applies to you.
[function=general remark]

(A.84) I think this is a matter of taste rather than any particular engineering issue.
[function=general remark]

(A.85) I guess bad things happen, but the good response has to be there to make everything better.
[function=general remark]

Also mark sentences as `GeneralRemark` if a remark is expressed that is "so general" that it neither refers to the product class nor any related topic. In cases such as examples (A.83) to (A.85) the reviewers names just common sense.

Lack

Earlier we introduced the polar fact attribute. A very common case when a rather factual sentence implies a negative assessment is when the reviewer mentions that a desired aspect of the product is completely missing. We introduce a specific discourse function `Lack` for such situations.

(A.86) The toiletries were missing and one towel in the bathroom.
[function=lack, topic=bathroom; polarity=negative; polarFact=true]

(A.87) The camera has no separate viewfinder, and it can be hard to see the LCD in bright sunlight, but I found I can see enough of the image to compose the picture, even in bright sun.
[function=lack, topic=screen; polarity=negative; polarFact=true]

(A.88) They don't offer free shuttle services.
[function=lack, topic=service; polarity=negative; polarFact=true]

(A.89) But I found that there is no coffee maker or any other way to boil water in the room, no complimentary drinking water.
[function=lack, topic=room amenities; polarity=negative; polarFact=true]

In examples (A.86) to (A.89) we mark the sentences as polar facts since common sense implies that the mentioned aspects and features should not be missing. So, if you can deduce that reviewers want to express their dissatisfaction about the fact that something is missing, annotate as discourse function `Lack`, mark it as polar fact, and set the polarity attribute to a negative value. If applicable set the topic attribute.

Furthermore, it is not uncommon that reviewers imply the lack of a desired feature by expressing their wish it would be present. Examples (A.90) and (A.92) highlight this case.

(A.90) I think it should come with a memory stick so it can be used right out the box.
[function=lack, topic=memory; polarity=negative]

(A.91) I would like it to be more pocket size, but the bulk is acceptable for me.
[function=lack, topic=dimensions; polarity=negative]

(A.92) I wish they'd move the controls down just a little and swap the locations of the microphone and the speaker.
[function=lack, topic=user interface; polarity=negative]

Helpful Criteria to Identify the Expression of a Lack This discourse function describes situations when something desired or commonly expected is missing. So, look out for words that directly express this situation, such as the verbs "miss" and "lack" or nouns such as "absence". It is also common to negate a word that expresses the presence of an attribute, such as "offer" in sentence (A.88). In addition, the use of the negative determiners "no" and "neither" or the cardinal number "zero" as modifier to a product aspect can serve as indicator.

Other Reviews

A further discourse function we differentiate is when reviewers cite other reviews. In fact, these are often situations when the opinion holder is not the author of the review. The subsequent examples show sentences which you should annotate as `OtherReview`.

(A.93) I bought this camera based on other's reviews on Amazon that because of it's diminutive size, one would be more apt to carry it and use it.
[function=other reviews; topic=dimensions; polarity=positive]

(A.94) No item is perfect and people are complaining about the battery life with the camera.
[function=other reviews; topic=battery; polarity=negative]

(A.95) Consumer Reports named this camera their Top Pick for Compacts!
[function=other reviews; topic=PRODUCT; polarity=positive]

(A.96) According to all the magazines, Canon is the standard right now, so we will see.
[function=other reviews]

Whenever the reviewer reports comments from other reviews or other persons in general, mark the sentence as `OtherReview`. If possible, set the topic and polarity attributes.

Personal Context

It is common that reviewers introduce themselves to the reader of their review by providing some personal context information. The following sentences exemplify this discourse function.

(A.97) I had this Canon for 3 days now, took 300 pictures, I'm not a professional nor hobbyist, just a father trying to find a good indoor camera for kids.
[function=personal context]

(A.98) We've stayed at the International House for several years on business and pleasure.
[function=personal context]

(A.99) As a music journalist, I always carry a spare blank card in case someone attempts to confiscate a memory card in one of the night clubs.
[function=personal context; topic=memory]

Authors may set their review into context by describing their own competence concerning the product. The intention is to help the reader figure out how confident to be in the validity of the reviewer's comments. For example in sentence (A.98) the reviewer implies that he or she knows the product (the hotel in this case) very well, thus trying to lend more substance to the personal assessment. Example (A.97) illustrates a similar intention.

(A.100) I've used two digital cameras prior to this one: a first- generation Powershot (nice for the price) and a company-owned Kodak DC-290.
[function=personal context]

(A.101) Just what I expected, I owned an older model and wanted to upgrade.
[function=personal context]

(A.102) Upgraded from a D-460 3x zoom for the 8x zoom.
[function=personal context]

(A.103) bought this for my 6 year old.
[function=personal context]

Competence may also be implied by naming the products previously possessed, as shown in example (A.100). In general, mark a sentence as `PersonalContext` if a reviewer names the products he previously owned, explains how he came to buy the product, or for whom the product has been bought, see examples (A.100) to (A.103).

Problem

The Oxford Dictionary of English [349] defines a problem as "a matter or situation regarded as unwelcome or harmful and needing to be dealt with and overcome". As it is quite common that reviewers describe the problems they encountered when using a product, we introduce a specific discourse function `Problem`.

(A.104) When holding down the shutter to snap a photo, the camera shuts itself off.
[function=problem; polarity=negative; polarFact=true]

(A.105) However, when shooting videos there is a very noticeable high pitched whine.
[function=problem; topic=video recording; polarity=negative; polarFact=true]

(A.106) It was guaranteed to work, but when I put in the memory card, it reads 'memory card error'.
[function=problem; topic=memory; polarity=negative; polarFact=true]

(A.107) This bldg is so old and the pipes make a horrible clanging sound in the middle of the night!

[function=problem; topic=facility; polarity=negative; polarFact=true]

(A.108) All our clothes and other belongings as well as our baby reeked of cigarette smoke.

[function=problem; topic=room; polarity=negative; polarFact=true]

Similar to the discourse function `Lack`, a problem is commonly annotated as a polar fact. Instead of lacking a desired property, a problem describes the presence of an undesired property — for example, an unwelcome situation, defect, shortcoming, or difficulty. Problem descriptions are typically provided as factual information. Applying common sense (that is, knowing that a specific situation is unwelcome), we can deduce the negativity of the utterance. In examples (A.104) to (A.108) we can easily derive that the characterized situations all imply an undesired status. Mark sentences that are similar to the ones exemplified as `Problem`, set the polarity to a negative value, and annotate them as polar fact. If applicable, set the topic attribute. If the problem description addresses the product as a whole, set the topic attribute to the special value `PRODUCT`.

(A.109) I can't really review, because it doesn't work.
[function=problem; topic=PRODUCT; polarity=negative; polarFact=true]

(A.110) Refrigerator was not working too.
[function=problem; topic=room amenities; polarity=negative; polarFact=true]

(A.111) While i waited, I discovered that not only did the toilet not work, but the shower did not either.
[function=problem; topic=bathroom; polarity=negative; polarFact=true]

Often a reviewer expresses that the product or a related aspect does not work at all. Mark those sentences with discourse function `Problem` and set the polar fact attribute to true.

(A.112) I had this same problem with the camera and had it shipped out to FujiFilm, thinking the camera was a a lemon.
[function=problem; polarity=negative; polarFact=true]

(A.113) The problem came when I tried to install the driver so I could connect it to my PC.
[function=problem; topic=software; polarity=negative; polarFact=true]

(A.114) I know this incident was beyond the hotel staff's control and I'm not looking to fault them, but it was a negative experience for my family.
[function=problem; topic=service; polarity=negative]

(A.115) One of my reasons for choosing this camera was the great video features - however, this audio problem is rendering all videos shot on this camera unwatchable.
[function=problem; topic=video recording; polarity=negative]

Opposed to the previous examples in sentences (A.112) to (A.115), the reviewers directly speak of a problem. They either just name that a problem exists, such as in examples (A.112) to (A.114), or specify the problem in more detail, as outlined in example (A.115) ("audio problem"). Furthermore, observe that all these examples do not express a polar fact.

Helpful Criteria to Identify the Expression of Problems Because identifying the description of a situation or behavior as problem is based on applying commonsense knowledge, we cannot give specific hints. Ask yourself the question whether a description describes an undesired fact. Does the reviewer inform that something is not working? Some words that may indicate a problem are: "problem", "issue", "difficulty", "deficit", "obstacle", "trouble", "limitation", "failure", "error".

Purchase

A further common discourse function is when the reviewer talks about the purchase of the product. We therefore denote this function as `Purchase`.

(A.116) Recently was looking through eBay and noted that G2's had fallen in price so that you got buy full kit G2's for about $50.
[function=purchase; topic=price]

(A.117) I bought this camera at Sam's Club for about $225.
[function=purchase; topic=price]

(A.118) Amazon.com did an excellent job getting the camera and memory sticks to the house on Christmas Eve.
[function=purchase; polarity=positive]

(A.119) NOTE: I ordered the camera with a next day delivery.
[function=purchase]

(A.120) I purchased this camera last August, and I am still extremely happy with my purchase.
[function=purchase; topic=PRODUCT; polarity=positive]

Annotate a sentence as `Purchase` in case the reviewers mention where or when they bought the product or how much they paid for it. If reviewers describe aspects of the delivery (sentences (A.118) and (A.119)), also mark the sentence as `Purchase`. Note that in example (A.120) they refer to their purchase of the camera, but also evaluate the product. Mark such sentences as `Purchase` instead of `Sentiment` and set topic and polarity attributes appropriately.

Requirement

Often, when introducing a review, authors point out their requirements with regard to the product prior to buying it. They may explain which reasons made them to purchase the product. You should annotate such cases as `Requirement`. In the following we present some examples sentences:

(A.121) My wife wanted something cheap to replace 35mm film disposable cameras.
[function=requirement; topic=price]

(A.122) I wanted a camera that used the same AA batteries as my Sony DSC-W5 and that did not employ a spare battery that costs $30 or more.
[function=requirement; topic=battery]

(A.123) That was one of the main reasons I chose it - I can't stand the almost credit card sized cameras, but I wanted something I could carry around in my purse.
[function=requirement; topic=dimensions]

(A.124) The Leica lens, 10x optical zoom (my friends were very jealous), and a very long list of other features is what attracted me.
[function=requirement; topic=optics]

(A.125) she didn't want a 'baby camera'.
[function=requirement]

Helpful Criteria to Identify the Expression of Requirements Very common words and phrases to express a requirement are "want", "need", "require", "demand", "look for", etc. As reviewers talk about the requirements prior to buying the product, the grammatical tense of these verbs is typically the past.

SectionHeading

Reviewers use headings to structure their review. We identified three major types of headings typically used in customer reviews. Types are distinguished with regard to the purpose of the section they introduce:

- Pros: A heading of this type introduces a section where the reviewer enumerates the positive aspects of the product.
- Cons: A heading of this type introduces a section where the reviewer enumerates the negative aspects of the product.
- Topic: Using this type of heading, the reviewer indicates that the following section sets focus on a very specific aspect of the product.

In the annotation scheme we distinguish the proposed types as follows: Mark all headings with the discourse function attribute set to `SectionHeading`. For headings of type *Pros* additionally set the

polarity attribute to `positive`. Analogously, set the sentiment polarity to `negative` for section headings of type *Cons*. Set the topic attribute for headings of type *Topic* to one of the predefined topics or leave it empty if no topic matches. If the reviewer uses a heading that implies that the following section addresses the (dis)advantages of a specific topic, set the polarity and topic attribute accordingly, such as in example (A.129). In case the section heading is of any other type (see example (A.130)), only set the discourse function attribute to `SectionHeading` and leave other attributes empty. The following examples give you an impression:

(A.126) Let me start with the positives.
[function=section heading; polarity=positive]

(A.127) Things to Watch For:
[function=section heading; polarity=negative]

(A.128) Size and Weight:
[function=section heading; topic=Dimensions]

(A.129) Service: There were a series of incidents which lead me to give this only 1/5:
[function=section heading; polarity=negative; topic=service]

(A.130) RANDOM COMMENTS
[function=section heading]

Sentiment

Naturally, the discourse function named `Sentiment` is the most representative one for customer reviews. Choose this function if the author expresses an explicit opinion or feeling and no other of the more specific discourse functions fits. Examples (A.131) to (A.135) outline situations when it is indicated to use the discourse function `Sentiment`.

(A.131) It's very easy to use, I never had to read the manual.
[function=sentiment; topic=ease of use; polarity=positive]

(A.132) Image quality is par excellence.
[function=sentiment; topic=picture quality; polarity=positive]

(A.133) The Washington Monument and Air/Space Museum are within walking distance, so that was nice, too.
[function=sentiment; topic=location; polarity=positive]

(A.134) We were warmly greeted by Frances who checked us in without any problems.
[function=sentiment; topic=check-in/out; polarity=positive]

(A.135) The front lobby from the Boulevard is cold, souless and dark.
[function=sentiment; topic=facility; polarity=negative]

Recall that the expression of sentiment is modeled independent of the topic relevance. A reviewer may express his opinions towards other entities or just mention his general feelings. You should also mark these sentences with the function `Sentiment`. Sentences (A.136) and (A.137) exemplify this aspect:

(A.136) I didn't like the atmosphere in the city.
[function=sentiment; polarity=negative]

(A.137) I really enjoyed the few days off.
[function=sentiment; polarity=positive]

Summary

We further provide a special discourse function that represents the case when a reviewer enumerates advantages and disadvantages in a single sentence. Often, such a sentence is not a grammatically correct one, but merely comes as a comma separated list of pros and cons. Mark these kind of sentences with the discourse function `Summary` and list the mentioned topics. In most cases you need to set the polarity attribute. A few examples are provided in the following:

(A.138) Large room, comfortable mattress and lots of pillows to choose from, clean, microwave and fridge with all rooms, nice mid-size bathroom, free internet in the lobby, standard continental breakfast except it includes a waffle maker and mix.
[function=summary; topic=bed, cleanliness, room amenities, bathroom, internet, breakfast; polarity=positive]

(A.139) Compact, great pictures, low cost.
[function=summary; topic=dimensions, picture quality, price; polarity=positive]

(A.140) I love the compact size, I love the image quality, I love the large view finder screen, the ability to capture video and most of all I love the ability to set my own controls.
[function=summary; topic=dimensions, picture quality, screen, video recording, settings; polarity=positive]

Usage

Mark a sentence with discourse function `Usage` if reviewers describe a concrete situation when they are using the product. They may potentially comment on their experience.

(A.141) I traveled for three weeks through Europe and this camera held up splendidly.
[function=usage; topic=PRODUCT; polarity=positive]

(A.142) I took it to a baseball game and It was very easy to use, even in low light images were clear and consise.
[function=usage; topic=ease of use, low-light performance; polarity=positive]

(A.143) On a recent vacation to Nova Scotia I took a lot of wildflower pictures and they turned out fantastic.
[function=usage; topic=picture quality; polarity=positive]

(A.144) Used this camera at a wedding a few days after receiving it.
[function=usage]

Mark a sentence with discourse function `Usage` if reviewers talk about concrete situations when they use the product. Often reviewers report situations they experienced in the past, such as highlighted in examples (A.141) to (A.143). If reviewers evaluate their experience with the product as a whole or with one of the aspects, set the topic and polarity attributes appropriately.

(A.145) I use it for recording family events and occasional scenery.
[function=usage]

(A.146) I use it all the time when I go snorkling!
[function=usage]

(A.147) Since I opened the box I've carried it EVERYWHERE with me!
[function=usage]

(A.148) My main use for this camera will be for taking pictures of properties as my wife and I start looking for a new house.
[function=usage]

Instead of reporting past experiences and naming a concrete event, it is also common that reviewers write about their typical usage or express how they intend to use the product in general. Also mark these sentences as `Usage`. In these cases normally no sentiment is expressed and also the topic remains unspecified. Sentences (A.145) to (A.148) exemplify this situation.

Usage versus Advice and Problem The discourse functions `Usage`, `Advice`, and `Problem` may sometimes be difficult to distinguish. The following examples show cases when you should not annotate as `Usage`:

(A.149) I tried to use the different flash modes requested for night, but I still couldn't produce any quality night photos.
[function=problem; topic=flash; polarity=negative; polarFact=true]

(A.150) I usually take the pictures with the viewfinder which is a good thing because it saves your batteries.
[function=advice; topic=battery]

(A.151) I finally realized I had to reset the minimum shutter speed to one second (its longest setting), and then I was able to take pictures indoors without flash, at least as long as there was a fair amount of ambient light.
[function=advice; topic=settings, low-light performance]

If a negative experience is mentioned, the sentence is likely to contain a problem description. For instance, in example (A.149) the reviewer explains the experience with different flash modes and the undesired outcome. Annotate such sentences as `Problem`. In the guidelines for the function `Advice`, we already pointed out that descriptions of best practices with regard to the product should be annotated as `Advice`. If a reviewer talks about the usage and implies a best practice, mark the sentence as `Advice`. Sentences (A.150) and (A.151) exemplify this situation.

Category OTHER

If you cannot associate a sentence with one of the predefined discourse functions, set the discourse function attribute to the value `OTHER`. The following examples give you an impression, which kind of sentences may not fit any of the functions. These can be incomplete or grammatically incorrect sentences, such as in examples (A.152) and (A.153), rhetorical questions, such as in example (A.154), or greetings, such as in example (A.155).

(A.152) well mostly anyway
[function=OTHER]

(A.153) (Knock wood) LOL
[function=OTHER]

(A.154) Am I babbling?????
[function=OTHER]

(A.155) Good Luck, Best Wishes, S.M.
[function=OTHER]

A.2.8. Annotation Tools and Annotation Process

The sentence level annotation scheme is implemented by means of a simple W3C XML-Schema[4] definition. A sentence level annotated corpus consists of a single, huge XML document. Our XML-

[4] http://www.w3.org/TR/xmlschema-0/

Schema defines the three XML entities *corpus*, *review*, and *discourse-segment*. A corpus entity is the root element of the document and has a single attribute that is used to name the corpus (e.g., "hotel-corpus"). A corpus is composed of arbitrary many review elements which in turn cover one or more discourse-segment elements. The review entity provides meta data about a review, such as the URL it has been originally fetched from or the overall numerical rating which has been provided by the reviewer. The textual content of a review is split into sentences and each sentence is covered by a (child-) discourse-segment element — that is, the textual content of a discourse-segment element is exactly one sentence of a review. The element's attributes directly correspond to the attributes of the *DiscourseSegment* annotation type. In effect, we can annotate a sentence by setting the attributes of an XML entity, namely the discourse-segment element. The described setup is depicted in Fig. A.1.

```
<corpus corpusName="hotel-reviews">
  <review url="http://[...]" overallRating="4.0" documentID="hotel-0001">
    <title>...</title>
    <discourse-segment discourseFunction="personalContext">...</discourse-segment>
    ...
    ...
    <discourse-segment discourseFunction="conclusion" sentimentPolarity="positive">...</discourse-segment>
  </review>
  ...
  ...
  <review url="http://[...]" overallRating="1.0" documentID="hotel-0300">
    ...
    ...
  </review>
</corpus>
```

Figure A.1.: An XML document storing sentence level annotations.

The process of annotating a corpus on the sentence level is as follows: First, the set of sampled review documents is processed by an automatic sentence splitter (we use the splitter provided as part of the Stanford CoreNLP tools). We convert the results to the described XML format and receive a single, huge XML document that represents the corpus. The human annotator annotates each sentence in accordance to the annotation scheme and the guidelines by setting the appropriate attributes of the discourse-segment element. The process of setting attributes is much relieved by using the auto-completion feature of an appropriate XML editor. In case the automatic sentence splitter detected erroneous sentence boundaries, the annotator can easily correct the error within the XML document. That is, besides annotating the review dataset, the annotator's secondary goal is to ensure a correct sentence splitting of the corpus.

A.3. Annotation Guidelines for the Expression Level Model

A.3.1. Basic Instructions

Your main task is to identify explicit expressions of opinions and to mark as well as relate the linguistic constituents that make up these expressions. Each customer review you are going to annotate is contained in a single document. A document is split into a sequence of sentences and each sentence is presented in a single line of a document. During annotation, consider every single sentence as an independent isolated unit. Except for the sentiment reference annotation, no relation between annotations must cross sentence boundaries. Shifters and targets of a sentiment expression must occur in the same sentence. A sentiment reference that represents a (pronoun) co-reference relation can occur in another sentence than the referencing sentiment target. Follow these basic instructions when annotating a sentence on the expression level. More detailed instructions with examples are provided as part of the annotation guidelines:

1. Read the sentence with care and identify sentiment expressions and their targets.

2. For each **sentiment target**, check if it is, or in case of a pronoun, refers to a valid product aspect (with respect to the focus entity of the document). If yes, create a sentiment target annotation and associate it with the corresponding span of text. If the target is the mention of a product name, set the *isProductName* attribute to `true`, otherwise leave it empty. In case the target is not a direct mention of a product aspect but implies one, set the *isImplicit* attribute to `true`, otherwise leave it empty.

3. If a previously identified sentiment target is a pronoun, determine the referenced entity and create a new **sentiment target reference** annotation (if not already existent for this entity). Take note that the reference may occur in another sentence. In the corresponding target annotation point to the reference by setting the *pronounReferenceID* attribute to the annotation ID of the sentiment reference annotation. In case the pronoun is exophoric (i.e., it refers to an entity not explicitly mentioned in the text), set the *isExophoric* attribute to `true`, otherwise leave it empty.

4. For each identified **sentiment expression**, create a new annotation comprising the corresponding span of text. Relate associated sentiment target annotations by adding the individual annotation IDs to a comma separated list in the *sentimentTargetIDs* attribute. Determine whether the polarity of the expression in this specific context is predominantly positive or negative and set the *sentimentPolarity* attribute accordingly. Take note that the polarity might be dependent on the referenced sentiment target(s). However, when setting the polarity, assess the expression in isolation and do not consider any associated sentiment shifters. Also determine the sentiment intensity in isolation. Set the *sentimentIntensity* attribute to either `strong` or `average`. If you find that the polarity of the expression is dependent on the sentiment target(s), set the *isTargetSpecific* flag to `true`, otherwise leave it empty.

5. For each identified sentiment expression, determine all **sentiment shifters** that modify the expression in one of the six predefined modes. For any shifter, create a new sentiment shifter expression (if not already existent) that comprises the corresponding span of text. Depending on the function of the shifter, set its *type* attribute to either `negator`, `amplifier`, `downtoner`, `solidifier`, `softener`, or `neutralizer`. Relate a shifter annotation to its referring sentiment expression by adding its annotation ID to a comma separated list in the *sentimentShifterIDs* attribute of the expression type.

6. For each explicitly mentioned product aspect that is not annotated as a sentiment target create a new **product aspect mention** annotation, comprising the corresponding span of text. Only annotate aspects that are relevant for the current product domain and never annotate pronouns or implied product aspects. If the product aspect mention refers to a product name, set the *isProductName* flag to `true`, otherwise leave it empty.

A.3.2. General Remarks

Text Spans

For each annotation type of the expression level model, it is your task to determine a corresponding text span in the document. The following rules and guidelines are generally applicable for all annotation types:

- Always annotate the minimum span of text that represents the linguistic constituent captured by the specific annotation type.

- Never annotate a text span across a sentence boundary.

- A text span must not start or end with a whitespace character.
- Do not include determiners in your annotation unless they are part of a longer phrase that constitutes the relevant linguistic feature. An annotated text span must never begin with a determiner. The rule applies to definite and indefinite articles as well as to quantifiers and demonstrative pronouns that are used as determiners.
- Do not include punctuation marks in your annotations. For example, leave out any periods, question marks, exclamation marks, commas, parentheses, quotation marks, or apostrophes.
- Do include apostrophes if they are part of a contraction that is relevant for your current annotation. In particular, annotate contracted negations such as "doesn't", "don't", "isn't", etc. by only marking the *n't* part of the contracted form and leaving out the prefixes ("does", "do", "is", etc.).
- Consider misspellings as if they were correctly spelled. For instance, if the words *digital camra* are the target of a sentiment expression, create a new sentiment target annotation comprising the whole compound, irrespective of the misspelling of the word *camera*.

Annotation IDs and Pointer

Whenever you create a new annotation, the annotation tool generates a new ID which uniquely identifies the annotation. The IDs are created sequentially and are unique over all annotation types. You use the IDs to relate one annotation to another one — for example, when you relate a sentiment shifter to a sentiment expression by setting the *sentimentShifterIDs* attribute. Double-check that you point to an existing and semantically reasonable ID (e.g., do not point to a sentiment shifter annotation in the *pronounReferenceID* attribute of the sentiment target). The annotation tool chain verifies that pointers are set in a correct manner and highlights potential mistakes that you have to fix manually.

Graphical Notation

To describe the constituents of the expression level model, we use the graphical notation that we introduced in Section 4.3.

A.3.3. Guidelines for the Sentiment Expression Annotation

The sentiment expression annotation can be regarded as an anchor when annotating a text passage. It refers to at least one target and is optionally modified by one or more sentiment shifters. A sentiment expression is that part of the surface text in which the actual opinion becomes manifest. This can be a single word or can be a whole phrase. All major parts of speech (adjectives, verbs, nouns, or adverbs) may act as a sentiment expression. To identify a sentiment expression, consider which words make up the opinion — that is make the text passage distinct from pure factual information. Refer to the guidelines presented in Appendix A.2.2, which provide hints on how to identify the textual expression of an opinion in general.

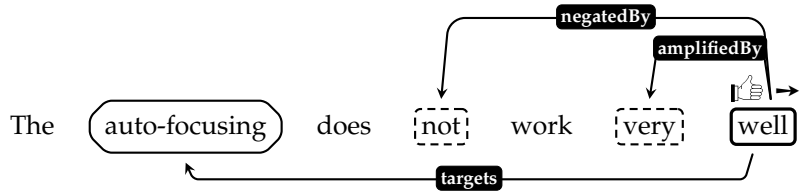
Only consider sentiment expressions which have a clear sentiment target. For example, do not annotate anything in a sentence such as "I really feel happy.". Although the word "happy" expresses a sentiment, it lacks a precise target. However, in a sentence such as "I really feel happy with this camera.", a clear target ("camera") exists and you have to create the appropriate annotations. If the target is not a product aspect or is not relevant in terms of the current product domain, also leave out the annotation. For example, if you are annotating a review that evaluates a digital camera, do not annotate passages where the reviewer talks in favor of his last vacation (where he used the camera).

Besides pointers to sentiment targets and shifters, the annotation type takes two mandatory attributes that you have to determine whenever you create a sentiment expression annotation:

Sentiment Polarity Attribute

The sentiment polarity attribute refers to the *target-specific prior polarity* of the word or phrase that you annotate. Decide whether the sentiment expression conveys a predominantly `positive`, `negative`, or `neutral` sentiment to its target. When making this decision, consider the sentiment expression in isolation from any sentiment shifter. In the following we make use of the graphical notation as introduced in Section 4.3.

(A.156)



In example (A.156) the author expresses a negative opinion on the auto-focus. In this case, the opinion is evoked by an adverb ("well"), which we mark as sentiment expression. Observe that we set the polarity attribute to `positive`, although the overall contextual polarity is clearly negative. When setting the polarity, we do not care about the negator "not" or the amplifier "very".

Sentiment Intensity Attribute

Similar to the prior polarity, a sentiment bearing phrase exhibits an immanent sentiment intensity. Your task is to classify the intensity of an expression into the two levels `average` and `strong`. So, the basic decision you have to take is whether an expressions's intensity is strong or not. When doing so, you can use the following list of words (presented in no specific order) as reference:

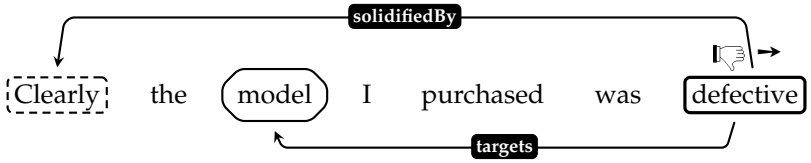
- Average: ok, satisfying, sufficient, ample, adequate, fair, acceptable, good, bad, to like, to dislike, great, poor, nice, pleasing, decent, easy to use, problem, disadvantage, ...
- Strong: excellent, superb, terrific, horrible, stunning, outstanding, wonderful, terrible, lousy, awful, fantastic, to love, to hate, nightmare, masterpiece, ...

Take note that setting the polarity attribute to `neutral` always implies a sentiment intensity of `average`. Neutral sentiment expressions are not gradable with respect to their intensity. The most common intensity value is `average`; only annotate an expression as `strong` if it clearly stands out with regard to other expressions. For example, we do not classify the word "great" as exhibiting a strong intensity.

Target Specific Sentiment Expressions

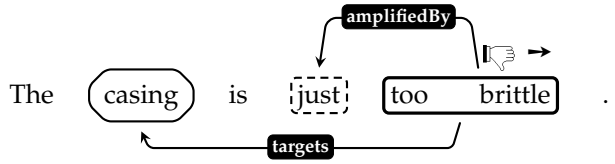
Some words or phrases only convey an opinion when used in conjunction with a specific target. Also the polarity may depend on the target. Such words are mostly adjectives. For example the word "long" alone has no prior polarity and usually transports factual information. However, when used in the context of the product aspect "battery life" it clearly has a positive connotation. Applying commonsense knowledge, we can easily deduce that a long battery life is generally considered a positive property of a digital camera. On the other hand, in the context of a property such as "shutter lag time" the adjective "long" can be negatively connoted.

If you encounter an expression (mostly adjectives) that does not possess an inherent sentiment, but unfolds its evaluative connotation only in the context of a specific target, set the *isTargetSpecific* flag to `true`. In the following, we present some examples that illustrate the use of the different attributes and the identification of sentiment expressions in general:



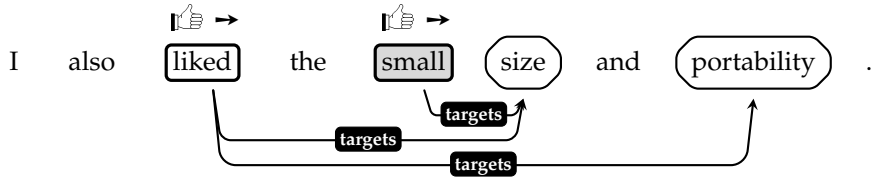
(A.157)

Example (A.157) shows a very simple case, where the adjective "defective" can be easily identified as the sentiment bearing phrase. The author underlines his impression with the word "clearly". We mark it therefore as solidifying sentiment shifter.



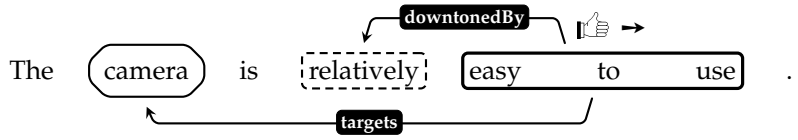
(A.158)

When reading the sentence of example (A.158), it might by unclear, whether to annotate the single word "brittle" as sentiment expression or the combination "too brittle". Although the word "brittle" alone exhibits a prior negative sentiment towards the "casing", the word "too" is part of the complete sentiment expression. We therefore annotate the combination. This is a very common case and similar expressions should be annotated analogously. Take note that "too" is not a sentiment shifter.



(A.159)

This example highlights the fact that a single sentiment expression ("liked") may target multiple product aspects and that one aspect ("size") may be targeted by multiple sentiment expressions. The adjective "small" is marked as target specific sentiment expression because a small size of a camera is generally considered as a positive attribute. Also take note that the sentence exemplifies a case where sentiment is conveyed by a verb ("liked").



(A.160)

We use this sentence to exemplify a case where the sentiment is expressed as part of a longer phrase. Although the actual polarity of the expression solely relies on the word "easy" (e.g., exchanging it with "difficult" would shift the polarity), the correct sentiment is conveyed only by the complete phrase. For example, think of a the sentence such as "The camera is easy to drop.". If we only would annotate "easy" as sentiment expression, it would be indistinguishable from example (A.160) on the annotation level. The word "easy" would have different polarities although the target is the same. Constructions of the form *adjective + to + verb* are quite common in reviews. Such phrases and similar ones should be annotated as illustrated by this example.

(A.161)

In example (A.161) we point out that reviewer may also use nouns (here: "shame") to express his sentiment.

A.3.4. Guidelines for the Sentiment Target Annotation

Sentiment targets always refer to product aspects that are relevant in terms of the examined product domain. A target may become manifest as a single word, a compound noun, or a complex phrase. The part of speech is variable, but most commonly nouns function as sentiment targets.

Compound Nouns as Sentiment Target

Whereas annotating a single-word target is straightforward, see examples (A.156) to (A.161), determining the correct text span of a compound noun can be more difficult. The general rule is to annotate the minimum span that makes up the target. The following guidelines help you to correctly and consistently identify multi-word sentiment targets:

- **Type versus attribute**: Most targets are formed by nouns. In case such a noun is modified by an adjective, you have to decide whether it is part of the target or not. In the examples, "digital camera", "compact camera", or "ultra-compact camera" all adjectives belong to the target, whereas in the compounds "versatile camera", "collapsible camera", or "professional camera" the modifiers are not part of the target. To make this distinction, determine whether the adjective specifies a type of the modified entity or describes an attribute of the entity. In the first case, include the adjective, in the latter case omit the adjective. A good rule of thumb is also to check whether the adjective is gradable or not. Gradable adjectives typically describe attributes and therefore should be omitted.

- **Numerical and color attributes**: Do not include numerical or color attributes in sentiment targets unless they form an integral part of a term. For instance, do not include "10x" in "10x optical zoom" or "red" in "red camera design". But include "red" in "red eye reduction".

- **Compounds with prepositions and conjunctions**: Compounds can be more complex and include other parts of speech than nouns and adjectives. Most commonly, complex compounds include prepositions and conjunctions, such as in "digital point-and-shoot camera", "shot-to-shot speed", or "out-of-focus shot". Analogously to the first rule, you should include complex modifiers if they specify a particular type of an entity.

- **Linking prefixes with coordination**: In case a coordination such as "and" or "or" is used to link relevant prefixes of an entity, annotate the complete phrase as a sentiment target. An example is given in sentence (A.162). Here, by using a conjunction, both prefixes "auto" and "manual" are linked to the noun "mode". As both describe a specific type of mode, they are relevant and thus, we annotate the whole phrase as a sentiment target.

(A.162) The auto and manual mode are both absolutely great .

Complex Phrases as Sentiment Target

In addition to compound nouns, more complex phrases may constitute a sentiment target. Typically, such a phrase represents a description of a certain product behavior or is a periphrasis of a product aspect. The following examples point out when to annotate a complex phrase as sentiment target:

(A.163) I really like the way the camera renders clouds .

In this example the reviewer expresses his appreciation regarding a very specific behavior of the product. No explicit term exists that could be used. The reviewer must describe the behavior in a longer phrase. In all cases a reviewer evaluates a certain behavior of a product, annotate the whole phrase that describes the behavior, but do not include any preceding determiner.

(A.164) I love the ability to check out late .

Here, the reviewer formulates a periphrasis of the product aspect "late check-out". Again, annotate the whole phrase and leave out any preceding determiner. As for the first example, you must set the *isImplicit* attribute to `true`; a periphrasis implicitly refers to a nominal product aspect.

(A.165) The hotel has a wonderful lobby with big sofas .

Do not include clauses that provide additional, subordinate information on a named product aspect. In this example, only the word "lobby" is the target instead of the whole phrase "lobby with big sofas". Take note that we annotate the word "hotel" as a product aspect mention.

Verbs as Sentiment Target

Also verbs may represent a valid sentiment target. For example, in the sentence "The camera focuses fast.", using the verb "focuses" implies that it is the focus which is fast. Substituting the verb with another verb, such as "is", "works", or "zooms", would alter the actual target of the sentiment expression. In the first two cases, the target would be the camera and in the third case the reviewer would implicitly refer to the zoom. Whenever a verb is directly derived from a nominal product aspect (e.g., to focus, to zoom, or to design) and is target of a sentiment expression, mark the verb as sentiment target. Verbs that are not directly derived from a product aspect may not function as a target. Verbs that are annotated as sentiment target always must be marked with the *isImplicit* flag set to `true`.

Implicit Sentiment Targets

Whenever you annotate a text span as sentiment target that does not explicitly represent a nominal product aspect, you must set the *isImplicit* attribute to `true`. As mentioned in the previous sections that is generally the case when the reviewer uses a periphrasis describing an aspect or a verb that implies a specific aspect. Do not set the flag if the sentiment target is a pronoun or refers to a product name.

Product Names as Sentiment Targets

Despite nominal product aspects and implicit mentions, a product name can constitute a valid sentiment target. We want to differentiate between these classes of sentiment targets and therefore introduce the flag *isProductName*. Whenever a sentiment expression targets a product name, the attribute must be set to `true`. A product name may refer to the name of a producer (e.g., "Canon", "Samsung", or "Sony"), to a specific model (e.g., "EOS 550D", "ES80", or "DSC-W570B"), or to a combination of both (e.g., "Canon EOS 550D").

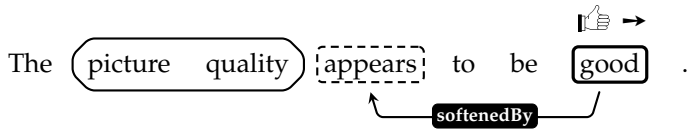
A.3.5. Guidelines for the Sentiment Shifter Annotation

A sentiment shifter modifies the polarity, intensity, or the degree of certainty of a sentiment expression. It is existentially dependent on such an expression; that is, a sentiment shifter annotation must be referenced by at least one sentiment expression annotation. We model a many-to-many relation between shifters and sentiment expressions: A sentiment expression may refer to zero or more shifters and a shifter may be referenced by one or more sentiment expressions. This many-to-many relation is implemented by allowing a comma separated list (of pointers to shifter annotations) in the *sentimentShifterIDs* property of the sentiment expression annotation type.

The sentiment shifter annotation takes the single attribute *type* that defines which characteristic of a related sentiment expression is modified. Setting the type to `amplifier` or `downtoner` indicates that the intensity of a sentiment expression is either increased or decreased. Examples (A.156) and (A.158) contain amplifying shifters, whereas example (A.160) illustrates the use of a downtoner. The majority of amplifying and downtoning shifters refer to adverbs such as "very", "extremely", "really", "greatly", "hardly", "relatively", "little", or "barely".

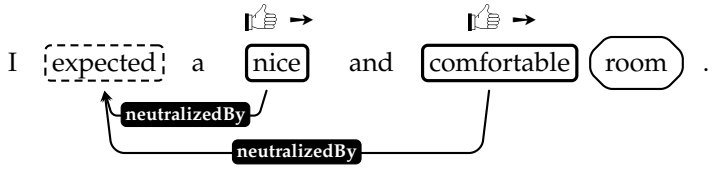
Solidifying and softening shifters affect the degree of certainty evoked by a sentiment expression — that is, the commitment the author attributes to his opinion. Take note that a priori, a sentiment expression does not exhibit any specific certainty degree. This attribute is solely induced by solidifying or softening sentiment shifters. Using a solidifier, the author indicates that his certainty regarding the opinion is strong, whereas using a softener indicates a weak certainty. Both types of shifters may be expressed by single words or longer phrases. An example for a solidifying shifter is given in sentence (A.157). Here, the word "clearly" increases the author's commitment towards his sentiment expression. An example for a softening shifter is presented in (A.166). Here, it is the word "appears" that reduces the certainty of the author. The reader is not sure whether the picture quality is good or just appears to be good.

(A.166)



A neutralizing sentiment shifter refers to words or phrases that set the the sentiment expression into an irrealis context, effectively "nullifying" the prior polarity of the related expression. Typical cases include the use of subjunctive forms (indicating hypothetical or unlikely events) or the use of conditional forms (indicating that an event is dependent on a condition). Furthermore, words and phrases that express wishes, requirements, hopes, or expectations of the author often act as neutralizers. An example for a neutralizing sentiment shifter is given in (A.167). Here the reviewer expresses an expectation, setting the positive evaluation of the room in an irrealis context.

(A.167)



The most common sentiment shifter is a negation. It predominantly influences the sentiment polarity by shifting the prior polarity of the sentiment expression (e.g., from `positive` to `negative`). However, negation also may influence the sentiment intensity and need not necessarily flip polarity. Typical words that indicate negation are "not", "no", "never", "none", "nothing", or "neither". Besides these adverbs, other parts of speech may function as negators. For instance, verbs such as "avoid" or "prevent" can behave as a negator. Often the negative particle "not" is contracted to "n't". If that is the case, only annotate the text span covering the contracted form of the particle and leave out the prefix (e.g., only mark "n't" in "isn't"). Do not mark negations that point out the non-existence of an entity.

A.3.6. Guidelines for the Sentiment Target Reference Annotation

A pronoun reference is annotated by means of a sentiment target reference annotation. This annotation type takes no attributes; it is a simple marker. You should only annotate a pronoun reference if the referring pronoun is labeled as a sentiment target. The relation between reference and referrer is established via the *pronounReferenceID* attribute of the sentiment target annotation type. Most commonly, a reference is *anaphoric* (i.e., is mentioned prior to the pronoun). However, a target reference may also be *cataphoric*. We also consider a special case when the pronoun refers to a target that is not part of the text. In such a case the reference is called *exophoric*. Use the *isExophoric* flag in the sentiment target type to capture this situation.

A.3.7. Guidelines for the Product Aspect Mention Annotation

The product aspect mention annotation type stands on its own and is not related to any other annotation type of the expression level annotation scheme. It is used to capture all mentions of product aspects that are not addressed by a sentiment target annotation — that is, mentions of aspects which are not target of any opinion.

With this additional annotation we only capture nominal and named mentions of product aspects. We do not consider any implicit mentions through complex phrases or verbs. With regard to compound nouns, the same rules apply that we set up for sentiment target annotation type. Also handle the *isProductName* attribute in the same way as for sentiment targets.

A.3.8. Annotation Tools and Annotation Process

As the expression level annotation scheme is more complex (annotators need to mark individual words/phrases and are required to interlink created annotations), editing an XML document (as done for the sentence level annotations) is not an option. We therefore use a dedicated annotation tool: We implement the expression level annotation scheme by means of *GATE*[5] *annotation schemata*. GATE requires to define a separate W3C XML-Schema for each annotation type. The definition language is actually a subset of XML-Schema. It allows to distinguish optional and required attributes as well as to restrict the valid range of attribute values. The advantage of defining GATE annotation schemata is that the *GATE Schema Annotation Editor* is aware of valid attribute values and thus can present drop-down menus or lists of possible entries so that the annotator is relieved from the task of typing in attribute values. With GATE annotation schemata it is not possible to model relations between annotation types. In consequence, the GATE Schema Annotation Editor cannot automatically verify the validity of created links. The annotation tool also does not support for graphical editing of relations. The annotator is forced to the error-prone task of linking individual annotations by typing in the corresponding annotation IDs in the appropriate attribute fields. For this reason, we created a tool for post-processing annotated documents. It identifies and highlights invalid annotations (e.g., an illegal link from a sentiment target to a sentiment shifter annotation or an isolated sentiment target annotation that is not linked to any sentiment expression), so that annotators can easily revise incorrect annotations. Figure A.2 depicts a screen shot of the GATE Schema Annotation Editor while annotating a sentiment expression. It gives an impression about the annotation procedure.

Opposed to the annotated sentence level corpora, which are stored as single huge XML documents, the expression level corpora are composed of collections of individual documents. However, the textual content (i.e., the covered set of customer review documents) is identical in corresponding corpora.

Take note that we use the GATE framework only for the purpose of creating the hand-annotated expression level corpora. Machine-based, automatic annotators and the corresponding evaluation

[5] http://gate.ac.uk/

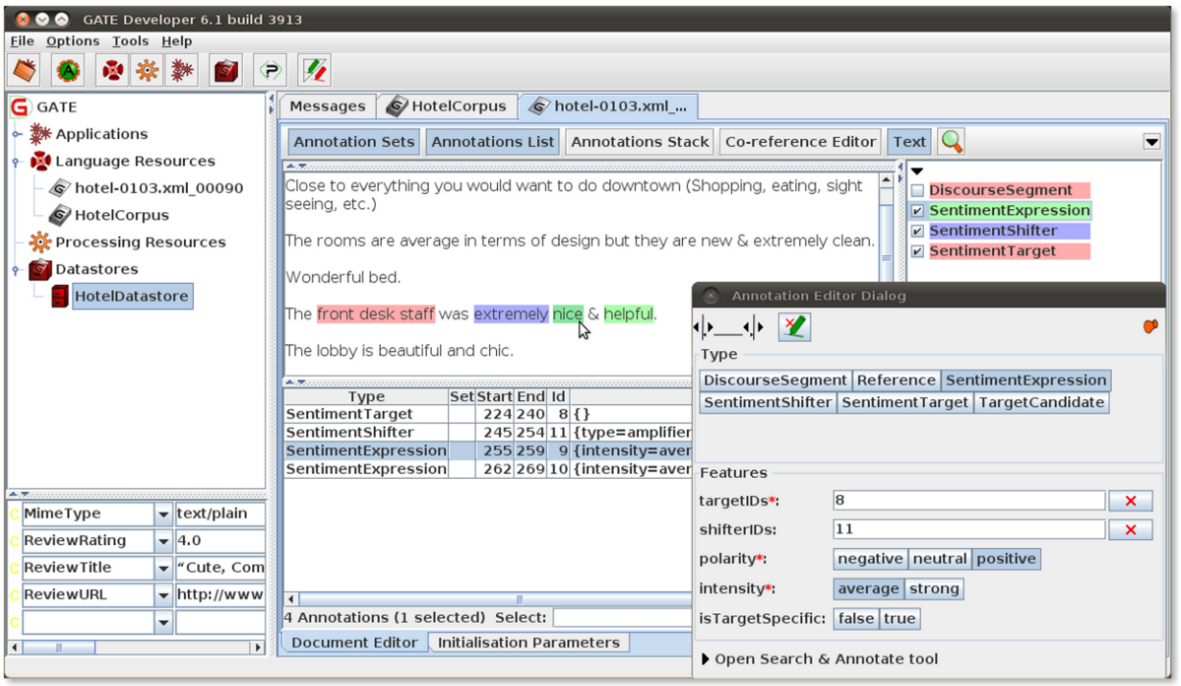


Figure A.2.: Screenshot of the GATE Schema Annotation Editor while creating a sentiment expression annotation.

systems are implemented within the UIMA framework. At least at the time of creating the annotated corpora, the GATE Schema Annotation Editor was far superior to its counterpart (*CAS Editor*) in the UIMA framework. We implemented and use tools that allow us to transform annotated GATE documents to UIMA compatible documents and vice versa.

B. Corpus Analysis Data

rating	polar	¬polar	positive	negative	neutral	both
1	275 (62.93%)	162 (37.07%)	37 (8.47%)	226 (51.72%)	4 (0.92%)	8 (1.83%)
2	205 (63.47%)	118 (36.53%)	48 (14.86%)	141 (43.65%)	1 (0.31%)	15 (4.64%)
3	384 (72.32%)	147 (27.68%)	139 (26.18%)	186 (35.03%)	32 (6.03%)	27 (5.08%)
4	771 (74.71%)	261 (25.29%)	526 (50.97%)	184 (17.83%)	24 (2.33%)	37 (3.59%)
5	905 (78.49%)	248 (21.51%)	799 (69.30%)	80 (6.94%)	4 (0.35%)	22 (1.91%)
all	2540 (73.07%)	936 (26.93%)	1549 (44.56%)	817 (23.50%)	65 (1.87%)	109 (3.14%)

(a) hotel corpus

rating	polar	¬polar	positive	negative	neutral	both
1	178 (67.94%)	84 (32.06%)	16 (6.11%)	153 (58.40%)	1 (0.38%)	8 (3.05%)
2	98 (67.12%)	48 (32.88%)	18 (12.33%)	72 (49.32%)	1 (0.68%)	7 (4.79%)
3	196 (52.69%)	176 (47.31%)	66 (17.74%)	117 (31.45%)	5 (1.34%)	8 (2.15%)
4	653 (59.36%)	447 (40.64%)	398 (36.18%)	219 (19.91%)	19 (1.73%)	17 (1.55%)
5	1066 (66.09%)	547 (33.91%)	871 (54.00%)	155 (9.61%)	20 (1.24%)	20 (1.24%)
all	2191 (62.73%)	1302 (37.27%)	1369 (39.19%)	716 (20.50%)	46 (1.32%)	60 (1.72%)

(b) camera corpus

Table B.1.: The distribution of polar sentences in the review corpora. The numbers in brackets represent the proportion of sentences for a particular type and rating (e.g., "positive" in 5-star reviews).

recall	hotel	camera
100%	100.00%	100.00%
90%	57.76%	59.75%
80%	24.29%	24.74%
70%	12.04%	10.06%
60%	6.12%	5.03%
50%	2.86%	2.31%
40%	1.43%	1.05%
30%	0.41%	0.21%

Table B.2.: Recall levels and corresponding proportions of nominal mentions.

	combined	hotel	camera
discourse function	**frequency (share)**	**frequency (share)**	**frequency (share)**
sentiment	2880 (41.33%)	1554 (44.71%)	1326 (37.96%)
fact	1645 (23.60%)	885 (25.46%)	760 (21.76%)
conclusion	431 (6.18%)	231 (6.65%)	200 (5.73%)
personal context	383 (5.50%)	179 (5.15%)	204 (5.84%)
advice	304 (4.36%)	138 (3.97%)	166 (4.75%)
OTHER	213 (3.06%)	87 (2.50%)	126 (3.61%)
problem	199 (2.86%)	90 (2.59%)	109 (3.12%)
usage	162 (2.32%)	38 (1.09%)	124 (3.55%)
comparison	132 (1.89%)	38 (1.09%)	94 (2.69%)
general remark	97 (1.39%)	34 (0.98%)	63 (1.80%)
requirement	92 (1.32%)	20 (0.58%)	72 (2.06%)
lack	90 (1.29%)	37 (1.06%)	53 (1.52%)
heading	86 (1.23%)	29 (0.83%)	57 (1.63%)
summary	84 (1.21%)	41 (1.18%)	43 (1.23%)
review	74 (1.06%)	31 (0.89%)	43 (1.23%)
purchase	62 (0.89%)	20 (0.58%)	42 (1.20%)
expectation	35 (0.50%)	24 (0.69%)	11 (0.31%)

Table B.3.: The distribution of discourse functions in the review corpora.

discourse function DF	**Pr** (**polar** | **DF**)
summary	1.00
lack	1.00
problem	1.00
sentiment	1.00
conclusion	0.99
expectation	0.83
comparison	0.81
review	0.68
fact	0.33
advice	0.33
purchase	0.32
general remark	0.31
personal context	0.17
requirement	0.15
usage	0.14
OTHER	0.03

(a) Correlation with polarity dimension.

discourse function DF	**Pr** (**on-topic** | **DF**)
summary	1.00
conclusion	0.97
lack	0.94
problem	0.93
comparison	0.93
sentiment	0.93
expectation	0.80
fact	0.69
advice	0.68
requirement	0.66
review	0.54
purchase	0.47
usage	0.35
general remark	0.34
personal context	0.20
OTHER	0.11

(b) Correlation with topic dimension.

Table B.4.: The correlation of the discourse function dimension with the polarity and topic dimensions.

C. Automatic Construction of Product Aspect Lexicons

C.1. Estimates for the Likelihood Ratio Test

C.1.1. Candidate Ranking

Using maximum likelihood estimates, we derive p, p_1, and p_2 as follows:

$$p = \frac{D_{11} + D_{21}}{D_{11} + D_{12} + D_{21} + D_{22}} \ , \ p_1 = \frac{D_{11}}{D_{11} + D_{12}} \ , \ p_2 = \frac{D_{21}}{D_{21} + D_{22}}$$

Presuming that the counts are binomially distributed with $b(k; n, x) = \binom{n}{k} x^k (1-x)^{n-k}$, the likelihoods for obtaining the observed counts under H_0 and H_1 are calculated as

$$\begin{aligned} L(H_0) &= b(D_{11}; D_{11} + D_{12}, p) * b(D_{21}; D_{21} + D_{22}, p) \\ L(H_1) &= b(D_{11}; D_{11} + D_{12}, p_1) * b(D_{21}; D_{21} + D_{22}, p_2) \end{aligned}$$

and we get the log-likelihood ratio $log\lambda$ as

$$\begin{aligned} log\lambda = \log \frac{L(H_0)}{L(H_1)} &= \log \frac{\binom{D_{11}+D_{12}}{D_{11}} p^{D_{11}} (1-p)^{D_{12}} * \binom{D_{21}+D_{22}}{D_{21}} p^{D_{21}} (1-p)^{D_{22}}}{\binom{D_{11}+D_{12}}{D_{11}} p_1^{D_{11}} (1-p_1)^{D_{12}} * \binom{D_{21}+D_{22}}{D_{21}} p_2^{D_{21}} (1-p_2)^{D_{22}}} \\ &= \log p * (D_{11} + D_{21}) + \log(1-p) * (D_{12} + D_{22}) \\ &- D_{11} \log p_1 - D_{12} \log(1-p_1) - D_{21} \log p_2 - D_{22} \log(1-p_2). \end{aligned} \tag{C.1}$$

C.1.2. Pros/Cons Pre-Modifier Filter

Let $D_{Pros}(ws)$ be the document frequency of a word sequence ws in a corpus of pros lists, where each list is considered an individual document and let $D_{Cons}(ws)$ be defined analogously. Then we estimate p, p_1, and p_2 as

$$p = \frac{D_{Pros}(pm, head) + D_{Cons}(pm, head)}{D_{Pros}(head) + D_{Cons}(head)}, \quad p_1 = \frac{D_{Pros}(pm, head)}{D_{Pros}(head)}, \quad p_2 = \frac{D_{Cons}(pm, head)}{D_{Cons}(head)}$$

where $D_x(s_1, s_2)$ refers to the document frequency of the co-occurrence of strings s_1 and s_2 in corpus x. Analogously to Eq. (C.1), we calculate a log-likelihood ratio as

$$\begin{aligned} \log \lambda &= \log p * (D_{Pros}(pm, head) + D_{Cons}(pm, head)) \\ &+ \log(1-p) * (D_{Pros}(head) + D_{Cons}(head) - D_{Pros}(pm, head) - D_{Cons}(pm, head)) \\ &- \log p_1 * D_{Pros}(pm, head) - \log(1-p_1) * (D_{Pros}(head) - D_{Pros}(pm, head)) \\ &- \log p_2 * D_{Cons}(pm, head) - \log(1-p_2) * (D_{Cons}(head) - D_{Cons}(pm, head)) \end{aligned} \tag{C.2}$$

C.2. Aspect Detection Algorithms

To conduct the extrinsic evaluation of the automatically constructed aspect lexicons (refer to Section 7.7), we need to define how to apply them for detecting product aspects in natural language text. This section describes and evaluates two approaches, which differ mostly in the way linguistic information is incorporated.

The most straight forward approach is described by Algorithm C.1. The algorithm gets as input a stream of tokens (optionally lemmas) and finds all matches of lexicon entries. These resulting matches are restricted to a set of of non-overlapping matches: From multiple overlapping matches, only the left-most, longest-matching, and highest-scoring lexicon entry is selected. We apply the Aho-Corasick algorithm [5] as implemented in the LingPipe NLP framework[1]. If the *BNP1* pattern was used to build the lexicon (see Section 7.4.2), it only contains singular form terms. By lemmatizing the string, we simulate a simple inflectional variant aggregation.

Algorithm C.1 Product Aspect Detection (1)

apply linguistic preprocessing to document ▷ tokenization, lemmatization, POS tagging
$\mathfrak{L} \leftarrow$ the lexicon
$\mathfrak{p} \leftarrow$ part-of-speech pattern used to generate lexicon
if $\mathfrak{p} =$ *BNP1* **then**
  $C \leftarrow$ set of non-overlapping lexicon matches in lemmatized and lower-cased input
else
  $C \leftarrow$ set of non-overlapping lexicon matches in tokenized and lower-cased input
end if
for all $c \in C$ **do**
  mark c as detected product aspect and attach score from $\mathfrak{L}$
end for

Algorithm C.2 first extracts candidate strings by applying the part-of-speech tag pattern *BNP2* and then tries to match the candidates with lexicon entries. The algorithm applies two linguistically informed heuristics to match substrings of candidates. The first step is to conduct linguistic pre-processing to the document. Next, a set of candidate extractions C is generated by applying the part-of-speech tag pattern *BNP2*. Extracting candidates beforehand ensures that potential matches exhibit valid part-of-speech tags. This allows for shallow word sense disambiguation based on part-of-speech tags. Then, for each candidate $c \in C$, we try to match an entry of the lexicon. Depending on the part-of-speech pattern used to construct the lexicon, we either try to match the lower-cased lemma or the lower-cased token string s.

[1] refer to http://alias-i.com/lingpipe

Algorithm C.2 Product Aspect Detection (2)

```
apply linguistic preprocessing to document                ▷ tokenization, lemmatization, POS tagging
𝔏 ← the lexicon
𝔭 ← part-of-speech pattern used to generate lexicon
C ← set of candidate extractions generated by application of part-of-speech pattern BNP2
for all c ∈ C do
    if 𝔭 = BNP1 then
        s ← lower-cased concatenation of lemmas covered by c
    else
        s ← lower-cased concatenation of tokens covered by c
    end if
    if s ∈ 𝔏 then
        mark c as detected product aspect and attach score from 𝔏          ▷ exact match!
        continue looping over C
    end if
    while first token in c is adjective do                ▷ remove preceding adjectives and match
        c ← c with first token removed
        if c ∈ 𝔏 then
            mark c as detected product aspect and attach score from 𝔏
            continue looping over C
        end if
    end while
    𝔖 ← set of consecutive substrings of c that exist in 𝔏          ▷ try to match noun substrings
    if 𝔖 ≠ ∅ then
        c* ← the longest substring s ∈ 𝔖, in case of draw with highest score
        mark c* as detected product aspect and attach score from 𝔏
    end if
end for
```

If the candidate string s is contained in the lexicon, we are done, mark the candidate as detected product aspect and associate it with the score provided by the lexicon. If no exact match is found and s is a compound, the algorithm tries to match *reasonably* constructed substrings of s. First, the algorithm truncates any preceding adjectival modifiers one by one and each time checks for a lexicon match. The intuition is that in English language, typically the final nouns in a compound refer to the general class of an entity and any adjectival pre-modifiers derive a more specific type of that class (e.g., "digital camera" $\rightarrow$ "camera"). If this heuristic does not produce a match, we try to match any *token n-gram* of s. Take note, that at this point s may only consist of nouns. From all token n-gram matches, we choose the one which is longest (in terms of tokens). If two or more longest matches have the same length, we choose the match with the highest score.

C.3. Evaluation of the Baseline Approach

In this section, we present a detailed analysis of the results we obtained with the baseline method for product aspect lexicon construction (refer to Section 7.6).

C.3.1. Intrinsic Evaluation

The lexicons generated by the baseline method contain 1182 (hotel) and 953 (digital camera) entries. In our setting, the dictionaries' sizes are around two orders of magnitude larger than reported by Yi et al. [452] or Ferreira et al. [128]. Their lexicons consist only of around 40 entries. This difference is mainly due to the different sizes of foreground and background corpora (we examine this more

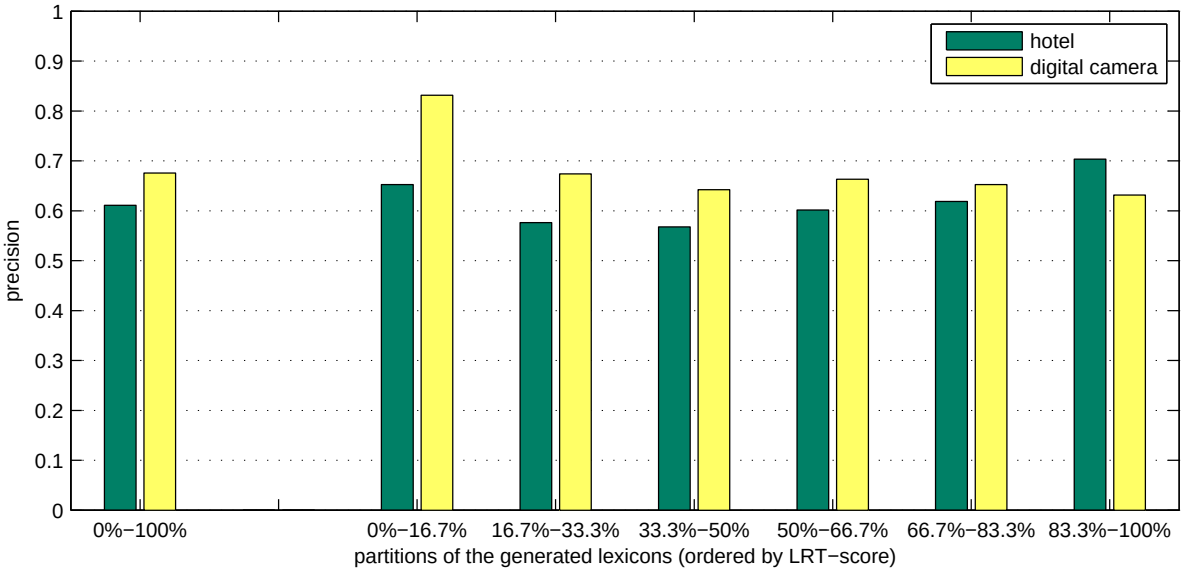


Figure C.1.: Intrinsic evaluation of the baseline approach. The bar chart shows precision values for different partitions of the ranked list of extracted terms.

closely in Appendix C.5 and Sections 7.7.5 and 7.7.6). While we apply the algorithm on corpora of several ten thousand of documents, they examine corpora comprising only a few hundred documents. Yi et al. [452] perform only an intrinsic evaluation of the algorithm and find *precision@n* values of nearly 1.0 on two different corpora ($n = 38$ and $n = 31$). We cannot confirm these high values with a precision@40 of 0.625 (hotel) and 0.800 (digital camera). Regarding the complete lexicons, we achieve precision values of 0.612 (hotel) and 0.676, respectively (digital camera). Figure C.1 depicts the precision of different portions of the extracted lexicons. The two leftmost bars (0%-100%) represent the precision when evaluated over the entire lexicon, whereas the other bars each represent one sixth of the lexicons — for example, the first portion represents the $16^2/_3$% of the lexicon with highest scores. Unexpectedly, the precision of the different portions of the lexicons does not decrease with lower ranks. Indeed, for the hotel domain, the last sixth ($83^1/_3$%-100%) with lowest ranks exhibits the highest precision value. Although, for the digital camera dataset, we find that the first sixth has a significantly higher precision, no constant decrease of precision is observed for portions with lower ranks. That is, even if we would choose higher threshold values for the the candidate selection, we cannot increase the precision of the extracted lexicons, but would lower the recall in the aspect detection task. On the other hand, this raises the question whether it is feasible to lower the threshold (or apply a different selection method) so that a higher recall for the detection task can be achieved at a constant level of precision (see Section 7.7.6).

Mistake analysis reveals that mainly four phenomenons are the cause of incorrect lexicon entries (i.e., false positives). One cause is that the lexicon generation process erroneously identifies **product related terms** as aspects, although they do not represent valid product aspects according to our criteria. For example, terms such as "subway stop", "financial district", or "shopping center", which are clearly related to the domain of hotel reviews, but not valid product aspects, are wrongly extracted. In the digital camera domain we find terms such as "computer", "hard disk", or "JPEG". As these types of terms naturally occur with high frequency in the corpora, many false matches are produced when detecting product aspects.

A second major cause is **falsely identified pre-modifiers**. As mentioned in Section 7.4.3, these are either sentiment bearing modifiers (e.g., "*well-appointed* room", "*lovely* room", "*fantastic* zoom") or uni-

		nominal mentions			all mentions		
dataset	**scenario**	precision	recall	f-measure	precision	recall	f-measure
Hotel	A	0.485 (0.582)	0.734 (0.881)	0.584 (0.701)	–	–	–
Hotel	B1	0.397 (0.489)	0.715 (0.880)	0.511 (0.629)	0.386 (0.499)	0.614 (0.794)	0.474 (0.613)
Hotel	B2	0.636 (0.786)	0.625 (0.773)	0.630 (0.779)	0.615 (0.792)	0.537 (0.692)	0.573 (0.739)
Hotel	B3	0.793 (0.976)	0.715 (0.880)	0.752 (0.926)	0.744 (0.962)	0.614 (0.794)	0.673 (0.870)
Camera	A	0.592 (0.737)	0.725 (0.902)	0.652 (0.811)	–	–	–
Camera	B1	0.408 (0.525)	0.697 (0.896)	0.515 (0.662)	0.386 (0.521)	0.563 (0.760)	0.458 (0.618)
Camera	B2	0.609 (0.772)	0.600 (0.760)	0.604 (0.766)	0.587 (0.777)	0.487 (0.645)	0.532 (0.705)
Camera	B3	0.736 (0.946)	0.697 (0.896)	0.716 (0.921)	0.687 (0.927)	0.563 (0.760)	0.619 (0.835)

Table C.1.: Results for product aspect and sentiment target detection with the baseline method. Results that are based on the lenient evaluation measure are shown in parentheses.

versal modifiers (e.g., "*particular* room", "*other* room", "*new* zoom", "*large* display screen"). Besides not generating appropriate product aspects, wrongly determined modifiers lead to lower term/document frequency counts of the actual product aspect — for instance, occurrences of "particular room" or "other room" should be counted as occurrence of the actual aspect "room". With regard to the LRT-score, this may lower the significance of a term candidate, which is then not added to the lexicon. Both types of false pre-modifiers have a negative influence on the performance of sentiment target or aspect detection: First, when evaluating with the strict measure, each match of such a lexicon entry produces both, a false positive and a false negative. Second, with regard to detecting sentiment targets, a match containing a sentiment bearing pre-modifier may hinder the identification of sentiment words associated with the match. Thus, the recall in detecting sentiment targets in scenario *B2* is potentially lowered.

A third cause for false positives in the lexicon is the extraction of terms which are significant for the domain of customer reviews in general, but not for the actual product type (e.g., "review", "problem", or "highlight"). Although very common, these **review-related terms** occur with significantly higher frequency in customer reviews than in general language (e.g., the "ukWaC Web Corpus") and are thus extracted by the contrastive LRT-ranking. We discussed these types of terms in Section 7.4.3.

A fourth cause are terms that are too specific to be considered as valid product aspects (e.g., "19th floor", "12x optical zoom", "4gb memory card"). Admittedly, the validity of such terms as aspects is a question of definition. We decided not to regard these types of terms as valid, which is also reflected by our annotation guidelines (see Appendix A.3). Most commonly these **overly specific terms** contain numerical modifiers and often mention a measurement unit (e.g., "4gb card", "28mm wide angle lens", or "30sqm room").

C.3.2. Extrinsic Evaluation — Scenario A (Aspect Mentions)

Results for the extrinsic evaluation of the baseline method are presented in Table C.1. We apply the baseline approach on the two test corpora and show evaluation results distinguishing between *nominal only* versus *all* mention types, *strict* versus *lenient* evaluation measure, and between the four evaluation scenarios *A* and *B1-B3*. For scenario A, we only present results for nominal mention types as our annotation scheme does not cover reference or implicit mention types that are not targeted by a sentiment expression.

In scenario A, the baseline method achieves moderate values for the f-measure on both corpora. Strict evaluation results in an f-measure of 0.584 for the hotel review corpus and 0.652 for the camera dataset, respectively. Considering only nominal mentions, for both datasets we find that the recall values are comparably high with 0.734 and 0.725. Regarding the camera dataset, we exhibit a higher

precision of 0.592 compared to 0.485. We attribute this to the generally higher precision of the extracted camera aspect lexicon and in particular to the high precision of the first portion of this lexicon (which contains many of the frequently mentioned aspects). We make similar observations when considering the lenient evaluation results. Here, we measure f-scores of 0.701 and 0.811.

Mistake Analysis Examining the false negatives produced by the lexicon-based aspect detection method in scenario A, we identify several different causes for missing extractions. A frequent cause is that the method **fails to recognize common terms** that also represent product aspects. For instance, in the hotel domain the lexicon does not include aspects such as "breakfast", "management", "Internet", or "shower". In the camera corpus, valid aspects such as "options", "flexibility", "handling", or "value" are missing. Although these terms occur with high frequency in the foreground corpus, they are so common in general language that their LRT-score is lower than the defined threshold of 3.84. A further factor for reduced recall is due to the approximate Zipfian distribution of aspect mentions. The foreground corpora can never be large enough — there will always exist mentions that occur only once or with very low frequency. For instance, in the digital camera corpus, false negatives such as "AF assist lamp", "charging cradle", or "fluorescent lighting setting", appear only once or not at all in the foreground corpus. Thus, the LRT-approach or any other frequency based ranking method has **difficulties in extracting low frequency terms**. The low-recall/high-precision heuristic, employed to initially discover candidate terms, even amplifies this effect. Jakob [182][chap. 3] studies the influence of low frequency terms more closely and reports similar findings[2]. As previously indicated, another frequent cause of false negatives is **wrongly identified pre-modifiers**. For instance, lexicon entries such as "great camera", "8x zoom", or "comfortable bed" produce false positives and false negatives in the strict evaluation scenario. Further reasons are **misspellings** (e.g., "continetal breakfast" or "pictues"), **unrecognized part-of-speech patterns** (e.g., "shot to shot speed") and **unidentified compositional variants** (e.g., "size of the room" or "quality of the images").

Examining false positives in more detail, we identify mainly five different types of errors. For each of the two test corpora, we (randomly) sample subsets of 200 false positives produced by the baseline method and analyze these more closely. With regard to the hotel dataset, we find that 65% of false positives are caused by **(1) incorrect lexicon entries** (e.g., "car", "wife", "annoyance", "city"). Another 16% of false positives can be ascribed to **(2) partial matches** (e.g., "air con" instead of "air con unit" or "parking lot" instead of "indoor parking lot"). Take note that these at the same time count as false negatives and that a single false negative may produce multiple false positives: For instance, the lexicon contains the product aspects "bed" and "linens", but not "bed linens". Then in a sentence "The bed linens had stains on them.", the strict evaluation produces a false negative for missing the aspect "bed linens" and two false positives for the matchings "bed" and "linens". Partial matches may also be caused by **(3) false pre-modifiers** (e.g., "entire room", "great location"). We count this type of error separately and find that these account for a share of 7%. A fourth type of error is rooted in the **(4) missing context awareness** of a lexicon-based approach. Depending on the context, the occurrence of a valid product aspect term may actually refer to an entity that is not associated with the reviewed product. For instance in the sentence "The *bed* has convinced me that my next *bed* at home will be a Tempurpedic!", the first occurrence of the term "bed" refers to the hotel bed (which is thus annotated as product aspect mention), whereas the second mention is not related to the reviewed product (and is thus not annotated). Such errors also typically occur when reviewers provide general remarks (e.g., "As with many *hotels* today, ..."). On the hotel test corpus this error type is responsible for 8% of all false positives. Furthermore, we find that 4% or errors are caused by **(5) lexical ambiguity** of lexicon entries.

With respect to the digital camera test corpus, error types (3) and (5) are similarly distributed. In

[2] Although the author examines a different setting, the basic results are transferable. In contrast to us, Jakob [182] uses the test corpora as foreground corpora for contrastive ranking. No separate, larger foreground corpora are used. The analysis of low frequency terms is based on sentiment target detection scenarios.

the corresponding sample, we find that 8% of false positives can be attributed to false pre-modifiers and 4% relate to lexical ambiguity. We observe different analysis results for type (1), (2), and (4). In the digital camera corpus, partial matches as well as false lexicon entries make up a share of 35% each (i.e., together account for 70%). We attribute the lower share of type (1) errors to the higher precision of the generated lexicon (0.676 to 0.612). The relatively higher amount of partial match errors becomes apparent in the larger gap between strict and lenient results for the precision (0.141 for the camera dataset compared to 0.095 for the hotel corpus).

Extrinsic Evaluation — Scenarios B1-B3 (Sentiment Target Detection)

Analyzing the synthetic sentiment target scenarios B1-B3, a first observation is that even in the best case setting (B3, nominal mentions only), the baseline method does not achieve f-measures higher than 80 percent. In fact, for the hotel dataset we measure an f-score of 0.752 and for the camera corpus a score of 0.716. Reconsider that in scenario B3 the algorithm has perfect knowledge of sentiment targets. The algorithm extracts only candidates that overlap with a sentiment target. In consequence, one source of false positives is partial matches of valid nominal targets. To measure the amount of the different partial match failure types, we examined the whole set of false positives produced on both corpora. We distinguish between matches that preserve the correct sense of the associated aspect (e.g., "wireless internet" instead of "wireless internet access") and matches that do not (e.g., "auto" instead of "auto zoom feature"). With respect to the hotel dataset we find that 77% of false positives actually preserve the sense and 23% do not. We further subdivide the 77% share of sense preserving failures into errors caused by partial matches due to false pre-modifiers (44%) and other partial matches (33%). Regarding digital camera corpus, the basic ratio is similar with 73% to 27%. Here, false pre-modifier partial matches account for 20%, whereas other partial matches sum up to 53%. Considering sense-preserving partial matches as correct and reevaluating our results with this knowledge leads to precision values of 0.953 for the hotel test corpus and 0.928 for the digital camera test corpus in scenario B3. The lenient evaluation measure thus slightly overestimates this reevaluated precision.

Discussion of Precision Results As expected, precision increases the more accurate and complete information on sentiment expressions is available to the extraction algorithm. For the hotel corpus, the baseline approach achieves precision values of 0.397, 0.636, and 0.793 for scenarios B1, B2, and B3. For the digital camera dataset these numbers are slightly less (except for B1) with 0.408, 0.609, and 0.736. In scenario B1 only sentence level information on sentiment is given. Here, we observe that a great share of false positives is caused by a small share of high frequency terms, such as "camera", "picture", and "shot" for the digital camera corpus or "hotel" and "room" in the hotel review dataset. These terms, which either refer to the product type itself ("camera", "hotel") or to a very important concept regarding the product ("picture", "room"), often occur in constructs that actually refer to other more specific product aspects. For instance, in the sentences "The beds were very comfortable in our room.", or "My dad bought this camera before me, and I was impressed with the picture quality.", the occurrences of "hotel" and "camera" are not direct target of the sentiment expression (the actual sentiment targets are "bed" and "picture quality"). References to the product type or important concepts also occur quite often in compositional constructs such as in "The location of the hotel is extremely convenient.", which leads to a false extraction of the match "hotel".

Discussion of Recall Results The recall values that we achieve in the best case scenario B3 under lenient evaluation, in effect, correspond to the coverage of the extracted lexicons. For the baseline method, we report recall values (nominal-only, lenient) of 0.880 for the hotel corpus and 0.896 for the digital camera dataset. For both datasets, we perform mistake analysis and examine the whole set

of false negatives more closely. We find that the major share (hotel: 77.2%, camera: 82.5%) of missed sentiment targets can be ascribed to non sufficient coverage of the extracted lexicons, either because aspects correspond to common words in general language (e.g., "value") or because targets exhibit too low frequency (e.g., "spirituality menu"). Furthermore, misspellings are responsible for 6.6% (hotel) and 6.2% (camera) of false negatives. Another 8.8% (hotel) and 8.2% (camera) can be attributed to non-covered part-of-speech tag patterns[3](e.g., the tagger generates "checking/VBG in/IN" or "ease/NN of/IN use/NN"). The rest (hotel: 7.4%, camera: 3.1%) of false negatives is due to multiple partial matches of a single sentiment target (e.g., we match "room service" and "staff" in "room service staff", and count the second match as false negative even in lenient evaluation).

Due to the specific setting, recall values for scenarios B1 and B3 are equal: Extracting each match in a polar sentence (B1) or extracting each match overlapping a sentiment target (B3) result in the same set of covered sentiment targets. Recall values for scenario B2 are lower (hotel: 0.625 compared to 0.715, camera: 0.600 compared to 0.697), as the "closest match" heuristic only extracts a single match per sentiment expression. That is, in a situation when a single sentiment expression targets n product aspects, $(n-1)$ aspects are not recognized (unless targeted by another sentiment expression). Lenient recall values for scenarios A and B3 (both representing lexicon coverage) are very similar (hotel: 0.881 to 0.880, camera: 0.902 to 0.896), reflecting the observation that terms referring to general aspect mentions (A) are similarly distributed in the corpora to terms related to sentiment targets (B3). Unsurprisingly, the recall values, when evaluated on all mention types, are lower compared to evaluation of nominal mentions only. The lexicon-based approach is simply unable to resolve pronominal mentions and also misses implicit mentions. Since we apply product name filtering, the baseline method also does not extract named mentions. This results in around 10 percentage points lower recall for all scenarios.

C.4. Influence of the Aspect Detection Algorithm

In this evaluation scenario, we examine the influence of the detection algorithm. Whereas the extrinsic evaluation of the baseline approach is based on the application of the simple string matching Algorithm C.1, we now consider the linguistically more informed Algorithm C.2. Table C.2 reports the results for the baseline method when using Algorithm C.2 (i.e., the reported differences refer to Table C.1). We only consider nominal aspect mentions and the strict evaluation metric.

Comparing the two approaches lets not expect that recall values differ significantly as the main difference is related to the way overlapping matches are treated. This expectation is confirmed by the results. Although we can measure a minimally improved recall in most settings, the increase is statistically not significant at the chosen confidence level of 99%. Considering the precision achieved by the alternative detection algorithm, we can observe a statistically significant increase in all settings. For the hotel corpus we attain a maximum increase of 6.6 percentage points and regarding the digital camera corpus, the maximum increase is 5.8 percentage points (in scenario A). The general increase of precision can be mainly attributed to two reasons: First, restricting matches to adhere a given part-of-speech pattern reduces errors caused by the lexical ambiguity of lexicon entries. A second (minor) effect of applying the alternative algorithm is that the number of false positives produced by partial matches is lowered. In contrast to Algorithm C.1, it has access to candidate phrases (matching a part-of-speech pattern) and allows only one match per such candidate. For instance, assume that the generated lexicon contains the terms "battery" and "compartment door", but not "battery compartment door". In a sentence such as "The battery compartment door is very flimsy and feels like it's going to fly off.", the first algorithm would match the two non-overlapping occurrences of "bat-

[3] Take note that these numbers reflect the results of our corpus analysis with regard to the distribution of part-of-speech patterns (see Table 6.8). Corpus analysis showed that nearly 93% of nominal aspect mentions can be ascribed to simple noun phrases which are covered by our patterns.

		Nominal Mentions		
Dataset	**Scenario**	Precision	Recall	F-measure
Hotel	A	0.551 (0.066**)	0.730 (-0.004)	0.628 (0.044**)
Hotel	B1	0.433 (0.036**)	0.712 (-0.003)	0.539 (0.028**)
Hotel	B2	0.658 (0.022**)	0.630 (0.005)	0.644 (0.013)
Hotel	B3	0.813 (0.020**)	0.712 (-0.003)	0.759 (0.007**)
Camera	A	0.650 (0.058**)	0.725 (0.000)	0.686 (0.033**)
Camera	B1	0.443 (0.035**)	0.699 (0.002)	0.542 (0.027**)
Camera	B2	0.638 (0.029**)	0.617 (0.017)	0.627 (0.023)
Camera	B3	0.768 (0.032**)	0.699 (0.002)	0.732 (0.016**)

Table C.2.: Results for product aspect and sentiment target extraction with the baseline method when using the the alternative detection algorithm. Reported results are based on the strict evaluation measure.

tery" and "compartment door" as product aspects, thus generating two false positives (according to the strict measure). The second algorithm identifies "battery compartment door" as candidate and since the complete candidate is not part of the lexicon, matches the longest known substring which is "compartment door" in this case.

C.5. Comparability to Related Work

In our experiments with terminology extraction, we define the relevance of identified product aspects with regard to a whole class of products (e.g., hotels or digital cameras), instead of a single specific product or model. This becomes manifest in our use of a separate, very large foreground corpus, as well as in the composition of our evaluation corpora. In contrast, the major share of related work, such as Ferreira et al. [128] or Hu and Liu [177], define relevance towards an individual product. As a consequence, foreground and evaluation corpora contain reviews commenting on a single, specific product only. Furthermore, which is not a direct corollary, in their studies, foreground and test corpora are in fact identical — no separate dataset is used to acquire the lexicon. Independent of the definition of relevance, we believe that especially this experimental setup is contra-indicated when evaluating an unsupervised approach. Unsupervised approaches have the great advantage of gathering knowledge from unlabeled data. Such data is typically easily available (e.g., by a web crawl) and should be exploited.

In this section we are interested in answering the following two questions: How well does our (more realistic) approach of using a separate, large foreground corpus perform in comparison to an approach that does not? And second, how well does a "product-class-centric" lexicon perform on a dataset that only contains reviews of a specific product (belonging to that class). We answer both questions in combination and propose a single experimental setup:

As test corpora, we use the "Hu/Liu dataset" [177] and its extension by Ferreira et al. [128] (see also Section 5.4.3). In particular, we run our evaluations on the two sections of the corpora that cover digital camera reviews. The first section contains 45 customer reviews for the Canon G3 camera and the second section consists of 34 reviews of the Nikon Coolpix 4300 camera model. Recall that the original corpus contains annotations only for sentiment targets, whereas as Ferreira's extension covers each mention of a product aspect. To answer our questions, we are only interested in the performance with regard to recall. The precision value can be assumed to be similar to the findings with our own corpora. We calculate the recall value by extracting all sentiment targets/aspect mentions from each test corpus and looking up whether it is contained in our extracted lexicon. We compare the

values with the results reported by Ferreira et al. [128] who examined the same approach, but without exploiting a separate foreground corpus.

Ferreira's extension contains 594 aspect mentions (161 distinct aspects) for the first camera corpus and 340 aspect mentions (120 distinct aspects) for the second dataset. In these corpora each mention annotation refers to a nominal mention. Out of 594 mentions we are able to find 475 with our lexicon, resulting in a recall of 0.800. In the second corpus, we detect 271 out of the 340 mentions which corresponds to a recall of 0.797. Compared to the results reported by Ferreira et al. [128], we achieve recall values which are 54.8 percentage points (increase of 317%) and 64.1 percentage points (increase of 511%) higher. The absolute results achieved on these datasets are in compliance to findings on our own corpus — we can compare to the results of evaluation scenario *A* for the camera corpus which shows a recall of 0.725 (Table C.2). Assuming that the precision value of 0.592 (as measured on our corpus) is constant, we can *estimate* an f-measure of 0.680 for the Canon G3 dataset and a value of 0.679 for the Nikon Coolpix corpus. Compared to Ferreira's findings we observe improvements of 29.7 (+178%) and 41.9 (+261%) percentage points in f-measure.

We also apply our lexicons to the original Hu/Liu corpus. The Canon G3 section of the corpus contains 257 nominal mentions of sentiment targets (99 distinct aspects). Out of these we recognize 188, corresponding to a recall value of 0.732. For the Nikon Coolpix dataset, we achieve a recall 0.741 with 137 matches out of 185 targets (74 distinct aspects). Again these results are in compliance with the findings regarding our own corpus. The setting is comparable to evaluation scenario *B3*, which shows a recall of 0.699 for the camera dataset (Table C.2). Ferreira et al. [128] do not report results for the original corpus, so that we cannot calculate any potential improvements.

In summary, the results show that utilizing a separate, large foreground corpus for lexicon acquisition substantially improves the recall. As hypothesized, the examined unsupervised approach benefits significantly from utilizing a large collection of unlabeled data. Also the second question can be answered positively. In the aspect detection (*A*) as well as in the sentiment target extraction scenario (*B3*), the "product-class-centric" lexicon achieves good recall values, that is, detects the major share of the product's aspects.

D. Acquiring Coarse-Grained Product Aspects with Probabilistic Topic Modeling

D.1. Overview and Related Work

In this section we consider the task of initially determining a set of coarse-grained aspects that adequately describe a given product type. Basically, two problems need to be solved. First, the actual level of granularity must be defined. For example, we need to decide whether aspects such as "room amenities", "bathroom", and "view" are considered separately, or whether they are all subsumed in an (even) more coarse-grained topic "room". Second, we need to identify the individual relevance of a specific topic. For example, we need to decide whether we want to include an aspect such as "housekeeping" or do not consider it at all. Obviously, these decisions are also application specific, that is depend on concrete requirements for the customer review mining system.

In analogy to the task of *ontology engineering*, we can most basically distinguish two main paradigms for determining a set of relevant topics (concepts):

- **Domain experts**: Relevant concepts are identified manually by a group of persons which have good knowledge of the product domain. Existing knowledge bases and the actual data (i.e., the customer reviews) may be consulted during the manual acquisition process.
- **Data-driven, semi-automatic**: Unsupervised, statistical methods are used to analyze the actual data. The most prevalent concepts are automatically identified and presented along with a relevance score to a person which manually refines them in a post-processing step.

As already pointed out in previous chapters, we pursue a data-driven approach. Obviously, this approach has the advantage of taking into account the real distribution of the different topics as found in the data. For example, a domain expert may find that "low-light performance" is not a relevant topic on its own and subsumes it as part of a more general topic "picture quality". However, the actual data may show that "low-light performance" is a concept which is very frequently discussed by reviewers and consequently should considered separately. The data-driven approach thus helps in guiding the decision process. Our data-driven approach is based on probabilistic topic modeling techniques:

In Section 4.4.3 we already introduced topic models (and adaptations) as an approach to statistically model the information contained in customer review documents. We briefly reconsider the concept and in particular point out its applicability for analyzing the topic dimension of reviews. Blei [42] describes *probabilistic topic models* as "algorithms for discovering the main themes that pervade a large and otherwise unstructured collection of documents". In contrast to traditional document clustering techniques[1], a topic model is capable of assigning multiple topics to a single document — that is, a document is considered as a mixture of latent themes. For example, a hotel review may contain a mixture of the themes "check-in process", "hotel location", and "room amenities". Topic models thus allow for decomposing a document's content with regard to its underlying thematic structure. Naturally, the main fields of application are the organization and summarization of large document collections, exploratory corpus analysis, and tasks concerned with prediction problems.

Currently, the most commonly used approach to probabilistic topic modeling is the *latent Dirichlet allocation* (LDA) model introduced by Blei et al. [43]: LDA assumes a generative, topic-driven proba-

[1] see for example Manning et al. [250, chapters 16,17]

bilistic model of text corpora[2]. Each document is ascribed to a finite mixture of latent topics (distribution over topics) and each word in a document is generated by one of the document's topics (i.e., each topic is associated with a distribution over words). Standard statistical inference methods are then applied to estimate the model parameters that best describe the observed data (i.e., the collection of documents). With such a probabilistic model it is then possible to derive an association between individual words and the latent topics prevalent in the corpus — for example, that the words *shower*, *sink*, *toilet*, or *tub* are very likely to be generated by a latent topic *bathroom* in a corpus of hotel reviews. Several researchers have studied the applicability of topic modeling approaches within the context of customer review mining:

For example, Titov and McDonald [382] cast the task of detecting coarse-grained, ratable aspects in customer reviews as a topic modeling problem. They propose to extend the LDA method as they found that such a standard topic modeling approach does not fit well in the task of detecting ratable aspects. They argue that, since the bag-of-words assumption of the standard model only allows to incorporate co-occurrence statistics at the document level, mainly "global" topics are inferred that distinguish the individual documents (e.g., reviews of New York hotels from London or Berlin hotels). Instead, they propose a **Multi-grain LDA** that distinguishes "global" (document level) and "local" topics.

Whereas global topics are fixed for each document, local topics are induced on the sentence level (depending on a local context which is defined as a sliding window over sentences). Each word is then either sampled from one of the local or global topics. The authors hypothesize that the words associated with the induced local topics correspond to the ratable aspects in a review corpus. A qualitative evaluation on different review corpora underpins their assertion: Local topics in the MG-LDA model pretty well correspond to ratable aspects. Compared to the MG-LDA approach, the LDA model's coverage of ratable aspects is lower and the topic coherence is worse. However, they also find that the quality of results is much dependent on the dataset. For example, both models fail to capture the majority of aspects when applied to a corpus of restaurant reviews[3].

It is not directly clear how to apply the inferred topic models in a prediction task (e.g., extracting product aspects mentioned in a customer review). In contrast to the MG-LDA model, the standard LDA does not provide estimates for the topic distribution of individual syntactical units such as sentences. Titov and McDonald basically propose to "set the probability of a topic for a sentence to be proportional to the number of words assigned to this topic". Due to the lack of a dataset that is annotated on the sentence level, evaluation of aspect detection is only performed indirectly as part of a sentiment analysis task. Results of this extrinsic evaluation show improvements with LDA and MG-LDA in the task of predicting numerical ratings of individual, coarse-grained product aspects.

Despite the conclusions of the previously cited work, Brody and Elhadad [55] apply the "standard" LDA topic modeling approach to find ratable aspects. To overcome the problem with the derivation of global topics, they propose to treat each single sentence as a separate document during the inference process. In other words, instead of altering the algorithm, they change the shape of the input to the algorithm. With regard to this specific **local LDA** version, the findings of Titov and McDonald [382] and Brody and Elhadad [55] are inconsistent. Also Titov and McDonald experiment with the local LDA configuration, but they report inferior results, in particular that many inferred topics do not correspond to ratable aspects and that valid topics partly exhibit a lack of coherency. Brody and Elhadad evaluate the local LDA approach on a corpus of manually labeled sentences stemming from the domain of restaurant reviews. Whereas the original corpus [138] contains sentences with multiple topic labels, they simplify the test corpus to a subset consisting of sentences with only a single label. Only four topics corresponding to ratable aspects are defined in the gold standard ("food&drink",

[2] LDA is however not restricted to textual data.

[3] In this case, Titov and McDonald [382] hypothesize that the large number of different cuisines and the related specific vocabulary (e.g., "pizza", "pasta" vs. "sushi", "noodles") hinders the unsupervised methods in detecting generic concepts, such as "meal dishes" or "fish dishes", and propose hierarchical topic models (e.g., [150]) to cope with this issue.

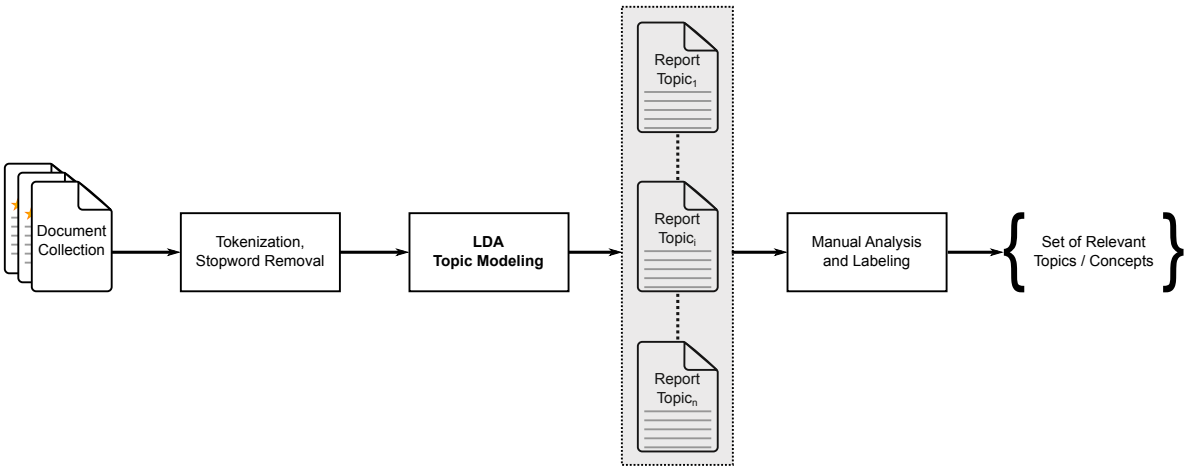


Figure D.1.: Probabilistic topic modeling for the exploration of text corpora.

"service", "price", and "atmosphere"). Two topics do not correspond to ratable aspects ("anecdotes" and "miscellaneous"). As the input to the local LDA are sentences, the inferred model can directly applied to estimate the topic distribution of individual sentences. A sentence is labeled with the most probable topic if the estimate is higher than a given threshold value. Otherwise no label is associated. Unfortunately, the authors do not provide very detailed results. Only precision-recall curves are reported and only for the three best performing topics. The results are not further discussed. Due to the restriction to a very small set of topics and the rather shallow evaluation of the topic assignment task, it thus remains unclear to which extent the local LDA approach is suited for detecting coarse-grained product aspects.

As discussed in Section 4.4.3, other researchers propose to consider the topic and sentiment dimensions jointly [194, 233, 259, 383, 465] and adapt standard topic models. We are not going into further details as these approaches are less relevant in the context of this chapter.

D.2. Implementation

We now discuss our concrete approach of utilizing topic modeling techniques. As we have sketched in the previously, the techniques are well suited for exploratory corpus analysis. Also, due to the unsupervised nature, very large document collections can be analyzed easily. Our basic approach is depicted in Fig. D.1. In the following we give a more detailed description:

For each application domain (hotels and digital cameras) we randomly sample a subset 50,000 review documents from the complete web crawls. The resulting datasets consist of 413,288 (hotel) and 523,003 (camera) sentences, spanning 6,898,712 and 8,196,561 tokens[4], respectively. We experiment with two representations of the document collections. We either represent the collection as the original set of 50,000 documents, or consider each single sentence as an individual document. The intuition behind the sentence-oriented representation is the same as pointed out by Brody and Elhadad [55]. By providing sentences instead of complete documents, the hope is to force the probabilistic topic model to find "local" instead of "global" topics. Each document (be it the original document or a sentence) is tokenized and case folded. We filter stop words with a predefined list[5] of words (524 entries).

[4]Tokenization is performed by a simple regular expression, considering whitespace boundaries and punctuation. Sentence splitting is based on the output of the Stanford CoreNLP tool.

[5]We use the list of stop words provided with the distribution of the MALLET software package.

As our task is mainly of exploratory nature ("find relevant topics"), we believe that a standard topic model based on LDA suffices to fulfill our needs. More sophisticated methods, such as the multi-grain topic model proposed by Titov and McDonald [382] or the approaches presented in [194, 233, 259], may be indicated for fully automatic settings where the models are directly used for prediction. We make use of the LDA implementation which is part of the open source *Machine Learning for Language Toolkit* (MALLET)[6] developed by McCallum [255]. The MALLET toolkit provides many configuration parameters with regard to the LDA inference process. Most importantly, the number of topics must be provided beforehand. It is unclear which amount of topics is reasonable for our task, but it is also not our goal to extensively experiment with this parameter. We only consider two different settings, 50 topics and 100 topics. Further relevant parameters are listed with a short description in Table D.1.

parameter	value	description
num-iterations	1000	the number of sampling iterations
optimize-interval	10	optimizes hyperparameters every x iterations to better fit the data
optimize-burn-in	20	starts hyperparameter optimization after x iterations

Table D.1.: Definition of parameter values for the LDA component of MALLET.

MALLET allows for several output formats, including a binary representation of the learned topic model. But since our goal is to explore the corpus, we are primarily interested in a human readable/interpretable form. As we have learned earlier, the LDA approach models each topic by a distribution over words. Thus, the words exhibiting the highest probability mass of such a distribution, typically characterize a topic/concept quite well. By looking at these top-n words, a human can relatively easy decide whether the extracted topic shows adequate coherency and if yes, can provide a meaningful label. For each topic, MALLET also reports the most common phrases (token sequences) that can be attributed to the topic. As an example, Table D.2 presents the top ten words and phrases for a topic that we easily identified as describing the concept "image stabilization".

rank	word	phrase
1	image (2282)	image stabilization (284)
2	stabilization (1572)	anti shake (58)
3	zoom (904)	image stabilizer (46)
4	camera (893)	hand held (27)
5	shake (765)	image stabilization works (21)
6	feature (571)	optical zoom (21)
7	tripod (559)	image stabilization feature (19)
8	hand (497)	anti blur (17)
9	anti (412)	image stablization (16)
10	steady (394)	steady hand (16)

Table D.2.: Top ten words and phrases for the topic "image stabilization". The number in brackets refers to the frequency a word or phrase is attributed to the topic.

In case the top keywords and phrases do not suffice to identify a coherent topic, we extract and consult exemplary sentences from the document collection that the topic model automatically associated with the concept under consideration[7]. If also this analysis does not help, we mark the concept as incoherent.

[6] http://mallet.cs.umass.edu/

[7] This additional step is only possible if we choose a sentence-oriented representation of the document collection.

indicator	original-100	sentence-100	sentence-50
relevant aspects	51/100 (56.2%)	57/100 (64.2%)	37/50 (72.1%)
incoherent topics	8/100 (8.6%)	5/100 (2.5%)	0/100 (0.0%)
global topics	14/100 (8.7%)	9/100 (7.0%)	2/100 (3.1%)

(a) hotel dataset

indicator	original-100	sentence-100	sentence-50
relevant aspects	48/100 (50.7%)	60/100 (67.7%)	30/50 (67.1%)
incoherent topics	12/100 (12.1%)	5/100 (2.5%)	1/100 (0.4%)
global topics	11/100 (4.9%)	4/100 (3.8%)	1/100 (0.5%)

(b) digital camera dataset

Table D.3.: The quality of the generated topic models for different inputs and a varying number of topics. The numbers in brackets indicate the share of tokens of the complete corpus that is attributed to the described topics.

We decide on the relevance of an extracted topic based on the relative importance of the concept and its "singularity". The relative importance of a topic is given by the share of the total corpus attributed to that topic (in comparison to other relevant topics). With "singularity" we refer to a concept's property of no or few overlap with other concepts. We are aware that both measures are rather fuzzy, but they reflect the indicators a human annotator can use during the decision process. With a model of 100 topics, we consider concepts exhibiting a support of 1% or more (i.e. that account for at least 1% of the complete corpus) as separate aspects. Concepts with a lower support are either not further considered or are subsumed within a more coarse-grained aspect. For instance, we find the concepts "color reproduction" and "image noise", both with a support significantly lower than 1%, that we consequently subsume within the aspect "picture quality". We also subsume concepts that we find have a too strong overlap with other corresponding concepts.

The two-level hierarchy as described in Section 5.2.2 is derived completely manual. We analyze the flat list of extracted topics and distill some main topics. Most of these main topics, such as "room", "location", or "service", are also extracted as concepts by the topic model, only a few, such as "sleep quality" (subsuming the aspects "bed" and "noise"), are rather "virtual", that is are not explicitly found by the LDA approach. Given a flat list of aspects, it is easy and effortless for a human (also with rather basic understanding of the specific domain) to build a simple two-level hierarchy.

D.3. Results

D.3.1. Documents vs. Sentences

We find that the sentence-oriented representation performs better than using the original documents. For this result, we mainly consider three indicators that characterize the quality of a generated topic model in our context. First, the number or share[8] of coherent topics which refer to a coarse-grained product aspect shows the ability of the model to focus on relevant concepts. Second, we characterize coherency by the amount and share of incoherent topics produced by the model. And third, the number and share of "global" topics (mostly cities in case of hotel reviews and brands in case of camera reviews) indicate the capability of finding the more relevant "local" topics.

Table D.3 compares the indicator values for the original and sentence-oriented input. With regard to this comparison, we only consider the setting with 100 topics. For the hotel review dataset and with

[8] We compute the share or relative frequency of a topic, by dividing the amount of tokens associated with the topics by the total number of tokens found in the corpus (stopwords not counted).

rank	word	phrase
1	water (2100)	hot water (81)
2	shower (1827)	water pressure (70)
3	hot (1082)	shower head (25)
4	pressure (609)	water pressure shower (17)
5	bathroom (549)	hot tub (15)

(a) bathroom - water temperature/pressure

rank	word	phrase
1	bathroom (1658)	bathroom amenities (11)
2	nice (728)	bath products (10)
3	towels (467)	bath shower (9)
4	toiletries (421)	soap shampoo (6)
5	products (387)	bath robes (6)

(b) bathroom - toiletries

Table D.4.: Top five words and phrases for two distinct topics that both refer to the aspect "bathroom". The number in brackets refers to the frequency a word or phrase is attributed to the topic.

complete reviews as input, we find 51 out of 100 topics that represent coherent coarse-grained product aspects. In total, these topics account for 56.2% of the complete corpus. With sentence-oriented input, the LDA model generates 56 out of 100 relevant topics which together cover a share of 64.2% of the corpus. Concerning the digital camera dataset, these numbers are even more pronounced. Here, with the original input, we find 48 out of 100 (corresponding to a share of 50.7%) topics to be relevant, whereas 60 out of 100 (share of 67.7%) are found with sentence-oriented input. Thus, we consistently observe a significantly higher ratio of relevant topics for the sentence-oriented input. We also find a lower number of incoherent topics (i.e., topics for which a human annotator could not find an adequate label) for this type of input. With the original input 8 (hotel) and 12 (camera) out of 100 topics are marked incoherent, covering a share of 8.6% and 12.1%, respectively. Using sentences as input, the number of incoherent concepts is less. In both datasets we observe 5 incoherent topics which span 2.5% of all tokens in the corpus. Furthermore, we find that the absolute number of global topics that refer to a city (hotel) or brand (camera) is reduced. Whereas for the original input, 14 (hotel) and 11 (camera) concepts refer to global topics, with sentence-oriented input we only observe 9 and 4 global topics. However, the share of the complete corpus attributed to global topics is not significantly reduced. For the hotel dataset the reduction is from 8.7% to 7.0% and for the camera corpus it is 4.9% to 3.8%. In summary, all three indicators show better results for the sentence-oriented representation. We conclude that this input format is better suited to guide the decision process of a human annotator during the initial acquisition of coarse-grained product aspects.

D.3.2. Generated Topic Models and Number of Topics

We discuss the learned topic models and compare the models generated with 100 and 50 topics. For the comparison, we consider the indicators introduced in the previous section, but also compare the coverage of the different models. Setting the number of topics to 100, the LDA process generates 56 different concepts that refer to relevant aspects for the hotel dataset. Within the 56 extracted topics, we initially identify 22 unique aspects, that is some aspects are represented by multiple concepts. Table D.4 provides an example for the aspect "bathroom". The keywords and phrases for the first topic show that this concept primarily addresses the water temperature and water pressure of the shower or tub. The second topic describes a concept that we labeled as "toiletries". As both topics account for less than 1% of the complete corpus, we subsume them with the more general aspect "bathroom".

Out of the 22 unique topics, we reject only one topic ("hotel website") as it does not fulfill the minimum support and cannot be subsumed as part of another aspect. For the topic model with 50 topics we find 36 different concepts which are also subsumed to 22 unique aspects. Here, we reject the aspect "clientele" due to low support. Within the hotel domain, we observe a nearly perfect overlap between the aspects found by both topic models. The two aspects "hotel website" and "clientele" are not element of the intersection, but anyhow, as mentioned earlier both are rejected.

		100-topics model		50-topics model	
	aspect label	**# concepts**	**share**	**# concepts**	**share**
1	air conditioning	1	1.1%	1	1.5%
2	bathroom	3	2.7%	1	2.9%
3	bed	1	1.2%	1	1.5%
4	breakfast	2	3.0%	1	2.0%
5	check-in/out	2	2.8%	1	2.4%
6	cleanliness	2	2.3%	2	3.2%
7	decoration	1	1.6%	1	2.5%
8	dining	2	2.8%	1	3.7%
9	elevator	2	1.1%	1	2.0%
10	facility	3	2.1%	2	3.0%
11	internet	1	1.4%	1	1.7%
12	location	8	9.6%	3	8.7%
13	noise	2	3.3%	1	2.8%
14	parking	1	1.5%	1	1.9%
15	price	3	4.2%	2	3.8%
16	recreation	2	1.8%	1	2.1%
17	room	4	3.3%	2	4.1%
18	room amenities	3	2.5%	2	3.0%
19	security	1	0.9%	1	1.2%
20	service	10	13.4%	8	15.1%
21	view	1	1.3%	1	2.3%
		55	63.9%	35	71.4%

Table D.5.: Comparison of the 100-topics and 50-topics models for the hotel dataset. Numbers are rounded to a single digit after the decimal point.

In Table D.5 we show the distribution of the remaining 21 aspects for the topic models with 100 and 50 topics. Naturally the average amount of concepts related to a single identified aspect is higher in the more fine-grained 100-topics model. Here, the average number is 2.6 compared to a value of 1.7 for the 50-topics model. We find that the most important aspects are represented by a multitude of different concepts. For example the aspect "service" is composed of 10 distinct concepts. Some of these concepts are mixes of sentiment and aspect, that is refer to different ways of expressing positive or negative sentiment on this particular aspect. Others refer to different shades of the same aspect, but have too low support on their own (e.g., "concierge service", "front desk service", or "staff attitude" are all subsumed within the aspect service).

To further estimate the coherence between both models, we compute the average absolute difference between the shares of the individual aspects. We observe an average difference of 0.68 percentage points. It shows that the distribution of each aspect as computed by the two different models is very similar. In summary, we find that the 50-topics model identifies nearly the same set of relevant aspects, it produces less incoherent topics (0 compared to 5), and fewer irrelevant global topics (2 compared to 9). Further, as the 100-topics model does not provide more relevant information, but a higher number of topics induces more manual effort, we conclude that choosing the number of topics roughly between 50 to 70 seems reasonable[9].

For the digital camera domain the 100-topics model generates 60 relevant concepts out of which we initially determine 29 unique aspects. We further reject 3 aspects ("accessories", "file format", "printer dock") as they exhibit only a low support within the corpus and cannot be subsumed as part of another aspect. Table D.6 summarizes the distribution of the identified aspects for both models. Also here, the overlap of the models is very good, only five of the non rejected aspects differ ("face

[9] We also tried 30 and 40 topics, but these models failed to cover important aspects such as "noise", "view", or "air conditioning".

		100-topics model		50-topics model	
	aspect label	**# concepts**	**share**	**# concepts**	**share**
1	battery	3	4.4%	1	4.3%
2	built quality	2	1.4%	2	3.2%
3	connectivity	2	2.3%	1	2.7%
4	customer service	2	2.1%	1	1.8%
5	dimensions	3	2.9%	2	3.8%
6	ease of use	2	2.5%	1	3.0%
7	face detection	—	—	1	1.2%
8	features	2	1.6%	—	—
9	flash	2	1.0%	1	1.0%
10	focusing	1	1.0%	—	—
11	image stabilization	1	1.0%	1	2.0%
12	lens	2	2.2%	1	1.6%
13	low-light performance	2	3.5%	1	3.7%
14	macro mode	1	0.9%	1	2.4%
15	manual mode	2	1.9%	2	3.1%
16	memory	3	3.3%	1	3.0%
17	picture quality	10	11.9%	4	12.4%
18	price	3	4.3%	1	1.8%
19	scene modes	1	1.2%	1	2.4%
20	screen	1	2.5%	1	2.8%
21	software	1	1.3%	1	1.3%
22	speed	4	3.9%	2	3.7%
23	underwater capability	1	1.2%	1	1.7%
24	user interface	3	3.6%	1	2.1%
25	user manual	1	1.0%	—	—
26	video recording	1	2.1%	1	2.4%
27	zoom	1	1.4%	—	—
		57	66.1%	29	67.2%

Table D.6.: Comparison of the 100-topics and 50-topics models for the digital camera dataset. Numbers are rounded to a single digit after the decimal point.

detection", "features", "focusing", "user manual", and "zoom"). For the digital camera dataset the average absolute difference between the shares of the individual aspects (w.r.t. the intersection of both sets) is slightly higher with 0.77 percentage points. However, despite the minimally lower overlap and coherency, we believe that the camera dataset further underpins our earlier conclusions.

It is worth to note that the majority of coherent concepts that do not relate to local or global product aspects, represent one of the discourse functions as introduced in Chapter 5. We present two examples in Table D.7.

D.4. Summary

In this chapter, our goal was to examine methods to derive a relevant set of aspect-related topics from a large collection of customer reviews. For this purpose, we analyzed the utility of probabilistic topic modeling approaches. We learned that this type of modeling basically allows to discover "the main themes that pervade a large and otherwise unstructured collection of documents" [42]. We proposed to use the topic modeling technique for exploratory corpus analysis. Applying the algorithms to our large web crawls of customer reviews, we automatically gathered an initial set of relevant topics that we manually refined in a post-processing step. The primary purpose of the refinement step was to filter out domain irrelevant or incoherent topics. Our approach was to manually look at the keywords and phrases associated with a generated topic. Based on this inspection we decided on the topics' validity.

As the relevant literature was inconclusive with regard to the best document representation for topic modeling in the context of customer reviews, we experimented with two different methods. In

rank	word	phrase
1	stay (6674)	enjoyed stay (57)
2	hotel (3191)	back hotel (27)
3	back (2285)	stay hotel (26)
4	time (1664)	time stay (15)
5	return (1309)	recommend hotel (13)

(a) conclusion

rank	word	phrase
1	reviews (3487)	read reviews (109)
2	hotel (2326)	reading reviews (88)
3	read (1320)	trip advisor (30)
4	reading (879)	read review (23)
5	trip (632)	previous reviews (20)

(b) other reviews

Table D.7.: Top five words and phrases attributed to concepts that represent the two discourse functions "conclusion" and "other reviews" (hotel review dataset).

particular, we considered using the original review documents or splitting all documents into individual sentences. A further parameter was the number of topics that the topic modeling algorithm should extract. Although it is was unclear which amount of topics is reasonable for our task, it was not our goal to extensively experiment with this parameter. We only considered two different settings with 50 and 100 topics. Our main findings in this section were:

- The topic modeling approach is an effective tool for initially discovering the main themes in a set of customer reviews. If used as a tool for corpus exploration, the distinction between "global" and "local" topics is less important as global topics can be filtered out manually.
- The sentence-oriented representation is better suited to guide the decision process of a human annotator during the initial acquisition of coarse-grained product aspects. Obtained results are more coherent and better represent the coarse-grained product aspects.
- We find that the 50-topics model identifies nearly the same set of relevant aspects as the 100-topics model, produces less incoherent topics and fewer irrelevant global topics. It is further more convenient to analyze and induces less effort. We conclude that setting the number of topics between 50 and 70 is most reasonable.

E. Evaluation of Multi-Label and Hierarchical Classifiers

E.1. Evaluation of Multi-Label Classifiers

Whereas for traditional binary or multi-class classification (as well as for most information extraction problems), it is straightforward to calculate the standard evaluation metrics *accuracy*, *precision*, *recall*, and *f-measure*, it is not directly obvious for multi-label classification tasks. With traditional classification, the prediction for a single instance is either correct or incorrect. In contrast, by allowing for multiple labels, the prediction for a single instance may also be partially correct. For example, we may correctly label a sentence as discussing the "service" aspect of a hotel, but fail to identify that the reviewer also mentions the "dining" aspect. The prediction for this instance would only be partially correct. So the basic question with respect to evaluating multi-label classifiers is how to account for partially correct predictions.

Following Sorower [357], we can basically distinguish between *instance-based* and *label-based* evaluation methods. With instance-based evaluation, we calculate the specific metric (e.g., precision) for each instance in isolation and average over all instances in the test corpus. For a single instance, precision is defined as the number of correctly predicted labels divided by the amount of all predicted labels and recall is computed analogously. In other words, partial correctness of an instance directly translates to the precision/recall value for this particular instance. Another instance-based method is to calculate the *Hamming Loss* [333] which basically computes the average *Hamming distance* [159] between predicted and gold standard labels for an instance. The disadvantage of these instance-based methods is that the different classes cannot be evaluated separately. For example, we cannot tell how well a classifier performs in detecting mentions of the "picture quality" aspect in comparison to the "ease of use" aspect for digital camera reviews. We therefore opt for label-based evaluation. Here, we compute the evaluation metrics separately for each label (i.e., for each class) and then average over all labels. Adapting the notation of Sorower [357], we formalize the different label-based evaluation metrics as follows:

Let C be an evaluation corpus composed of n multi-label instances (x_i,Y_i), $1 \leq i \leq n$, $x_i \in \mathcal{X}$ with $Y_i \in \mathcal{P}(\mathcal{L})$ being the set of gold labels for instance x_i. Let $Z_i = c(x_i)$ be the set of labels predicted by a multi-label classifier for instance x_i. We further use the notation Y_i^λ to denote the projection of a gold label set Y_i to the label λ. That is $Y_i^\lambda = \begin{cases} \lambda, & \text{if } \lambda \in Y_i \\ \emptyset, & \text{if } \lambda \notin Y_i \end{cases}$. The notation Z_i^λ is defined analogously. Besides computing precision, recall, and f-measure for each individual class, we calculate total values by averaging over instances and classes. In particular, we report macro and micro-averaged results. Whereas the macro-average computes a simple average of the values obtained for each label, micro-averaging computes a global average by pooling the true/false positives of all classes within a single contingency table[1]. Equations (E.1) to (E.3) define the label-based computation of precision and recall for multi-label problems, as well as the macro and micro-averages for both metrics. With *λ-precision* and *λ-recall* we denote the precision and recall values for a single label/class λ:

$$\lambda\text{-precision } (P^\lambda) = \frac{\sum_{i=1}^{n} |Y_i^\lambda \cap Z_i^\lambda|}{\sum_{i=1}^{n} |Z_i^\lambda|}, \quad \lambda\text{-recall } (R^\lambda) = \frac{\sum_{i=1}^{n} |Y_i^\lambda \cap Z_i^\lambda|}{\sum_{i=1}^{n} |Y_i^\lambda|} \tag{E.1}$$

[1] For further information see for example Yang [448].

$$\textit{macro-precision} = \frac{1}{k}\sum_{i=1}^{k} P^i, \quad \textit{macro-recall} = \frac{1}{k}\sum_{i=1}^{k} R^i \tag{E.2}$$

$$\textit{micro-precision} = \frac{\sum_{j=1}^{k}\sum_{i=1}^{n}|Y_i^j \cap Z_i^j|}{\sum_{j=1}^{k}\sum_{i=1}^{n}|Z_i^j|}, \quad \textit{micro-recall} = \frac{\sum_{j=1}^{k}\sum_{i=1}^{n}|Y_i^j \cap Z_i^j|}{\sum_{j=1}^{k}\sum_{i=1}^{n}|Y_i^j|} \tag{E.3}$$

The F-measure is computed in the standard way, that is as harmonic mean of precision and recall. We report F-measure values for the individual classes as well as macro and micro-averaged results.

E.2. Evaluation of Hierarchical Classifiers

The simplest way to evaluate classifiers in presence of a hierarchy is to flatten the hierarchy and relabel the test corpus accordingly: Let $x \in \mathcal{X}$ be an instance of the test set and let $Y \in \mathcal{P}(\mathcal{L})$ be the corresponding set of correct labels (i.e., according to the manual annotation). Then, we relabel the instance with the set $Y' = Y \cup \{\lambda | \lambda \in \bigcup_{\lambda_i \in Y} \Uparrow (\lambda_i)\}$. In other words, x's new label set Y' contains all original labels and in addition the labels of all ancestor ("broader") categories that can be reached from any of the labels in Y. Take note that this kind of relabeling basically exploits the first property postulated in Definition 8.2. Having relabeled each instance of the test corpus, we can apply the standard label-based evaluation as defined earlier. The original instance x becomes a positive example for each of the categories in Y'.

The following example illustrates the approach. Consider the sentence "The bed was very comfortable and I had a nice view of downtown LA.". It may be annotated as expressing positive sentiment on the coarse-grained aspects "bed" and "view". In this case $Y = \{\textit{"bed"}, \textit{"view"}\}$. By definition of the product type taxonomy for the hotel domain, the aspect "bed" is subordinate to the topic "sleep quality" and the aspect "view" is a child of the concept "room". We thus have $Y' = \{\textit{"bed"}, \textit{"view"}\} \cup \{\textit{"sleep quality"}, \textit{"room"}\} = \{\textit{"sleep quality"}, \textit{"room"}, \textit{"bed"}, \textit{"view"}\}$. Thus, applying label-based evaluation, the original sentence becomes a positive example for each of the four categories.

Take note that, depending on the concrete approach of incorporating hierarchical information into a classifier, the simple evaluation method of flattening the hierarchy may not be reasonable. However, as our approaches are not affected, we will not go into further details and refer to [73, 122, 328]. More information on how we handle the hierarchical organization of coarse-grained product aspects will be provided alongside the different approaches we discuss in the following sections.

F. Lists of Polar and Neutral Seed Words

F.1. Seed Words with Positive Prior Polarity

1. amazing_JJ
2. awesome_JJ
3. cute_JJ
4. decent_JJ
5. excellent_JJ
6. fantastic_JJ
7. fine_JJ
8. friendly_JJ
9. good_JJ
10. great_JJ
11. happy_JJ
12. impressive_JJ
13. lovely_JJ
14. nice_JJ
15. perfect_JJ
16. positive_JJ
17. attraction_NN
18. awe_NN
19. awesomeness_NN
20. compliment_NN
21. excellence_NN
22. goodness_NN
23. impressiveness_NN
24. loveliness_NN
25. masterpiece_NN
26. pleasure_NN
27. recommendation_NN
28. decently_RB
29. excellently_RB
30. flawlessly_RB
31. fortunately_RB
32. gorgeously_RB
33. luckily_RB
34. magnificently_RB
35. marvelously_RB
36. nicely_RB
37. perfectly_RB
38. pretty_RB
39. splendidly_RB
40. superbly_RB
41. terrifically_RB
42. well_RB
43. wonderfully_RB
44. amaze_VB
45. appreciate_VB
46. attract_VB
47. captivate_VB
48. enjoy_VB
49. fascinate_VB
50. impress_VB
51. improve_VB
52. like_VB
53. love_VB
54. recommend_VB

F.2. Seed Words with Negative Prior Polarity

1. awful_JJ
2. bad_JJ
3. cracked_JJ
4. dirty_JJ
5. disappointed_JJ
6. disappointing_JJ
7. evil_JJ
8. frustrating_JJ
9. horrific_JJ
10. ill_JJ
11. inferior_JJ
12. nasty_JJ
13. negative_JJ
14. poor_JJ
15. sick_JJ
16. terrible_JJ
17. ugly_JJ
18. unclean_JJ
19. unfortunate_JJ
20. awfulness_NN
21. awkwardness_NN
22. complaint_NN
23. con_NN
24. concern_NN
25. critic_NN
26. criticism_NN
27. disadvantage_NN
28. disappointment_NN
29. downer_NN
30. downfall_NN
31. downside_NN
32. drawback_NN
33. error_NN
34. fault_NN
35. flaw_NN
36. frustration_NN
37. gripe_NN
38. hate_NN
39. horror_NN
40. issue_NN
41. lack_NN
42. limitation_NN
43. mess_NN
44. problem_NN
45. regret_NN
46. terribleness_NN
47. weakness_NN
48. awfully_RB
49. badly_RB
50. disgustingly_RB
51. horribly_RB
52. ridiculously_RB
53. rottenly_RB
54. sadly_RB
55. terribly_RB
56. unfortunately_RB
57. unhappily_RB

58. unluckily_RB
59. bother_VB
60. break_VB
61. complain_VB
62. disappoint_VB
63. dislike_VB
64. frustrate_VB
65. hate_VB
66. suck_VB

F.3. Neutral Words

1. about_JJ
2. according_JJ
3. achromatic_JJ
4. actual_JJ
5. adopted_JJ
6. affected_JJ
7. after_JJ
8. all_JJ
9. alone_JJ
10. another_JJ
11. any_JJ
12. available_JJ
13. away_JJ
14. back_JJ
15. beforehand_JJ
16. beginning_JJ
17. behind_JJ
18. bittie_JJ
19. bitty_JJ
20. black_JJ
21. blue_JJ
22. both_JJ
23. brief_JJ
24. cardinal_JJ
25. certain_JJ
26. chromatic_JJ
27. different_JJ
28. done_JJ
29. down_JJ
30. due_JJ
31. each_JJ
32. eight_JJ
33. eighty_JJ
34. enceinte_JJ
35. even_JJ
36. every_JJ
37. ex_JJ
38. expectant_JJ
39. far_JJ
40. fifth_JJ
41. first_JJ
42. five_JJ
43. following_JJ
44. former_JJ
45. found_JJ
46. four_JJ
47. full_JJ
48. further_JJ
49. giving_JJ
50. go_JJ
51. gone_JJ
52. gravid_JJ
53. green_JJ
54. here_JJ
55. home_JJ
56. hundred_JJ
57. i_JJ
58. important_JJ
59. in_JJ
60. inward_JJ
61. just_JJ
62. last_JJ
63. later_JJ
64. latter_JJ
65. less_JJ
66. likely_JJ
67. looking_JJ
68. made_JJ
69. million_JJ
70. more_JJ
71. most_JJ
72. much_JJ
73. must_JJ
74. near_JJ
75. neither_JJ
76. neutral_JJ
77. new_JJ
78. next_JJ
79. nine_JJ
80. ninety_JJ
81. no_JJ
82. none_JJ
83. noted_JJ
84. off_JJ
85. ok_JJ
86. old_JJ
87. on_JJ
88. one_JJ
89. only_JJ
90. ordinal_JJ
91. other_JJ
92. otherwise_JJ
93. out_JJ
94. outside_JJ
95. over_JJ
96. overall_JJ
97. owing_JJ
98. own_JJ
99. particular_JJ
100. past_JJ
101. placed_JJ
102. plus_JJ
103. poorly_JJ
104. possible_JJ
105. pregnant_JJ
106. present_JJ
107. recent_JJ
108. red_JJ
109. regardless_JJ
110. regular_JJ
111. related_JJ
112. said_JJ
113. same_JJ
114. sec_JJ
115. seeing_JJ
116. seeming_JJ
117. self_JJ
118. sent_JJ
119. seven_JJ
120. several_JJ
121. shed_JJ
122. similar_JJ
123. six_JJ
124. some_JJ
125. sometime_JJ
126. specified_JJ
127. still_JJ
128. such_JJ
129. teensy_JJ
130. teentsy_JJ
131. teeny_JJ
132. wee_JJ
133. weensy_JJ
134. weeny_JJ
135. white_JJ
136. yellow_JJ
137. a_NN
138. act_NN
139. affect_NN
140. am_NN
141. an_NN
142. are_NN
143. as_NN
144. aside_NN
145. asking_NN
146. at_NN
147. back_NN
148. be_NN
149. begin_NN
150. beginning_NN
151. behind_NN
152. being_NN
153. brief_NN
154. can_NN
155. cause_NN
156. come_NN
157. date_NN
158. deal_NN
159. do_NN
160. due_NN
161. effect_NN
162. eight_NN
163. eighty_NN
164. end_NN
165. entrance_NN
166. even_NN
167. ex_NN
168. far_NN
169. fifth_NN
170. first_NN
171. five_NN
172. fix_NN
173. following_NN
174. former_NN
175. forth_NN
176. found_NN
177. four_NN
178. get_NN
179. getting_NN
180. give_NN
181. giving_NN
182. go_NN
183. have_NN
184. he_NN
185. here_NN
186. hereafter_NN
187. hi_NN
188. home_NN
189. hundred_NN
190. i_NN
191. in_NN
192. inc_NN
193. index_NN
194. information_NN
195. invention_NN
196. it_NN
197. keep_NN
198. kg_NN
199. km_NN
200. know_NN
201. last_NN
202. latter_NN
203. let_NN
204. line_NN
205. look_NN
206. looking_NN
207. lot_NN
208. make_NN
209. mass_NN
210. may_NN
211. me_NN
212. mean_NN
213. means_NN
214. meantime_NN
215. meanwhile_NN
216. mg_NN
217. million_NN
218. miss_NN
219. ml_NN
220. more_NN
221. mr_NN
222. mrs_NN
223. much_NN
224. mug_NN
225. must_NN
226. na_NN
227. name_NN
228. nay_NN
229. nd_NN
230. necessary_NN
231. nine_NN
232. ninety_NN
233. no_NN
234. nobody_NN
235. none_NN
236. nothing_NN
237. now_NN
238. nowhere_NN
239. oh_NN
240. ok_NN
241. okay_NN
242. old_NN
243. one_NN
244. or_NN
245. out_NN
246. outside_NN
247. over_NN
248. overall_NN
249. page_NN
250. part_NN
251. particular_NN
252. past_NN
253. plenty_NN
254. plus_NN
255. possible_NN
256. present_NN
257. put_NN
258. re_NN
259. recent_NN
260. ref_NN
261. research_NN
262. run_NN
263. same_NN
264. saw_NN
265. say_NN
266. saying_NN
267. sec_NN
268. section_NN
269. see_NN
270. seeing_NN
271. self_NN
272. sent_NN
273. seven_NN
274. shed_NN
275. show_NN
276. six_NN
277. so_NN
278. somebody_NN
279. someone_NN
280. somewhere_NN
281. state_NN
282. still_NN
283. stop_NN
284. sub_NN
285. sup_NN
286. about_RB
287. accordingly_RB
288. across_RB
289. actually_RB
290. after_RB
291. afterwards_RB
292. again_RB
293. all_RB
294. almost_RB
295. alone_RB
296. along_RB
297. already_RB
298. also_RB
299. altogether_RB
300. always_RB
301. any_RB
302. anyhow_RB
303. anymore_RB
304. anyway_RB
305. anyways_RB
306. anywhere_RB
307. apparently_RB
308. approximately_RB
309. around_RB
310. as_RB
311. aside_RB
312. astronomically_RB
313. away_RB
314. back_RB
315. before_RB
316. beforehand_RB
317. behind_RB
318. besides_RB
319. between_RB
320. beyond_RB
321. briefly_RB
322. but_RB
323. by_RB
324. certainly_RB
325. consequently_RB
326. considerably_RB
327. critically_RB
328. deeply_RB
329. diametrically_RB
330. divisively_RB
331. down_RB
332. downwards_RB
333. due_RB
334. each_RB
335. either_RB
336. elsewhere_RB
337. emphatically_RB
338. enormously_RB
339. entirely_RB
340. especially_RB
341. ever_RB
342. everywhere_RB
343. evidently_RB
344. exactly_RB
345. exceedingly_RB
346. explicitly_RB
347. extraordinarily_RB
348. extremely_RB
349. far_RB
350. fervently_RB
351. fervidly_RB
352. finally_RB
353. first_RB
354. forgetfully_RB
355. formerly_RB
356. forth_RB
357. fundamentally_RB
358. further_RB
359. furthermore_RB
360. greatly_RB
361. hardly_RB
362. heavily_RB
363. hence_RB
364. here_RB
365. hereafter_RB

366. hereby_RB
367. herein_RB
368. hereupon_RB
369. hither_RB
370. home_RB
371. hopefully_RB
372. however_RB
373. ideally_RB
374. immediately_RB
375. imminently_RB
376. improbably_RB
377. impulsively_RB
378. in_RB
379. incisively_RB
380. incredibly_RB
381. indeed_RB
382. indescribably_RB
383. indubitably_RB
384. inextricably_RB
385. instead_RB
386. inward_RB
387. irrefutably_RB
388. knowingly_RB
389. last_RB
390. lastly_RB
391. lately_RB
392. later_RB
393. latterly_RB
394. likely_RB
395. likewise_RB
396. mainly_RB
397. maybe_RB
398. meantime_RB
399. meanwhile_RB
400. merely_RB
401. more_RB
402. moreover_RB
403. most_RB
404. mostly_RB
405. much_RB
406. namely_RB
407. nay_RB
408. near_RB
409. nearly_RB
410. necessarily_RB
411. needs_RB
412. never_RB
413. nevertheless_RB
414. new_RB
415. next_RB
416. no_RB
417. non_RB
418. none_RB
419. nonetheless_RB
420. normally_RB
421. not_RB
422. notably_RB
423. nothing_RB
424. now_RB
425. nowhere_RB
426. obviously_RB
427. off_RB
428. often_RB
429. ok_RB
430. okay_RB
431. on_RB
432. once_RB
433. only_RB
434. otherwise_RB
435. ought_RB
436. out_RB
437. outside_RB
438. over_RB
439. part_RB
440. particularly_RB
441. past_RB
442. perhaps_RB
443. perspicuously_RB
444. plainly_RB
445. please_RB
446. plenty_RB
447. possibly_RB
448. potentially_RB
449. predominantly_RB
450. presumably_RB
451. previously_RB
452. primarily_RB
453. probably_RB
454. quickly_RB
455. quite_RB
456. radically_RB
457. rather_RB
458. realistically_RB
459. really_RB
460. recently_RB
461. relatively_RB
462. remarkably_RB
463. respectively_RB
464. right_RB
465. satisfactorily_RB
466. scarcely_RB
467. seemingly_RB
468. seriously_RB
469. significantly_RB
470. similarly_RB
471. so_RB
472. some_RB
473. somehow_RB
474. sometime_RB
475. sometimes_RB
476. somewhat_RB
477. somewhere_RB
478. soon_RB
479. specifically_RB
480. staggeringly_RB
481. still_RB
482. strictly_RB
483. strikingly_RB
484. strongly_RB
485. stupendously_RB
486. such_RB
487. super_RB
488. surely_RB
489. surprisingly_RB
490. therefore_RB
491. though_RB
492. thus_RB
493. thusly_RB
494. tolerably_RB
495. tolerantly_RB
496. tremendously_RB
497. truly_RB
498. ultimately_RB
499. unbelievably_RB
500. undoubtedly_RB
501. unusually_RB
502. utterly_RB
503. vehemently_RB
504. act_VB
505. affect_VB
506. announce_VB
507. argue_VB
508. arise_VB
509. ask_VB
510. bang_VB
511. be_VB
512. become_VB
513. bed_VB
514. begin_VB
515. believe_VB
516. bonk_VB
517. borrow_VB
518. brief_VB
519. bring_VB
520. buy_VB
521. call_VB
522. can_VB
523. cause_VB
524. change_VB
525. clean_VB
526. come_VB
527. contain_VB
528. count_VB
529. cut_VB
530. dance_VB
531. date_VB
532. do_VB
533. down_VB
534. draw_VB
535. drink_VB
536. drive_VB
537. eat_VB
538. eff_VB
539. effect_VB
540. end_VB
541. even_VB
542. except_VB
543. explain_VB
544. feel_VB
545. fill_VB
546. find_VB
547. finish_VB
548. fix_VB
549. fly_VB
550. forget_VB
551. found_VB
552. fuck_VB
553. further_VB
554. get_VB
555. give_VB
556. go_VB
557. have_VB
558. hear_VB
559. home_VB
560. hump_VB
561. index_VB
562. jazz_VB
563. keep_VB
564. know_VB
565. last_VB
566. learn_VB
567. let_VB
568. line_VB
569. listen_VB
570. live_VB
571. look_VB
572. make_VB
573. mean_VB
574. mug_VB
575. name_VB
576. near_VB
577. need_VB
578. obtain_VB
579. off_VB
580. okay_VB
581. open_VB
582. organize_VB
583. ought_VB
584. out_VB
585. own_VB
586. page_VB
587. part_VB
588. pay_VB
589. play_VB
590. please_VB
591. present_VB
592. put_VB
593. rain_VB
594. read_VB
595. reply_VB
596. research_VB
597. run_VB
598. saw_VB
599. say_VB
600. screw_VB
601. section_VB
602. see_VB
603. seem_VB
604. sell_VB
605. send_VB
606. shed_VB
607. show_VB
608. sign_VB
609. sing_VB
610. sit_VB
611. sleep_VB
612. smoke_VB
613. speak_VB
614. specify_VB
615. spell_VB
616. spend_VB
617. stand_VB
618. state_VB
619. still_VB
620. stop_VB
621. study_VB
622. sub_VB
623. sup_VB
624. take_VB
625. talk_VB
626. teach_VB
627. tell_VB
628. think_VB
629. translate_VB
630. travel_VB
631. try_VB
632. type_VB
633. use_VB
634. wait_VB
635. want_VB
636. watch_VB
637. work_VB
638. write_VB

G. Zusammenfassung

Kundenrezensionen im Internet spielen heutzutage eine wichtige Rolle bei unseren alltäglichen Kaufentscheidungen. Ebenso sind die unzähligen Produktbewertungen von großem Wert für Unternehmen, beispielsweise zur Marktforschung, Trendanalyse oder Qualitätssicherung. In vielen Fällen wird allerdings das beschriebene Informationsbedürfnis von einem Informationsüberfluss überdeckt. Für populäre Produkte existieren oft tausende Rezensionen und eine individuelle Sichtung ist daher keine Option.

In dieser Dissertation befassen wir uns damit, wie man die meinungsbehaftete, in natürlicher Sprache vorliegende Information in Kundenrezensionen modellieren und automatisiert zusammenfassen kann. Im Speziellen untersuchen wir Verfahren zur *aspekt-orientierten Meinungsanalyse* von Kundenbewertungen. Ziel dieses Textanalyseverfahrens ist es automatisiert alle bewerteten Produkteigenschaften in einer Rezension zu erfassen und die jeweilig geäußerte Meinungsrichtung zu bestimmen (z.B. positiv gegenüber negativ). Viele Systeme zur automatisierten Textanalyse beruhen auf speziell entwickelten Wissensdatenbanken oder setzen (im Falle maschineller Lernverfahren) die Existenz von Trainingsdaten voraus. Als ein übergeordnetes Thema dieser Arbeit betrachten wir daher so genannte *Distant Supervision (DS) Ansätze* die es ermöglichen den manuellen Aufwand bei der Erstellung der genannten Ressourcen zu verringern. Wir konzentrieren uns auf die zwei wichtigsten Teilprobleme der aspekt-orientierten Meinungsanalyse: (i) die Identifikation von relevanten Produktaspekten und (ii) die Erkennung und Bewertung von Meinungsäußerungen. Wir betrachten beide Teilprobleme jeweils auf Wort-/Phrasen- und auf Satzebene. Für beide Detailstufen untersuchen wir jeweils lexikonbasierte Ansätze und Verfahren des überwachten maschinellen Lernens. Ebenso experimentieren wir mit verschiedenen DS-Techniken.

Die Aspekterkennung auf Wortebene erachten wir als ein Terminologieextraktionsproblem. Auf Satzebene modellieren wir die Problemstellung als ein Multi-Label Textklassifikationsproblem. Bezüglich der Sentimentanalyse untersuchen wir Verfahren zur automatischen Erstellung von Sentimentlexika und zur Sentimentklassifikation. Wir evaluieren unsere Ansätze im Detail (inklusive aufschlussreicher Fehleranalysen), wenn möglich, im Vergleich zu anderen relevanten Methoden. Insbesondere zeigen unsere Ergebnisse, dass wir mit den präsentierten Distant Supervision Methoden erfolgreich den manuellen Aufwand bei der Erstellung von notwendigen Ressourcen reduzieren können. Generell ermöglichen es die Verfahren sehr große Mengen an Trainingsdaten zu extrahieren (in unserem Fall beträgt der Unterschied zu den manuell annotierten Datensätzen zwei bis drei Größenordnungen). Die vorgeschlagenen Verfahren können daher vorteilhaft im Rahmen von Systemen zur aspekt-orientierten Sentimentanalyse von Kundenrezensionen eingesetzt werden.

Erklärung

Ich versichere hiermit, dass ich die vorliegende Dissertation selbständig verfasst habe und alle Hilfsmittel und Hilfen als solche gekennzeichnet sind. Die Arbeit wurde bei keiner anderen Prüfungsbehörde eingereicht.

Berlin, den 16. Juli 2013

Jürgen Broß

Bibliography

[1] Steven P. Abney. *Semisupervised Learning in Computational Linguistics*. Computer Science and Data Analysis Series. Chapman & Hall/CRC, 2008. URL http://books.google.de/books?id=VCd67cGB_rAC. (Cited on page 157.)

[2] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. Sentiment analysis of Twitter data. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2021109.2021114. (Cited on pages 213 and 217.)

[3] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM Conference on Digital Libraries*, pages 85–94, 2000. (Cited on page 34.)

[4] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=645920.672836. (Cited on pages 94 and 97.)

[5] Alfred V. Aho and Margaret J. Corasick. Efficient string matching: An aid to bibliographic search. *Communications of the ACM*, 18(6):333–340, June 1975. URL http://doi.acm.org/10.1145/360825.360855. (Cited on pages 221 and 304.)

[6] Alan Akbik and Jürgen Bross. Wanderlust: Extracting semantic relations from natural language text using dependency grammar patterns. In *Proceedings of the 2009 Semantic Search Workshop at the 18th International World Wide Web Conference*, SemSearch '09, pages 6–15, Madrid, Spain, April 2009. URL http://ceur-ws.org/Vol-491/semse2009_7.pdf. (Cited on page 30.)

[7] Alan Akbik and Alexander Löser. KrakeN: N-ary facts in open information extraction. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 52–56, Montréal, Canada, June 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W12-3010.pdf. (Cited on page 30.)

[8] AlchemyAPI. Product Website: Alchemy API Sentiment Analysis. URL http://www.alchemyapi.com/api/sentiment/. [Online; accessed 12/2012]. (Cited on page 26.)

[9] Ben Allison. Sentiment detection using lexically-based classifiers. In *Text, Speech and Dialogue*, volume 5246 of *Lecture Notes in Computer Science*, pages 21–28. Springer Berlin / Heidelberg, 2008. URL http://dx.doi.org/10.1007/978-3-540-87391-4_5. (Cited on page 203.)

[10] Cecilia O. Alm, Dan Roth, and Richard Sproat. Emotions from text: Machine learning for text-based emotion prediction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1220575.1220648. (Cited on page 13.)

[11] Saima Aman and Stan Szpakowicz. Identifying expressions of emotion in text. In *Text, Speech and Dialogue*, volume 4629 of *Lecture Notes in Computer Science*, pages 196–205. Springer Berlin

Heidelberg, 2007. URL http://dx.doi.org/10.1007/978-3-540-74628-7_27. (Cited on page 13.)

[12] S. Ananiadou and J. McNaught. *Text Mining for Biology and Biomedicine*. Artech House Bioinformatics Series. Artech House, 2006. URL http://books.google.de/books?id=xkNRAAAAMAAJ. (Cited on page 100.)

[13] Alina Andreevskaia and Sabine Bergler. Mining WordNet for a fuzzy sentiment: Sentiment tag extraction from WordNet glosses. In *Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics*, EACL '06, April 2006. URL http://www.aclweb.org/anthology/E/E06/E06-1027.pdf. (Cited on page 172.)

[14] Alina Andreevskaia and Sabine Bergler. When specialists and generalists work together: Overcoming domain dependence in sentiment tagging. In *Proceedings of the 46th Annual Meeting of the ACL: Human Language Technologies*, ACL-HLT '08, pages 290–298, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P08/P08-1034.pdf. (Cited on pages 169, 206 and 264.)

[15] Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2:343–370, 1988. URL http://dx.doi.org/10.1023/A%3A1022873112823. (Cited on page 264.)

[16] Shilpa Arora, Elijah Mayfield, Carolyn Penstein Rosé, and Eric Nyberg. Sentiment classification using automatically extracted subgraph features. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, Los Angeles, CA, June 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W10-0216.pdf. (Cited on pages 207, 213, 214 and 217.)

[17] Nicholas Asher, Farah Benamara, and Yvette Y. Mathieu. Distilling opinion in discourse: A preliminary study. In *Coling 2008: Companion volume: Posters*, pages 7–10, Manchester, UK, August 2008. Coling 2008 Organizing Committee. URL http://www.aclweb.org/anthology/C08-2002.pdf. (Cited on page 44.)

[18] Nicholas Asher, Farah Benamara, and Yvette Y. Mathieu. Appraisal of opinion expressions in discourse. *Lingvisticae Investigationes*, 32:279–292, December 2009. URL http://www.ingentaconnect.com/content/jbp/li/2009/00000032/00000002/art00011. (Cited on page 44.)

[19] Attensity. Technical Info: Attensity Analytics. URL http://www.attensity.com/assets/Accuracy-MattersMay2011.pdf. [Online; accessed 12/2012]. (Cited on page 26.)

[20] Anthony Aue and Michael Gamon. Customizing sentiment classifiers to new domains: A case study. Technical report, Microsoft Research, 2005. URL http://research.microsoft.com/pubs/65430/new_domain_sentiment.pdf. (Cited on pages 206 and 264.)

[21] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Multi-facet rating of product reviews. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, volume 5478 of *ECIR '09*, pages 461–472, Berlin, Heidelberg, 2009. Springer-Verlag. URL http://dx.doi.org/10.1007/978-3-642-00958-7_41. (Cited on pages 19, 207, 211 and 217.)

[22] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the 7th Conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf. (Cited on pages 24, 172, 173, 174 and 175.)

[23] Alexandra Balahur, Zornitsa Kozareva, and Andrés Montoyo. Determining the polarity and source of opinions expressed in political debates. In *Computational Linguistics and Intelligent Text Processing*, volume 5449 of *Lecture Notes in Computer Science*, pages 468–480. Springer Berlin / Heidelberg, 2009. URL http://dx.doi.org/10.1007/978-3-642-00382-0_38. (Cited on page 2.)

[24] Alexandra Balahur, Ralf Steinberger, Mijail Kabadjov, Vanni Zavarella, Erik van der Goot, Matina Halkia, Bruno Pouliquen, and Jenya Belyaeva. Sentiment analysis in the news. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA). (Cited on page 203.)

[25] Jason Baldridge and Miles Osborne. Active learning and the total cost of annotation. In *Proceedings of EMNLP 2004*, pages 9–16, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL http://acl.ldc.upenn.edu/acl2004/emnlp/pdf/Baldridge.pdf. (Cited on pages 35 and 264.)

[26] Carmen Banea, Rada Mihalcea, and Janyce Wiebe. A bootstrapping method for building subjectivity lexicons for languages with scarce resources. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA). URL http://www.cse.unt.edu/~rada/papers/banea.lrec08.pdf. (Cited on pages 34 and 170.)

[27] Ann Banfield. *Unspeakable Sentences: Narration and Representation in the Language of Fiction*. Routledge and Kegan Paul, 1982. (Cited on page 54.)

[28] Luciano Barbosa, Ravi Kumar, Bo Pang, and Andrew Tomkins. For a few dollars less: Identifying review pages sans human labels. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACLHLT '09, pages 494–502, Boulder, Colorado, June 2009. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N/N09/N09-1056.pdf. (Cited on pages 13, 20, 32, 206 and 264.)

[29] F. Baron and G. Hirst. Collocations as cues to semantic orientation. In *AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, 2004. URL http://www.cs.utoronto.ca/pub/gh/Baron+Hirst-2003.pdf. (Cited on page 172.)

[30] Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. The WaCky Wide Web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43:209–226, 2009. URL http://dx.doi.org/10.1007/s10579-009-9081-4. (Cited on page 117.)

[31] Roberto Basili, Alessandro Moschitti, and Maria T. Pazienza. NLP-driven IR: Evaluating performances over a text classification task. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, volume 2 of *IJCAI'01*, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1642194.1642266. (Cited on page 149.)

[32] Leonard E. Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 37(6), 1966. URL http://www.jstor.org/stable/2238772. (Cited on pages 22 and 97.)

[33] A. Beal and J. Strauss. *Radically Transparent: Monitoring and Managing Reputations Online*. John Wiley & Sons, 2009. (Cited on page 2.)

[34] Paul N. Bennett, David M. Chickering, and Anton Mityagin. Learning consensus opinion: Mining data from a labeling game. In *WWW '09: Proceedings of the 18th International Conference on World Wide Web*, pages 121–130, New York, NY, USA, 2009. ACM. URL http://dx.doi.org/10.1145/1526709.1526727. (Cited on page 34.)

[35] A. Bergamo and L. Torresani. Exploiting weakly-labeled web images to improve object classification: A domain adaptation approach. *Advances in Neural Information Processing Systems (NIPS)*, 2010. URL http://books.nips.cc/papers/files/nips23/NIPS2010_0093.pdf. (Cited on page 29.)

[36] S. Bethard, H. Yu, A. Thornton, V. Hatzivassiloglou, and D. Jurafsky. Automatic extraction of opinion propositions and their holders. In *2004 AAAI Spring Symposium on Exploring Attitude and Affect in Text*, 2004. URL http://www.aaai.org/Papers/Symposia/Spring/2004/SS-04-07/SS04-07-005.pdf. (Cited on page 14.)

[37] Steven Bethard, Hong Yu, Ashley Thornton, Vasileios Hatzivassiloglou, and Dan Jurafsky. Extracting opinion propositions and opinion holders using syntactic and lexical cues. In *Computing Attitude and Affect in Text: Theory and Applications*, volume 20 of *The Information Retrieval Series*, pages 125–141. Springer Netherlands, 2006. URL http://dx.doi.org/10.1007/1-4020-4102-0_11. (Cited on page 19.)

[38] Adrian Bickerstaffe and Ingrid Zukerman. A hierarchical classifier applied to multi-way sentiment detection. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1873781.1873789. (Cited on pages 19 and 207.)

[39] C. M. Bishop. *Pattern Recognition And Machine Learning*. Information Science and Statistics. Springer, 2006. (Cited on pages 22 and 156.)

[40] S. Blair-Goldensohn, K. Hannan, R. McDonald, T. Neylon, G. A. Reis, and J. Reynar. Building a sentiment summarizer for local service reviews. In *WWW Workshop on NLP in the Information Explosion Era*, 2008. URL http://research.google.com/pubs/archive/34368.pdf. (Cited on pages 14, 22, 24, 26, 64, 97, 157, 171, 173, 175, 176, 177, 178, 187, 190, 200, 207, 211, 212, 217, 230, 231, 235 and 263.)

[41] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Semi-supervised novelty detection. *Journal of Machine Learning Research*, 11, December 2010. URL http://dl.acm.org/citation.cfm?id=1756006.1953028. (Cited on page 226.)

[42] David M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, April 2012. URL http://doi.acm.org/10.1145/2133806.2133826. (Cited on pages 23, 313 and 320.)

[43] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003. URL http://dl.acm.org/citation.cfm?id=944937. (Cited on page 313.)

[44] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P07-1056.pdf. (Cited on pages 31, 206 and 264.)

[45] Kenneth Bloom, Navendu Garg, and Shlomo Argamon. Extracting appraisal expressions. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 308–315, Rochester, New York, April 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N/N07/N07-1039.pdf. (Cited on pages 39, 55, 144 and 203.)

[46] BNC Consortium. The British National Corpus, version 3, 2007. URL http://www.natcorp.ox.ac.uk. (Cited on pages 96 and 117.)

[47] J. Bollen, A. Pepe, and H. Mao. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*, pages 450–453, 2011. URL http://arxiv.org/pdf/0911.1583. (Cited on pages 2 and 13.)

[48] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1 – 8, 2011. URL http://www.sciencedirect.com/science/article/pii/S187775031100007X. (Cited on pages 2 and 171.)

[49] Kevin W. Bowyer, Nitesh V. Chawla, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. URL http://dx.doi.org/10.1613/jair.953. (Cited on page 157.)

[50] M. M. Bradley and P. J. Lang. Affective norms for english words (ANEW): Instruction manual and affective ratings. Technical report, The Center for Research in Psychophysiology, University of Florida, 1999. URL http://www.uvm.edu/~pdodds/files/papers/others/1999/bradley1999a.pdf. (Cited on page 171.)

[51] S. R. K. Branavan, Harr Chen, Jacob Eisenstein, and Regina Barzilay. Learning document-level semantic properties from free-text annotations. In *Proceedings of ACL-08: HLT*, pages 263–271, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology-new/P/P08/P08-1031.pdf. (Cited on page 32.)

[52] Eric Breck and Claire Cardie. Playing the telephone game: Determining the hierarchical structure of perspective and speech expressions. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1220355.1220373. (Cited on page 19.)

[53] Eric Breck, Yejin Choi, and Claire Cardie. Identifying expressions of opinion in context. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, IJCAI-2007, Hyderabad, India, January 2007. URL http://www.aaai.org/Papers/IJCAI/2007/IJCAI07-431.pdf. (Cited on pages 24 and 208.)

[54] C. E. Brodley and M. A. Friedl. Identifying mislabeled training data. *Journal of Artificial Intelligence*, 11:131–167, 1999. URL http://www.jair.org/media/606/live-606-1803-jair.pdf. (Cited on page 264.)

[55] Samuel Brody and Noemie Elhadad. An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 804–812, Los Angeles, California, June 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N10-1122. (Cited on pages 46, 157, 314 and 315.)

[56] Jürgen Bross and Heiko Ehrig. Generating a context-aware sentiment lexicon for aspect-based product review mining. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '10, pages 435–439, Washington, DC, USA, 2010. IEEE Computer Society. URL http://dx.doi.org/10.1109/WI-IAT.2010.56. (Cited on page 169.)

[57] Rebecca F. Bruce and Janyce M. Wiebe. Recognizing subjectivity: A case study in manual tagging. *Natural Language Engineering*, 5(2):187–205, June 1999. URL http://dx.doi.org/10.1017/S1351324999002181. (Cited on page 53.)

[58] Chris Buckley and Ellen M. Voorhees. Evaluating evaluation measure stability. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 33–40, New York, NY, USA, 2000. ACM. URL http://doi.acm.org/10.1145/345508.345543. (Cited on page 115.)

[59] Alexander Budanitsky and Graeme Hirst. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and other Lexical Resources, 2nd Meeting of the North American Chapter of the Association for Computational Linguistics*, 2001. URL ftp://ftp.db.toronto.edu/pub/gh/Budanitsky+Hirst-2001.pdf. (Cited on pages 145 and 172.)

[60] Paul Buitelaar and Bernardo Magnini. Ontology learning from text: An overview. In *Ontology Learning from Text: Methods, Applications and Evaluation*, pages 3–12. IOS Press, 2005. URL http://books.google.de/books?id=gikMilZMFMMC. (Cited on pages 23, 40, 94, 100, 139 and 145.)

[61] Anais Cadilhac, Farah Benamara, and Nathalie Aussenac-Gilles. Ontolexical resources for feature-based opinion mining: A case study. In *Proceedings of the 6th Workshop on Ontologies and Lexical Resources* , pages 77–86, Beijing, China, August 2010. Coling 2010 Organizing Committee. URL http://www.aclweb.org/anthology/W10-3309.pdf. (Cited on page 145.)

[62] Lijuan Cai and Thomas Hofmann. Hierarchical document categorization with support vector machines. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management*, CIKM '04, New York, NY, USA, 2004. ACM. URL http://doi.acm.org/10.1145/1031171.1031186. (Cited on page 147.)

[63] Jaime G. Carbonell. *Subjective Understanding: Computer Models of Belief Systems*. PhD thesis, Yale University, New Haven, CT, USA, 1979. (Cited on page 11.)

[64] Giuseppe Carenini, Raymond T. Ng, and Ed Zwart. Extracting knowledge from evaluative text. In *K-CAP '05: Proceedings of the 3rd International Conference on Knowledge Capture*, pages 11–18, New York, NY, USA, 2005. ACM. URL http://dx.doi.org/10.1145/1088622.1088626. (Cited on page 145.)

[65] Maria F. Caropreso, Stan Matwin, and Fabrizio Sebastiani. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In *Text Databases and Document Management*. IGI Publishing, Hershey, PA, USA, 2001. URL http://dl.acm.org/citation.cfm?id=374247.374254. (Cited on page 149.)

[66] Bob Carpenter. Lazy sparse stochastic gradient descent for regularized multinomial logistic regression, 2008. URL http://lingpipe.files.wordpress.com/2008/04/lazysgdregression.pdf. (Cited on page 150.)

[67] Ronald Carter and Michael McCarthy. *Cambridge Grammar of English*. Cambridge University Press, 2006. (Cited on pages 51 and 68.)

[68] S. Cerini, V. Compagnoni, A. Demontis, M. Formentelli, and G. Gandini. Micro-WNOp: A gold standard for the evaluation of automatically compiled lexical resources for opinion mining. *Language Resources and Linguistic Theory: Typology, Second Language Acquisition, English Linguistics*, pages 200–210, 2007. URL http://www-3.unipv.it/wnop/. (Cited on page 175.)

[69] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, May 2011. URL http://doi.acm.org/10.1145/1961189.1961199. (Cited on pages 226 and 228.)

[70] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006. ISBN 9780262033589. URL http://www.kyb.tuebingen.mpg.de/ssl-book. (Cited on pages 29 and 34.)

[71] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. Adaptive Computation and Machine Learning. Mit Press, 2010. URL http://books.google.de/books?id=zHAOQgAACAAJ. (Cited on page 157.)

[72] Nitesh V. Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6:1–6, June 2004. URL http://doi.acm.org/10.1145/1007730.1007733. (Cited on pages 76 and 156.)

[73] Chun-hung Cheng, Jian Tang, Ada Wai-chee Fu, and Irwin King. Hierarchical classification of documents with error control. In *Advances in Knowledge Discovery and Data Mining*, volume 2035 of *Lecture Notes in Computer Science*, pages 433–443. Springer Berlin / Heidelberg, 2001. URL http://dx.doi.org/10.1007/3-540-45357-1_46. (Cited on page 324.)

[74] Weiwei Cheng and Eyke Hüllermeier. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76:211–225, 2009. URL http://dx.doi.org/10.1007/s10994-009-5127-5. (Cited on page 148.)

[75] Xiwen Cheng and Feiyu Xu. Fine-grained opinion topic and polarity identification. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA). URL http://www.dfki.de/~feiyu/OMINE_LREC_2008.pdf. (Cited on pages 145 and 146.)

[76] Yejin Choi and Claire Cardie. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1613715.1613816. (Cited on pages 24 and 208.)

[77] Yejin Choi and Claire Cardie. Adapting a polarity lexicon using unteger linear programming for domain-specific sentiment classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, EMNLP '09, pages 590–598, Morristown, NJ, USA, 2009. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1699571.1699590. (Cited on pages 169, 170 and 173.)

[78] Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1220575.1220620. (Cited on pages 13 and 14.)

[79] Yejin Choi, Eric Breck, and Claire Cardie. Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 431–439, Sydney, Australia, July 2006. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W06/W06-1651. (Cited on page 13.)

[80] Clarabridge. Product Website: Clarabridge Sentiment Analysis Component. URL http://www.clarabridge.com/Default.aspx?TabId=331. [Online; accessed 12/2012]. (Cited on page 26.)

[81] Amanda Clare and Ross King. Knowledge discovery in multi-label phenotype data. In *Principles of Data Mining and Knowledge Discovery*, volume 2168 of *Lecture Notes in Computer Science*, pages 42–53. Springer Berlin / Heidelberg, 2001. URL http://dx.doi.org/10.1007/3-540-44794-6_4. (Cited on page 148.)

[82] S. Clematide and M. Klenner. Evaluation and extension of a polarity lexicon for German. In *Proceedings of the First Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, WASSA 2010, pages 7–13, 2010. URL http://gplsi.dlsi.ua.es/congresos/wassa2010/fitxers/WASSA2010_Proceedings_.pdf#page=14. (Cited on page 171.)

[83] G. L. Clore, A. Ortony, and M. A. Foss. The psychological foundations of the affective lexicon. *Journal of Personality and Social Psychology*, 53(4):751, 1987. URL http://psycnet.apa.org/doi/10.1037/0022-3514.53.4.751. (Cited on page 53.)

[84] J. Cohen and Others. A coefficient of agreement for nominal scales. *Educational and Psychological measurement*, 20(1):37–46, 1960. URL http://psycnet.apa.org/doi/10.1177/001316446002000104. (Cited on page 72.)

[85] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 189–196, 1999. URL http://acl.ldc.upenn.edu/W/W99/W99-0613.pdf. (Cited on page 34.)

[86] comScore and The Kelsey Group. Online Consumer-Generated reviews have significant impact on offline purchase behavior. http://www.comscore.com/Press_Events/Press_Releases/2007/11/Online_Consumer_Reviews_Impact_Offline_Purchasing_Behavior, November 2007. (Cited on page 2.)

[87] Linguistic Data Consortium. Automated content extraction (ACE) program. URL http://projects.ldc.upenn.edu/ace/. [Online; accessed 12/2012]. (Cited on pages 49, 56, 59 and 73.)

[88] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995. URL http://dx.doi.org/10.1007/BF00994018. (Cited on page 22.)

[89] Isaac G. Councill, Ryan McDonald, and Leonid Velikovich. What's great and what's not: Learning to classify the scope of negation for improved sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, NeSp-NLP '10, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1858959.1858969. (Cited on pages 216 and 217.)

[90] M. Craven and J. Kumlien. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology*, pages 77–86. Heidelberg, Germany, 1999. URL http://www.aaai.org/Papers/ISMB/1999/ISMB99-010.pdf. (Cited on pages 29 and 31.)

[91] D. A. Cruse. *Lexical Semantics*. Cambridge Textbooks in Linguistics. Cambridge University Press, 1986. (Cited on page 41.)

[92] Hang Cui, Vibhu Mittal, and Mayur Datar. Comparative experiments on sentiment classification for online product reviews. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 1265–1270. AAAI Press, 2006. URL http://www.aaai.org/Papers/AAAI/2006/AAAI06-198.pdf. (Cited on pages 60, 203, 205, 210 and 217.)

[93] Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *Proceedings of the 20th National Conference on Artificial Intelligence*, pages 746–751. AAAI Press, 2005. URL http://portal.acm.org/citation.cfm?id=1619410.1619452. (Cited on pages 35 and 264.)

[94] Fred J. Damerau. Generating and evaluating domain-oriented multi-word terms from texts. *Information Processing and Management*, 29(4):433–447, July 1993. URL http://dx.doi.org/10.1016/0306-4573(93)90039-G. (Cited on page 101.)

[95] S. Das and M. Chen. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific Finance Association Annual Conference*, volume 35, page 43, 2001. URL http://papers.ssrn.com/sol3/papers.cfm?abstract_id=276189. (Cited on pages 207 and 215.)

[96] Kushal Dave, Steve Lawrence, and David M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International Conference on World Wide Web (WWW '03)*, pages 519–528, New York, NY, USA, 2003. ACM. URL http://dx.doi.org/10.1145/775152.775226. (Cited on pages 11, 19, 60, 203, 204, 205, 210, 211, 212, 214, 215 and 217.)

[97] Ernest C. Davenport and Nader A. El-Sanhurry. Phi/Phimax: Review and synthesis. *Educational and Psychological Measurement*, 51(4):821–828, Winter 1991. URL http://epm.sagepub.com/content/51/4/821.abstract. (Cited on page 78.)

[98] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Enhanced sentiment learning using Twitter hashtags and smileys. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10)*, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1944566.1944594. (Cited on page 171.)

[99] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990. URL http://lsa.colorado.edu/papers/JASIS.lsi.90.pdf. (Cited on page 96.)

[100] Ann Devitt and Khurshid Ahmad. Sentiment polarity identification in financial news: A cohesion-based approach. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 984–991, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology-new/P07/P07-1124.pdf. (Cited on pages 2 and 203.)

[101] Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3), 1945. URL http://www.jstor.org/stable/1932409. (Cited on page 110.)

[102] Xiaowen Ding, Bing Liu, and Philip S. Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the International Conference on Web Search and Web Data Mining*, WSDM '08, pages 231–240, New York, NY, USA, 2008. ACM. URL http://dx.doi.org/10.1145/1341531.1341561. (Cited on pages 21, 39, 41, 70, 71, 173, 175, 203 and 208.)

[103] Anhai Doan, Raghu Ramakrishnan, and Alon Y. Halevy. Crowdsourcing systems on the World-Wide Web. *Communications of the ACM*, 54(4):86–96, April 2011. URL http://doi.acm.org/10.1145/1924421.1924442. (Cited on page 34.)

[104] Douglas Douglas. The Multi-Dimensional approach to linguistic analyses of genre variation: An overview of methodology and findings. *Computers and the Humanities*, 26:331–345, 1992. URL http://dx.doi.org/10.1007/BF00136979. (Cited on page 13.)

[105] Weifu Du, Songbo Tan, Xueqi Cheng, and Xiaochun Yun. Adapting information bottleneck method for automatic construction of domain-oriented sentiment lexicon. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM '10)*, New York, NY, USA, 2010. ACM. URL http://doi.acm.org/10.1145/1718487.1718502. (Cited on pages 169, 170 and 173.)

[106] Susan Dumais and Hao Chen. Hierarchical classification of web content. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, New York, NY, USA, 2000. ACM. URL http://doi.acm.org/10.1145/345508.345593. (Cited on page 147.)

[107] Susan Dumais, John Platt, David Heckerman, and Mehran Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the 7th International Conference on Information and Knowledge Management (CIKM '98)*, New York, NY, USA, 1998. ACM. URL http://doi.acm.org/10.1145/288627.288651. (Cited on page 149.)

[108] Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, March 1993. ISSN 0891-2017. URL http://portal.acm.org/citation.cfm?id=972450.972454. (Cited on pages 96, 101 and 110.)

[109] Miriam Eckert, Lyndsie Clark, and Jason Kessler. Structural sentiment and entity annotation guidelines. [Online; accessed 12/2012], 2009. URL https://www.cs.indiana.edu/~jaskessl/annotationguidelines.pdf. (Cited on pages 55 and 56.)

[110] H. P. Edmundson and R. E. Wyllys. Automatic abstracting and indexing — survey and recommendations. *Communications of the ACM*, 4(5):226–234, May 1961. URL http://doi.acm.org/10.1145/366532.366545. (Cited on page 101.)

[111] K. Eguchi and V. Lavrenko. Sentiment retrieval using generative models. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 345–354, Stroudsburg, PA, USA, July 2006. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1610075.1610124. (Cited on pages 14 and 207.)

[112] Paul Ekman. Universals and cultural differences in facial expressions of emotion. In *Nebraska Symposium on Motivation*. University of Nebraska Press, 1971. URL http://books.google.de/books?id=ir4_NAAACAAJ. (Cited on page 13.)

[113] Paul Ekman. An argument for basic emotions. *Cognition & Emotion*, 6(3-4):169–200, 1992. URL http://www.tandfonline.com/doi/abs/10.1080/02699939208411068. (Cited on page 13.)

[114] Paul Ekman. Basic emotions. In *Handbook of Cognition and Emotion*, pages 45–60. John Wiley & Sons, Ltd, 1999. ISBN 9780470013496. URL http://dx.doi.org/10.1002/0470013494.ch3. (Cited on pages 13 and 32.)

[115] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. *Advances in Neural Information Processing Systems*, 14:681–687, 2001. URL http://www-2.cs.cmu.edu/Groups/NIPS/NIPS2001/papers/psgz/AA45.ps.gz. (Cited on page 148.)

[116] Charles Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, IJCAI'01, pages 973–978, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1642194.1642224. (Cited on page 157.)

[117] Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 213–220, New York, NY, USA, 2008. ACM. URL http://dx.doi.org/10.1145/1401890.1401920. (Cited on page 226.)

[118] Seyda Ertekin, Jian Huang, and C. Lee Giles. Active learning for class imbalance problem. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 823–824, New York, NY, USA, 2007. ACM. URL http://doi.acm.org/10.1145/1277741.1277927. (Cited on page 156.)

[119] Andrew Estabrooks, Taeho Jo, and Nathalie Japkowicz. A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 20(1), 2004. URL http://dx.doi.org/10.1111/j.0824-7935.2004.t01-1-00228.x. (Cited on page 157.)

[120] Andrea Esuli and Fabrizio Sebastiani. SENTIWORDNET: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06)*, pages 417–422, 2006. URL http://www.lrec-conf.org/proceedings/lrec2006/pdf/384_pdf. (Cited on pages 172 and 198.)

[121] Andrea Esuli and Fabrizio Sebastiani. PageRanking WordNet synsets: An application to opinion mining. In *Proceedings of ACL-07, the 45th Annual Meeting of the Association of Computational Linguistics*, pages 424–431. Association for Computational Linguistics, 2007. URL http://acl.ldc.upenn.edu/P/P07/P07-1054.pdf. (Cited on page 176.)

[122] Andrea Esuli, Tiziano Fagni, and Fabrizio Sebastiani. Boosting multi-label hierarchical text categorization. *Information Retrieval*, 11:287–313, 2008. URL http://dx.doi.org/10.1007/s10791-008-9047-y. (Cited on pages 137, 147 and 324.)

[123] O. Etzioni, M. Cafarella, D. Downey, A. M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134, 2005. URL http://dx.doi.org/10.1016/j.artint.2005.03.001. (Cited on page 95.)

[124] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74, December 2008. URL http://doi.acm.org/10.1145/1409360.1409378. (Cited on page 30.)

[125] A. Fahrni and M. Klenner. Old wine or warm beer: Target-specific sentiment analysis of adjectives. In *Symposion on Affective Language in Human and Machine, AISB Convention*, pages 60–63, 2008. URL http://www.aisb.org.uk/convention/aisb08/proc/proceedings/02AffectiveLanguage/11.pdf. (Cited on pages 49, 169, 173 and 175.)

[126] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9, June 2008. URL http://dl.acm.org/citation.cfm?id=1390681.1442794. (Cited on page 228.)

[127] Olga Feiguina and Guy Lapalme. Query-based summarization of customer reviews. In *Advances in Artificial Intelligence*, volume 4509 of *Lecture Notes in Computer Science*, pages 452–463. Springer Berlin / Heidelberg, 2007. URL http://dx.doi.org/10.1007/978-3-540-72665-4_39. (Cited on pages 72, 97 and 99.)

[128] Liliana Ferreira, Niklas Jakob, and Iryna Gurevych. A comparative study of feature extraction algorithms in customer reviews. In *Proceedings of the 2008 IEEE International Conference on Semantic Computing*, pages 144–151, Washington, DC, USA, 2008. IEEE Computer Society. URL http://dx.doi.org/10.1109/ICSC.2008.40. (Cited on pages 305, 311 and 312.)

[129] C. R. Fink, D. S. Chou, J. J. Kopecky, and A. J. Llorens. Coarse- and fine-grained sentiment analysis of social media text. *Johns Hopkins APL Technical Digest*, 30(1), 2011. URL http://www.jhuapl.edu/techdigest/TD/td3001/Fink.pdf. (Cited on page 207.)

[130] J. R. Firth. A synopsis of linguistic theory 1930-55. *Studies in Linguistic Analysis*, pages 1–32, 1957. (Cited on page 172.)

[131] J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5): 378, 1971. URL http://psycnet.apa.org/doi/10.1037/h0031619. (Cited on page 72.)

[132] Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. Automatic recognition of multi-word terms: The C-value/NC-value method. *International Journal on Digital Libraries*, 3:115–130, 2000. URL http://dx.doi.org/10.1007/s007999900023. (Cited on page 101.)

[133] Katerina T. Frantzi and Sophia Ananiadou. Extracting nested collocations. In *Proceedings of the 16th Conference on Computational Linguistics*, COLING '96, pages 41–46, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/992628.992639. (Cited on page 111.)

[134] G. M. Fung and O. L. Mangasarian. Multicategory proximal support vector machine classifiers. *Machine Learning*, 59(1):77–97, 2005. URL http://dx.doi.org/10.1007/s10994-005-0463-6. (Cited on page 157.)

[135] Michael Gamon. Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 841–847, Geneva, Switzerland, 2004. COLING. URL http://acl.ldc.upenn.edu/coling2004/MAIN/pdf/121-637.pdf. (Cited on pages 203, 205, 213 and 217.)

[136] Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric Ringger. Pulse: Mining customer opinions from free text. In *Advances in Intelligent Data Analysis VI*, volume 3646 of *Lecture Notes in Computer Science*, chapter 12, pages 121–132. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2005. URL http://dx.doi.org/10.1007/11552253_12. (Cited on pages 24, 46, 60, 146, 207 and 208.)

[137] Murthy Ganapathibhotla and Bing Liu. Mining opinions in comparative sentences. In *Proceedings of the 22nd International Conference on Computational Linguistics*, COLING '08, pages 241–248, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1599081.1599112. (Cited on page 23.)

[138] G. Ganu, N. Elhadad, and A. Marian. Beyond the stars: Improving rating predictions using review text content. In *Proceedings of the 12th International Workshop on the Web and Databases*, 2009. URL http://webdb09.cse.buffalo.edu/papers/Paper9/WebDB.pdf. (Cited on pages 22, 64, 70, 72, 157, 208 and 314.)

[139] David Garcia and Frank Schweitzer. Emotions in product reviews – empirics and models. In *Proceedings of the 3rd International Conference on Social Computing*, SocialCom '11, pages 483–488. IEEE, October 2011. URL http://dx.doi.org/10.1109/PASSAT/SocialCom.2011.219. (Cited on page 171.)

[140] D. Gayo-Avello. I wanted to predict elections with twitter and all i got was this lousy paper – a balanced survey on election prediction using twitter data, 2012. URL http://arxiv.org/abs/1204.6441. (Cited on page 2.)

[141] Anindya Ghose and Panagiotis G. Ipeirotis. Designing novel review ranking systems: Predicting the usefulness and impact of reviews. In *Proceedings of the 9th International Conference on Electronic Commerce*, ICEC '07, pages 303–310, New York, NY, USA, 2007. ACM. URL http://doi.acm.org/10.1145/1282100.1282158. (Cited on page 20.)

[142] Giorgio Giacinto, Roberto Perdisci, Mauro D. Rio, and Fabio Roli. Intrusion detection in computer networks by a modular ensemble of one-class classifiers. *Information Fusion*, 9(1):69–82, 2008. URL http://www.sciencedirect.com/science/article/pii/S1566253506000765. (Cited on page 226.)

[143] J. Giles. Internet encyclopaedias go head to head. *Nature*, 438(7070):900–901, 2005. ISSN 0028-0836. URL http://dx.doi.org/10.1038/438900a. (Cited on page 30.)

[144] S. Gindl, A. Weichselbraun, and A. Scharl. Cross-domain contextualisation of sentiment lexicons. In *Proceedings of the 19th European Conference on Artificial Intelligence*, ECAI '10, pages 771–776. IOS Press, August 2010. URL http://dx.doi.org/10.3233/978-1-60750-606-5-771. (Cited on page 173.)

[145] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. Technical report, Stanford University, 2009. URL http://www-cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf. (Cited on pages 26, 30, 32, 213 and 217.)

[146] N. Godbole, M. Srinivasaiah, and S. Skiena. Large-Scale sentiment analysis for news and blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, pages 219–222, 2007. URL http://icwsm.org/papers/3--Godbole-Srinivasaiah-Skiena.pdf. (Cited on pages 24, 171, 172, 174, 175, 176 and 203.)

[147] Shantanu Godbole and Sunita Sarawagi. Discriminative methods for multi-labeled classification. In *Advances in Knowledge Discovery and Data Mining*, volume 3056 of *Lecture Notes in Computer Science*, pages 22–30. Springer Berlin / Heidelberg, 2004. URL http://dx.doi.org/10.1007/978-3-540-24775-3_5. (Cited on page 148.)

[148] Andrew B. Goldberg and Xiaojin Zhu. Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In *Proceedings of the 1st Workshop on Graph Based Methods for Natural Language Processing*, pages 45–52, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1654758.1654769. (Cited on pages 19 and 207.)

[149] Ulrike Gretzel and Kyung H. Yoo. Use and impact of online travel reviews. In *Information and Communication Technologies in Tourism*, pages 35–46. Springer Vienna, 2008. URL http://dx.doi.org/10.1007/978-3-211-77280-5_4. (Cited on page 2.)

[150] Griffiths and Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Proceedings of the 16th Conference on Neural Information Processing Systems (NIPS)*, volume 16, pages 17–24. The MIT Press, 2003. URL http://books.nips.cc/papers/files/nips16/NIPS2003_AA03.pdf. (Cited on page 314.)

[151] Ralph Grishman and Beth Sundheim. Message understanding conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, pages

466–471, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/992628.992709. (Cited on page 59.)

[152] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, June 1993. URL http://dx.doi.org/10.1006/knac.1993.1008. (Cited on page 42.)

[153] Honglei Guo, Huijia Zhu, Zhili Guo, XiaoXun Zhang, and Zhong Su. Product feature categorization with multilevel latent semantic association. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 1087–1096, New York, NY, USA, 2009. ACM. URL http://doi.acm.org/10.1145/1645953.1646091. (Cited on page 97.)

[154] Narendra Gupta, Giuseppe Di Fabbrizio, and Patrick Haffner. Capturing the stars: Predicting ratings for service and product reviews. In *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search*, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1867767.1867772. (Cited on pages 19 and 207.)

[155] Narendra Gupta, Mazin Gilbert, and Giuseppe D. Fabbrizio. Emotion detection in email customer care. *Computational Intelligence*, 2012. URL http://dx.doi.org/10.1111/j.1467-8640.2012.00454.x. (Cited on page 13.)

[156] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997. URL http://books.google.de/books?id=Ofw5w1yuD8kC. (Cited on page 136.)

[157] Yaw Gyamfi, Janyce Wiebe, Rada Mihalcea, and Cem Akkaya. Integrating knowledge for subjectivity sense labeling. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1620754.1620757. (Cited on page 170.)

[158] M. A. K. Halliday and C. M. Matthiessen. *An Introduction to Functional Grammar (3rd edition)*. Routledge, 2004. ISBN 978-0340761670. (Cited on page 54.)

[159] R. W. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 29(2), 1950. URL http://lucent.com/bstj/vol29-1950/articles/bstj29-2-147.pdf. (Cited on page 323.)

[160] Vasileios Hatzivassiloglou and Kathleen R. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, ACL '98, pages 174–181, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/979617.979640. (Cited on pages 171, 173, 174 and 175.)

[161] Ben He, Craig Macdonald, Jiyin He, and Iadh Ounis. An effective statistical approach to blog post opinion retrieval. In *Proceeding of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 1063–1072, New York, NY, USA, 2008. ACM. URL http://dx.doi.org/10.1145/1458082.1458223. (Cited on pages 2 and 13.)

[162] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21:1263–1284, 2009. URL http://dx.doi.org/10.1109/TKDE.2008.239. (Cited on pages 76, 156 and 157.)

[163] M. Hewings. *Advanced Grammar in Use*. Cambridge University Press, 2005. ISBN 978-0521614030. (Cited on page 51.)

[164] Ryuichiro Higashinaka, Rashmi Prasad, and Marilyn A. Walker. Learning to generate naturalistic utterances using reviews in spoken dialogue systems. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 265–272, Sydney, Australia, July 2006. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P06-1034.pdf. (Cited on page 31.)

[165] Linh Hoang, Jung-Tae Lee, Young-In Song, and Hae-Chang Rim. Combining local and global resources for constructing an error-minimized opinion word dictionary. In *PRICAI 2008: Trends in Artificial Intelligence*, volume 5351 of *Lecture Notes in Computer Science*, chapter 63, pages 688–697. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. URL http://dx.doi.org/10.1007/978-3-540-89197-0_63. (Cited on page 174.)

[166] Thomas Hoffman. Online reputation management is hot — but is it ethical. *Computerworld*, February 2008. URL http://www.computerworld.com/s/article/9060960/. (Cited on page 2.)

[167] Raphael Hoffmann, Congle Zhang, and Daniel S. Weld. Learning 5000 relational extractors. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 286–295, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P10-1030.pdf. (Cited on page 31.)

[168] Wolfgang Holzinger, Bernhard Krüpl, and Marcus Herzog. Using ontologies for extracting product features from web pages. In *The Semantic Web - ISWC 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 286–299. Springer Berlin / Heidelberg, 2006. URL http://dx.doi.org/10.1007/11926078_21. (Cited on page 97.)

[169] X. Hong, S. Chen, and C. J. Harris. A kernel-based two-class classifier for imbalanced data sets. *IEEE Transactions on Neural Networks*, 18(1):28–41, 2007. URL http://dx.doi.org/10.1109/TNN.2006.882812. (Cited on page 157.)

[170] John Horrigan. Online shopping. *Pew Internet & American Life Project Report*, February 2008. URL http://www.pewinternet.org/Reports/2008/Online-Shopping.aspx. (Cited on page 2.)

[171] J. Howe. *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business*. Crown Publishing Group, 2008. ISBN 9780307396204. (Cited on page 34.)

[172] Jeff Howe. Crowdsourcing: A definition, 2006. URL http://www.crowdsourcing.com/cs/2006/06/crowdsourcing_a.html. [Online; accessed 12/2012]. (Cited on page 29.)

[173] Jeff Howe. The rise of crowdsourcing. *Wired Magazine*, 14(6):1–4, 2006. URL http://www.wired.com/wired/archive/14.06/crowds.html. (Cited on page 29.)

[174] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *Transactions on Neural Networks*, 13(2), March 2002. URL http://dx.doi.org/10.1109/72.991427. (Cited on page 223.)

[175] Pei-Yun Hsueh, Prem Melville, and Vikas Sindhwani. Data quality from crowdsourcing: A study of annotation selection criteria. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 27–35, Boulder, Colorado, June 2009. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W09-1904. (Cited on pages 29 and 34.)

[176] M. Hu and B. Liu. Mining opinion features in customer reviews. In *Proceedings of the National Conference on Artificial Intelligence*, pages 755–760, Menlo Park, California, July 2004. Association for the Advancement of Artificial Intelligence, AAAI Press. URL http://www.aaai.org/Papers/AAAI/2004/AAAI04-119.pdf. (Cited on pages 94 and 99.)

[177] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA, 2004. ACM. URL http://dx.doi.org/10.1145/1014052.1014073. (Cited on pages 21, 22, 40, 55, 70, 71, 94, 95, 97, 98, 99, 114, 145, 146, 171, 172, 173, 175, 176, 262 and 311.)

[178] Robert A. Hummel and Steven W. Zucker. On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(3):267–287, May 1983. URL http://dx.doi.org/10.1109/TPAMI.1983.4767390. (Cited on page 24.)

[179] IBM Corporation. Product Website: IBM Cognos Consumer Insight. URL www.ibm.com/software/products/us/en/cognos-consumer-insight. [Online; accessed 12/2012]. (Cited on page 26.)

[180] Nancy Ide. Making senses: Bootstrapping sense-tagged lists of semantically-related words. In *Computational Linguistics and Intelligent Text Processing*, volume 3878 of *Lecture Notes in Computer Science*, pages 13–27. Springer Berlin / Heidelberg, 2006. URL http://dx.doi.org/10.1007/11671299_2. (Cited on page 172.)

[181] Daisuke Ikeda, Hiroya Takamura, Lev arie Ratinov, and Manabu Okumura. Learning to shift the polarity of words for sentiment classification. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP*, 2008. URL http://www.aclweb.org/anthology/I/I08/I08-1039.pdf. (Cited on pages 207, 215 and 217.)

[182] Niklas Jakob. *Extracting Opinion Targets from User-Generated Discourse with an Application to Recommendation Systems*. PhD thesis, Technische Universtität Darmstadt, May 2011. URL http://tuprints.ulb.tu-darmstadt.de/2609/. (Cited on pages 14, 22, 114, 115 and 308.)

[183] Niklas Jakob and Iryna Gurevych. Extracting opinion targets in a single- and cross-domain setting with conditional random fields. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1035–1045, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1870658.1870759. (Cited on pages 22, 169, 208 and 262.)

[184] Niklas Jakob and Iryna Gurevych. Using anaphora resolution to improve opinion target identification in movie reviews. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1858842.1858891. (Cited on page 14.)

[185] Niklas Jakob, Mark-Christoph Müller, and Iryna Gurevych. LRTwiki: Enriching the likelihood ratio test with encyclopedic information for the extraction of relevant terms. In *WIKIAI 09 - IJCAI Workshop: User Contributed Knowledge and Artificial Intelligence: An Evolving Synergy*, pages 3–8, 2009. URL http://lit.csci.unt.edu/~wikiai09/papers/jakob.pdf. (Cited on pages 112, 130 and 136.)

[186] Niklas Jakob, Stefan H. Weber, Mark C. Müller, and Iryna Gurevych. Beyond the stars: Exploiting free-text user reviews to improve the accuracy of movie recommendations. In *Proceeding of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion*, TSA '09, pages 57–64, New York, NY, USA, 2009. ACM. URL http://doi.acm.org/10.1145/1651461.1651473. (Cited on page 2.)

[187] Jim Jansen. Online product research, September 2010. URL http://www.pewinternet.org/Reports/2010/Online-Product-Research.aspx. [Online; accessed 12/2012]. (Cited on page 2.)

[188] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6:429–449, October 2002. URL http://dl.acm.org/citation.cfm?id=1293951.1293954. (Cited on pages 76 and 156.)

[189] Valentin Jijkoun, Maarten de Rijke, and Wouter Weerkamp. Generating focused topic-specific sentiment lexicons. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1858681.1858741. (Cited on pages 169 and 173.)

[190] Wei Jin, Hung H. Ho, and Rohini K. Srihari. OpinionMiner: A novel machine learning system for web opinion mining and extraction. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 1195–1204, New York, NY, USA, 2009. ACM. URL http://doi.acm.org/10.1145/1557019.1557148. (Cited on page 22.)

[191] Nitin Jindal and Bing Liu. Identifying comparative sentences in text documents. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 244–251, New York, NY, USA, 2006. ACM. URL http://dx.doi.org/10.1145/1148170.1148215. (Cited on page 23.)

[192] Nitin Jindal and Bing Liu. Review spam detection. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 1189–1190, New York, NY, USA, 2007. ACM. URL http://doi.acm.org/10.1145/1242572.1242759. (Cited on page 20.)

[193] Nitin Jindal and Bing Liu. Opinion spam and analysis. In *Proceedings of the International Conference on Web Search and Web Data Mining*, WSDM '08, pages 219–230, New York, NY, USA, 2008. ACM. URL http://doi.acm.org/10.1145/1341531.1341560. (Cited on page 20.)

[194] Yohan Jo and Alice H. Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 815–824, New York, NY, USA, 2011. ACM. URL http://doi.acm.org/10.1145/1935826.1935932. (Cited on pages 23, 46, 56, 57, 315 and 316.)

[195] W. Johnson. Studies in language behavior. *Psychological Monographs: General and Applied*, 56(2): 1–15, 1944. URL http://dx.doi.org/10.1037/h0093508. (Cited on page 82.)

[196] Rosie Jones, Andrew McCallum, Kamal Nigam, and Ellen Riloff. Bootstrapping for text learning tasks. In *IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications*, pages 52–63, 1999. URL http://www.kamalnigam.com/papers/bootstrap-ijcaiws99.pdf. (Cited on pages 34 and 264.)

[197] Mahesh Joshi and Carolyn Penstein-Rosé. Generalizing dependency features for opinion mining. In *Proceedings of the ACL-IJCNLP 2009 Conference*, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P09/P09-2079.pdf. (Cited on pages 214 and 217.)

[198] Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2009. ISBN 978-0-13-504196-3. (Cited on pages 20, 43 and 44.)

[199] K. Kageura and B. Umino. Methods of automatic term recognition: A review. *Terminology*, 3(2): 259–289, 1996. URL http://dx.doi.org/10.1075/term.3.2.03kag. (Cited on pages 100 and 101.)

[200] Nobuhiro Kaji and Masaru Kitsuregawa. Automatic construction of polarity-tagged corpus from HTML documents. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 452–459, Morristown, NJ, USA, 2006. Association for Computational Linguistics. URL http://acl.ldc.upenn.edu/P/P06/P06-2059.pdf. (Cited on page 173.)

[201] Nobuhiro Kaji and Masaru Kitsuregawa. Building lexicon for sentiment analysis from massive collection of HTML documents. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL, pages 1075–1083, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D/D07/D07-1115.pdf. (Cited on pages 24, 173 and 175.)

[202] J. Kamps, M. Marx, R. J. Mokken, and M. De Rijke. Using WordNet to measure semantic orientation of adjectives. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, volume 4, pages 1115–1118. European Language Resources Association (ELRA), May 2004. URL http://dare.uva.nl/document/154122. (Cited on pages 172 and 176.)

[203] Hiroshi Kanayama and Tetsuya Nasukawa. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 355–363, Sydney, Australia, July 2006. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W06/W06-1642.pdf. (Cited on page 173.)

[204] Walter Kasper and Mihaela Vela. Sentiment analysis for hotel reviews. In *Proceedings of the Computational Linguistics-Applications Conference*, October 2011. URL http://www.dfki.de/lt/publication_show.php?id=5601. (Cited on page 203.)

[205] Alistair Kennedy and Diana Inkpen. Sentiment classification of movie reviews using contextual valence shifter. *Computational Intelligence*, 22(2):110–125, 2006. URL http://dx.doi.org/10.1111/j.1467-8640.2006.00277.x. (Cited on pages 203, 215 and 217.)

[206] Brett Kessler, Geoffrey Numberg, and Hinrich Schütze. Automatic detection of text genre. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, ACL '98, pages 32–38, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/976909.979622. (Cited on page 13.)

[207] J. S. Kessler and N. Nicolov. Targeting sentiment expressions through supervised ranking of linguistic configurations. In *Proceedings of the 3rd International AAAI Conference on Weblogs and Social Media*, pages 90–97, 2009. URL https://www.aaai.org/ocs/index.php/ICWSM/09/paper/viewFile/190/413. (Cited on pages 14, 41 and 169.)

[208] Jason S. Kessler. Polling the blogosphere: A rule-based approach to belief classification. In *International Conference on Weblogs and Social Media*, ICWSM, 2008. URL http://www.aaai.org/Papers/ICWSM/2008/ICWSM08-016.pdf. (Cited on pages 216 and 217.)

[209] Jason S. Kessler, Miriam Eckert, Lyndsay Clark, and Nicolas Nicolov. The 2010 ICWSM JDPA sentiment corpus for the automotive domain. In *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media Data Workshop Challenge (ICWSM-DWC 2010)*, 2010. URL http://www.cs.indiana.edu/~jaskessl/icwsm10.pdf. (Cited on pages 39, 50, 53, 56, 70 and 73.)

[210] Soo M. Kim and Eduard Hovy. Automatic identification of pro and con reasons in online reviews. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 483–490, Morristown, NJ, USA, 2006. Association for Computational Linguistics. URL http://acl.ldc.upenn.edu/P/P06/P06-2063.pdf. (Cited on pages 24, 32, 207 and 223.)

[211] Soo M. Kim and Eduard Hovy. Identifying and analyzing judgment opinions. In *Proceedings of the Conference on Human Language Technology of the North American Chapter of the Association of Computational Linguistics*, pages 200–207, Morristown, NJ, USA, 2006. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1220835.1220861. (Cited on pages 13, 14, 172 and 173.)

[212] Soo-Min Kim and Eduard Hovy. Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, SST '06, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1654641.1654642. (Cited on page 13.)

[213] Soo-Min Kim, Patrick Pantel, Tim Chklovski, and Marco Pennacchiotti. Automatically assessing review helpfulness. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 423–430, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1610075.1610135. (Cited on page 20.)

[214] Manfred Klenner, Angela Fahrni, and Stefanos Petrakis. PolArt: A robust tool for sentiment analysis. In *Proceedings of the 17th Nordic Conference of Computational Linguistics*, volume 4, pages 235–238. Northern European Association for Language Technology (NEALT), 2009. URL http://www.ifi.uzh.ch/pax/uploads/pdf/publication/1117/nodalida2009.pdf. (Cited on pages 49, 169, 170 and 203.)

[215] Manfred Klenner, Stefanos Petrakis, and Angela Fahrni. Robust compositional polarity classification. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, RANLP '09, Borovets, Bulgaria, September 2009. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/R09-1034.pdf. (Cited on page 203.)

[216] Nozomi Kobayashi, Kentaro Inui, and Yuji Matsumoto. Extracting aspect-evaluation and aspect-of relations in opinion mining. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '07, 2007. URL http://www.aclweb.org/anthology/D/D07/D07-1114.pdf. (Cited on page 14.)

[217] Daphne Koller and Mehran Sahami. Hierarchically classifying documents using very few words. In *Proceedings of the 14th International Conference on Machine Learning*, ICML '97, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=645526.657130. (Cited on page 147.)

[218] Moshe Koppel and Jonathan Schler. Authorship verification as a one-class classification problem. In *Proceedings of the 21st International Conference on Machine Learning*, ICML '04, New York, NY, USA, 2004. ACM. URL http://doi.acm.org/10.1145/1015330.1015448. (Cited on page 226.)

[219] Moshe Koppel and Jonathan Schler. The importance of neutral examples for learning sentiment. *Computational Intelligence*, 22(2), 2006. URL http://dx.doi.org/10.1111/j.1467-8640.2006.00276.x. (Cited on pages 206 and 207.)

[220] L. Kozakov, Y. Park, T. Fin, Y. Drissi, Y. Doganata, and T. Cofino. Glossary extraction and utilization in the information search and delivery system for IBM technical support. *IBM Systems Journal*, 43(3):546–563, 2004. URL http://dx.doi.org/10.1147/sj.433.0546. (Cited on pages 100, 104 and 105.)

[221] Taku Kudo and Yuji Matsumoto. A boosting algorithm for classification of semi-structured text. In *Proceedings of EMNLP 2004*, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL http://acl.ldc.upenn.edu/acl2004/emnlp/pdf/Kudo2.pdf. (Cited on pages 214 and 217.)

[222] Karen Kukich. Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24(4):377–439, 1992. URL http://doi.acm.org/10.1145/146370.146380. (Cited on page 107.)

[223] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. URL http://portal.acm.org/citation.cfm?id=645530.655813. (Cited on pages 22 and 97.)

[224] E. L. M. Law, L. Von Ahn, R. B. Dannenberg, and M. Crawford. Tagatune: A game for music and sound annotation. In *International Conference on Music Information Retrieval (ISMIR'07)*, pages 361–364, 2007. URL http://ismir2007.ismir.net/proceedings/ISMIR2007_p361_law.pdf. (Cited on page 34.)

[225] Neil D. Lawrence and Bernhard Schölkopf. Estimating a kernel fisher discriminant in the presence of label noise. In *Proceedings of the 18th International Conference on Machine Learning*, ICML '01, pages 306–313, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=645530.655665. (Cited on page 264.)

[226] C. W. K. Leung, S. C. F. Chan, and F. Chung. Integrating collaborative filtering and sentiment analysis: A rating inference approach. In *Proceedings of The ECAI 2006 Workshop on Recommender Systems*, pages 62–66. Citeseer, 2006. URL http://www.mysmu.edu/staff/caneleung/pub/rsw06_ratingInfer.pdf. (Cited on page 2.)

[227] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5, 2004. URL http://www.jmlr.org/papers/volume5/lewis04a/lewis04a.pdf. (Cited on page 148.)

[228] David D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '92, New York, NY, USA, 1992. ACM. URL http://dx.doi.org/10.1145/133160.133172. (Cited on page 149.)

[229] Lexalytics Inc. Technical Info: Lexalytics Sentiment Analysis Component. URL http://www.lexalytics.com/technical-info/sentiment-analysis-measuring-emotional-tone. [Online; accessed 12/2012]. (Cited on page 26.)

[230] Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Ying-Ju Xia, Shu Zhang, and Hao Yu. Structure-aware review mining and summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1873781.1873855. (Cited on page 22.)

[231] Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 22(140), 1932. URL http://psycnet.apa.org/psycinfo/1933-01885-001. (Cited on page 171.)

[232] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady W. Lauw. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 939–948, New York, NY, USA, 2010. ACM. URL http://doi.acm.org/10.1145/1871437.1871557. (Cited on pages 20 and 31.)

[233] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *Proceeding of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 375–384, New York, NY, USA, 2009. ACM. URL http://doi.acm.org/10.1145/1645953.1646003. (Cited on pages 23, 56, 57, 315 and 316.)

[234] Lithium Technologies Inc. Product Website: Lithium Social Media Monitoring. URL http://www.lithium.com/products/analytics/listening. [Online; accessed 12/2012]. (Cited on page 26.)

[235] Bing Liu, Yang Dai, Xiaoli Li, Wee S. Lee, and Philip S. Yu. Building text classifiers using positive and unlabeled examples. *Proceedings of the 3rd IEEE International Conference on Data Mining*, 0:179–186, 2003. URL http://doi.ieeecomputersociety.org/10.1109/ICDM.2003.1250918. (Cited on pages 226 and 264.)

[236] Bing Liu, Minqing Hu, and Junsheng Cheng. Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pages 342–351, New York, NY, USA, 2005. ACM. URL http://dx.doi.org/10.1145/1060745.1060797. (Cited on pages 97, 98 and 99.)

[237] Feifan Liu, Dong Wang, Bin Li, and Yang Liu. Improving blog polarity classification via topic analysis and adaptive methods. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Los Angeles, California, June 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N/N10/N10-1042.pdf. (Cited on pages 206 and 217.)

[238] Jingjing Liu and Stephanie Seneff. Review sentiment scoring via a parse-and-paraphrase paradigm. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 161–169, Singapore, August 2009. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D/D09/D09-1017.pdf. (Cited on page 51.)

[239] X. Y. Liu, J. Wu, and Z. H. Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(2), 2009. URL http://dx.doi.org/10.1109/TSMCB.2008.2007853. (Cited on page 157.)

[240] Christoph Lofi, Joachim Selke, and Wolf-Tilo Balke. Information extraction meets crowdsourcing: A promising couple. *Datenbank-Spektrum*, 12:109–120, 2012. URL http://dx.doi.org/10.1007/s13222-012-0092-8. 10.1007/s13222-012-0092-8. (Cited on page 34.)

[241] Yue Lu, Malu Castellanos, Umeshwar Dayal, and ChengXiang Zhai. Automatic construction of a context-aware sentiment lexicon: An optimization approach. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, New York, NY, USA, 2011. ACM. URL http://doi.acm.org/10.1145/1963405.1963456. (Cited on pages 24, 174, 175 and 203.)

[242] Craig Macdonald, Iadh Ounis, and Ian Soboroff. Overview of the TREC 2007 blog track. In *Proceedings of the 16th Text Retrieval Conference*, 2007. URL http://trec.nist.gov/pubs/trec16/papers/BLOG.OVERVIEW16.pdf. (Cited on pages 2, 13 and 207.)

[243] Craig Macdonald, Iadh Ounis, and Ian Soboroff. Overview of the TREC 2008 blog track. In *Proceedings of the 17th Text Retrieval Conference*. NIST, 2008. URL http://trec.nist.gov/pubs/trec17/papers/BLOG.OVERVIEW08.pdf. (Cited on pages 59, 70 and 207.)

[244] A. Maedche. *Ontology Learning for the Semantic Web*. The Springer International Series in Engineering and Computer Science Series. Springer-Verlag GmbH, 2002. URL http://books.google.de/books?id=Hm4jFCxk5VYC. (Cited on page 23.)

[245] Isa Maks and Piek Vossen. Building a fine-grained subjectivity lexicon from a web corpus. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/1018_Paper.pdf. (Cited on pages 170 and 172.)

[246] M. A. Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. In *ICML-2003 Workshop on Learning from Imbalanced Data Sets II*, 2003. URL http://www.site.uottawa.ca/~nat/Workshop2003/maloof-icml03-wids.pdf. (Cited on page 157.)

[247] R. Malouf and T. Mullen. Taking sides: user classification for informal online political discourse. *Internet Research*, 18(2):177–190, 2008. URL http://dx.doi.org/10.1108/10662240810862239. (Cited on page 2.)

[248] Larry M. Manevitz and Malik Yousef. One-class SVMs for document classification. *J. Machine Learning Research*, 2:139–154, March 2002. URL http://www.jmlr.org/papers/volume2/manevitz01a/manevitz01a.pdf. (Cited on pages 98 and 226.)

[249] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts, 1999. ISBN 9780262133609. URL http://nlp.stanford.edu/fsnlp/. (Cited on pages 19, 95, 101 and 210.)

[250] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. URL http://nlp.stanford.edu/IR-book/. (Cited on pages 22, 115, 153, 214, 233 and 313.)

[251] Mitchell P. Marcus, Mary A. Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL http://dl.acm.org/citation.cfm?id=972470.972475. (Cited on pages 102 and 149.)

[252] Shotaro Matsumoto, Hiroya Takamura, and Manabu Okumura. Sentiment classification using word sub-sequences and dependency sub-trees. In *Advances in Knowledge Discovery and Data Mining*, volume 3518 of *Lecture Notes in Computer Science*, chapter 37, pages 21–32. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2005. URL http://dx.doi.org/10.1007/11430919_37. (Cited on pages 203, 212 and 217.)

[253] A. McCallum, R. Rosenfeld, T. Mitchell, and A. Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of the 15th International Conference on Machine Learning*, 1998. URL http://www.robotics.stanford.edu/~ang/papers/icml98-hier.pdf. (Cited on page 147.)

[254] Andrew K. McCallum. Multi-label text classification with a mixture model trained by EM. In *AAAI '99 Workshop on Text Learning*, 1999. URL http://people.cs.umass.edu/~mccallum/papers/multilabel-nips99s.ps. (Cited on page 148.)

[255] Andrew K. McCallum. MALLET: A machine learning for language toolkit. [Online; accessed 12/2012], 2002. URL http://mallet.cs.umass.edu/. (Cited on page 316.)

[256] Peter McCullagh. Regression models for ordinal data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 42(2), 1980. URL http://www.jstor.org/stable/2984952. (Cited on page 207.)

[257] Ryan McDonald, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeff Reynar. Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P07/P07-1055.pdf. (Cited on page 208.)

[258] Arun Meena and T. Prabhakar. Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis. In *Advances in Information Retrieval*, volume 4425 of *Lecture Notes in Computer Science*, chapter 53, pages 573–580. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2007. URL http://dx.doi.org/10.1007/978-3-540-71496-5_53. (Cited on pages 203, 207 and 208.)

[259] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic sentiment mixture: Modeling facets and opinions in weblogs. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 171–180, New York, NY, USA, 2007. ACM. URL http://dx.doi.org/10.1145/1242572.1242596. (Cited on pages 14, 23, 56, 57, 315 and 316.)

[260] Y. Mejova and P. Srinivasan. Crossing media streams with sentiment: Domain adaptation in blogs, reviews and Twitter. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media*, ICWSM '12, 2012. URL http://www.aaai.org/ocs/index.php/ICWSM/ICWSM12/paper/viewFile/4580/4988. (Cited on pages 206 and 264.)

[261] Rada Mihalcea. Using Wikipedia for automatic word sense disambiguation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 196–203, Rochester, New York, April 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N/N07/N07-1025.pdf. (Cited on page 32.)

[262] Rada Mihalcea, Carmen Banea, and Janyce Wiebe. Learning multilingual subjective language via cross-lingual projections. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P07/P07-1123.pdf. (Cited on page 174.)

[263] George A. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38 (11):39–41, November 1995. ISSN 0001-0782. URL http://dx.doi.org/10.1145/219717.219748. (Cited on pages 42, 107, 145, 171 and 212.)

[264] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore, August 2009. Association for Computational Linguistics.

URL http://www.aclweb.org/anthology/P/P09/P09-1113.pdf. (Cited on pages 29, 30 and 31.)

[265] Samaneh Moghaddam, Mohsen Jamali, and Martin Ester. ETF: Extended tensor factorization model for personalizing prediction of review helpfulness. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 163–172, New York, NY, USA, 2012. ACM. URL http://doi.acm.org/10.1145/2124295.2124316. (Cited on pages 20 and 31.)

[266] Alexander A. Morgan, Lynette Hirschman, Marc Colosimo, Alexander S. Yeh, and Jeff B. Colombe. Gene name identification and normalization using a model organism database. *Journal of Biomedical Informatics*, 37(6):396–410, 2004. URL http://dx.doi.org/10.1016/j.jbi.2004.08.010. (Cited on page 31.)

[267] Alessandro Moschitti and Roberto Basili. Complex linguistic features for text classification: A comprehensive study. In *Advances in Information Retrieval*, volume 2997 of *Lecture Notes in Computer Science*, pages 181–196. Springer Berlin / Heidelberg, 2004. URL http://dx.doi.org/10.1007/978-3-540-24752-4_14. (Cited on page 149.)

[268] Fabrice Muhlenbach, Stéphane Lallich, and Djamel A. Zighed. Identifying and handling mislabelled instances. *Journal of Intelligent Information Systems*, 22:89–109, 2004. URL http://dx.doi.org/10.1023/A:1025832930864. 10.1023/A:1025832930864. (Cited on page 264.)

[269] Tony Mullen and Nigel Collier. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of EMNLP 2004*, pages 412–418, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL http://acl.ldc.upenn.edu/acl2004/emnlp/pdf/Mullen.pdf. (Cited on pages 203 and 217.)

[270] R. Munro, S. Bethard, V. Kuperman, V. T. Lai, R. Melnick, C. Potts, T. Schnoebelen, and H. Tily. Crowdsourcing and language studies: The new generation of linguistic data. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*, 2010. URL http://www.aclweb.org/anthology/W10-0719.pdf. (Cited on page 34.)

[271] M. L. Murphy. *Semantic Relations and the Lexicon: Antonymy, Synonymy and other Paradigms*. Cambridge University Press, 2003. URL http://books.google.de/books?id=7pAIpz87jbEC. (Cited on page 41.)

[272] J. C. Na, H. Sui, C. Khoo, S. Chan, and Y. Zhou. Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews. *Advances in Knowledge Organization*, 9, 2004. URL http://www.ntu.edu.sg/home/assgkhoo/papers/na_sui_khoo.sentiment_classification.ISKO2004.pdf. (Cited on pages 213 and 217.)

[273] Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. Dependency tree-based sentiment classification using CRFs with hidden variables. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1857999.1858119. (Cited on pages 207, 211, 214 and 217.)

[274] Ramanathan Narayanan, Bing Liu, and Alok Choudhary. Sentiment analysis of conditional sentences. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1*, EMNLP '09, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1699510.1699534. (Cited on pages 216 and 217.)

[275] National Institute of Standards and Technology. MUC data sets. URL http://www-nlpir.nist.gov/related_projects/muc/. [Online; accessed 12/2012]. (Cited on page 59.)

[276] Roberto Navigli and Paola Velardi. Semantic interpretation of terminological strings. In *Proceedings of the 6th International Conference on Terminology and Knowledge Engineering (TKE 2002)*, pages 95–100. Springer-Verlag, 2002. URL http://www.dsi.uniroma1.it/~navigli/pubs/TKE_2002_Navigli_Velardi.pdf. (Cited on page 101.)

[277] Roberto Navigli and Paola Velardi. Learning domain ontologies from document warehouses and dedicated web sites. *Computational Linguistics*, 30(2):151–179, June 2004. URL http://dx.doi.org/10.1162/089120104323093276. (Cited on page 100.)

[278] Tyler J. Neylon, Kerry L. Hannan, Ryan T. McDonald, Michael Wells, and Jeffrey C. Reynar. Domain specific sentiment classification. US Patent US2011/0252036, October 2011. URL http://www.google.com/patents/US20110252036. [Online; accessed 12/2012]. (Cited on page 26.)

[279] Vincent Ng, Sajib Dasgupta, and S. M. Niaz Arifin. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, COLING-ACL '06, pages 611–618, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P06/P06-2079.pdf. (Cited on pages 20, 203, 207, 210, 211, 213, 214 and 217.)

[280] Truc-Vien T. Nguyen and Alessandro Moschitti. Joint distant and direct supervision for relation extraction. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 732–740, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing. URL http://www.aclweb.org/anthology/I11-1082.pdf. (Cited on page 31.)

[281] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *Proceedings of the IJCAI-99 Workshop on Machine Learning for Information Filtering*, volume 1, 1999. URL http://www.cs.cmu.edu/~knigam/papers/maxent-ijcaiws99.pdf. (Cited on page 22.)

[282] Kamal Nigam and Matthew Hurst. Towards a robust metric of polarity. In *Computing Attitude and Affect in Text: Theory and Applications*, volume 20 of *The Information Retrieval Series*, chapter 20, pages 265–279. Springer-Verlag, Berlin/Heidelberg, 2006. URL http://dx.doi.org/10.1007/1-4020-4102-0_20. (Cited on pages 24 and 44.)

[283] Kamal Nigam, Andrew K. Mccallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2):103–134, May 2000. ISSN 08856125. URL http://dx.doi.org/10.1023/A:1007692713085. (Cited on page 264.)

[284] Daisuke Okanohara and Jun'ichi Tsujii. Assigning polarity scores to reviews using machine learning techniques. In *Natural Language Processing – IJCNLP 2005*, volume 3651 of *Lecture Notes in Computer Science*, pages 314–325. Springer Berlin / Heidelberg, 2005. URL http://dx.doi.org/10.1007/11562214_28. (Cited on pages 19 and 207.)

[285] F. Olsson. A literature survey of active machine learning in the context of natural language processing. Technical Report 06, Swedish Institute of Computer Science, April 2009. URL http://soda.swedish-ict.se/3600/1/SICS-T--2009-06--SE.pdf. (Cited on pages 29, 34 and 157.)

[286] A. Ortony, G. L. Clore, and A. Collins. *The Cognitive Structure of Emotions*. Cambridge Univ Press, 1990. ISBN 9780521386647. URL http://books.google.de/books?id=dA3JEEAp6TsC. (Cited on page 53.)

[287] C. E. Osgood, G. J. Suci, and P. H. Tannenbaum. *The Measurement of Meaning*. University of Illinois Press, 1957. ISBN 9780252745393. URL http://books.google.de/books?id=Qj8GeUrKZdAC. (Cited on page 53.)

[288] I. Ounis, C. Macdonald, and I. Soboroff. On the TREC blog track. In *Proceedings of the 2nd International Conference on Weblogs and Social Media (ICWSM)*, 2008. URL http://www.aaai.org/Papers/ICWSM/2008/ICWSM08-019.pdf. (Cited on pages 2, 13, 59 and 70.)

[289] Iadh Ounis, Maarten de Rijke, Craig Macdonald, Gilad Mishne, and Ian Soboroff. Overview of the TREC 2006 blog track. In *Proceedings of the 15th Text Retrieval Conference*, 2006. URL http://trec.nist.gov/pubs/trec15/papers/BLOG06.OVERVIEW.pdf. (Cited on pages 70, 71 and 207.)

[290] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. URL http://ilpubs.stanford.edu:8090/422/. (Cited on page 176.)

[291] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2010/pdf/385_Paper.pdf. (Cited on page 32.)

[292] Alexander Pak and Patrick Paroubek. Text representation using dependency tree subgraphs for sentiment analysis. In *Database Systems for Advanced Applications*, volume 6637 of *Lecture Notes in Computer Science*, pages 323–332. Springer Berlin / Heidelberg, 2011. URL http://dx.doi.org/10.1007/978-3-642-20244-5_31. (Cited on pages 214 and 217.)

[293] Sinno J. Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, New York, NY, USA, 2010. ACM. URL http://doi.acm.org/10.1145/1772690.1772767. (Cited on page 206.)

[294] B. Pang and L. Lee. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008. ISSN 1554-0669. URL http://www.cs.cornell.edu/home/llee/omsa/omsa.pdf. (Cited on pages 11, 13, 69, 206 and 209.)

[295] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics*, pages 271–278, Barcelona, Spain, July 2004. URL http://www.aclweb.org/anthology-new/P/P04/P04-1035.pdf. (Cited on pages 13, 18, 30, 31, 203, 205, 207, 208 and 222.)

[296] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 115–124, Morristown, NJ, USA, 2005. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1219840.1219855. (Cited on pages 19, 31 and 207.)

[297] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, volume 10 of *EMNLP '02*, pages 79–86, Morristown, NJ, USA, 2002. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1118693.1118704. (Cited on pages 11, 19, 30, 31, 203, 204, 205, 210, 213, 215 and 217.)

[298] Youngja Park, Roy J. Byrd, and Branimir K. Boguraev. Automatic glossary extraction: Beyond terminology identification. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1072228.1072370. (Cited on pages 101, 106, 108, 109, 110 and 111.)

[299] Alexandre Patry and Philippe Langlais. Corpus-based terminology extraction. In *Proceedings of the 7th International Conference on Terminology and Knowledge Engineering*, pages 313–321, Copenhagen, Denmark, August 2005. URL http://www-etud.iro.umontreal.ca/~patryale/papers/patry_langlais_2005_tke.pdf. (Cited on page 97.)

[300] John P. Pestian, Pawel Matykiewicz, Michelle Linn-Gust, Brett South, Ozlem Uzuner, Jan Wiebe, Kevin B. Cohen, John Hurdle, and Christopher Brew. Sentiment analysis of suicide notes: A shared task. *Biomedical Informatics Insights*, 5, 2012. URL www.la-press.com/sentiment-analysis-of-suicide-notes-a-shared-task-article-a3016. (Cited on page 13.)

[301] R. W. Picard. Affective computing. Technical Report 321, Perceptual Computing Section, Media Laboratory, Massachusetts Institute of Technology, 1995. URL http://affect.media.mit.edu/pdfs/95.picard.pdf. (Cited on page 13.)

[302] Robert Plutchik. The nature of emotions. *American Scientist*, 89(4):344+, July 2001. URL http://dx.doi.org/10.1511/2001.4.344. (Cited on pages 13 and 171.)

[303] Livia Polanyi and Annie Zaenen. Contextual valence shifters. In *Computing Attitude and Affect in Text: Theory and Applications*, volume 20 of *The Information Retrieval Series*, chapter 1, pages 1–10. Springer Netherlands, Berlin/Heidelberg, 2006. URL http://dx.doi.org/10.1007/1-4020-4102-0_1. (Cited on pages 24, 39, 49, 50, 51, 73, 209 and 277.)

[304] Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 339–346, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/H/H05/H05-1043.pdf. (Cited on pages 21, 22, 24, 39, 41, 55, 95, 99, 145 and 173.)

[305] Matthew Purver and Stuart Battersby. Experimenting with distant supervision for emotion classification. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–491, Avignon, France, April 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E12-1049.pdf. (Cited on page 32.)

[306] G. Qiu, B. Liu, J. Bu, and C. Chen. Expanding domain sentiment lexicon through double propagation. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence*, IJCAI'09, pages 1199–1204. Morgan Kaufmann Publishers Inc., 2009. URL http://dl.acm.org/citation.cfm?id=1661445.1661637. (Cited on pages 97, 169 and 173.)

[307] L. Qu, C. Toprak, N. Jakob, and I. Gurevych. Sentence level subjectivity and sentiment analysis experiments in NTCIR-7 MOAT challenge. In *Proceedings of the 7th*

NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering, and Cross-Lingual Information Access, 2008. URL http://www.informatik.tu-darmstadt.de/fileadmin/user_upload/Group_UKP/publikationen/2008/UKP07_MOAT_final_submission.pdf. (Cited on page 207.)

[308] Lizhen Qu, Georgiana Ifrim, and Gerhard Weikum. The bag-of-opinions method for review rating prediction from sparse text patterns. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 913–921, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1873781.1873884. (Cited on pages 207 and 210.)

[309] R. Quirk, S. Greenbaum, G. Leech, and J. Svartvik. *A Comprehensive Grammar of the English Language*. Longman, New York, 1985. (Cited on pages 53, 69 and 271.)

[310] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. URL http://dx.doi.org/10.1109/5.18626. (Cited on pages 22 and 97.)

[311] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336), 1971. URL http://www.jstor.org/stable/2284239. (Cited on page 97.)

[312] Delip Rao and Deepak Ravichandran. Semi-supervised polarity lexicon induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 675–682, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1609142. (Cited on pages 24, 171, 175, 176, 177, 190 and 200.)

[313] Rapid-I GmbH. Product Website: Rapid-I RapidSentilyzer. URL http://rapid-i.com/content/view/184/194/. [Online; accessed 12/2012]. (Cited on page 26.)

[314] AmirH Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin. Offensive language detection using multi-level classification. In *Advances in Artificial Intelligence*, volume 6085 of *Lecture Notes in Computer Science*, pages 16–27. Springer Berlin Heidelberg, 2010. URL http://dx.doi.org/10.1007/978-3-642-13059-5_5. (Cited on page 13.)

[315] Jonathon Read. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop*, ACLstudent '05, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1628960.1628969. (Cited on pages 32 and 206.)

[316] Jonathon Read, David Hope, and John Carroll. Annotating expressions of appraisal in English. In *Proceedings of the Linguistic Annotation Workshop*, LAW '07, pages 93–100, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1642059.1642074. (Cited on page 55.)

[317] Umaa Rebbapragada and Carla Brodley. Class noise mitigation through instance weighting. In *Machine Learning: ECML 2007*, volume 4701 of *Lecture Notes in Computer Science*, pages 708–715. Springer Berlin / Heidelberg, 2007. URL http://dx.doi.org/10.1007/978-3-540-74958-5_71. 10.1007/978-3-540-74958-5_71. (Cited on page 264.)

[318] George Reis, Sasha Blair-Goldensohn, and Ryan T. McDonald. Aspect-based sentiment summarization. US Patent US2009/0193328, July 2009. URL http://www.google.com/patents/US20090193328. [Online; accessed 12/2012]. (Cited on page 26.)

[319] R. Remus and C. Hänig. Towards well-grounded phrase-level polarity analysis. *Computational Linguistics and Intelligent Text Processing*, 6608:380–392, 2011. URL http://dx.doi.org/10.1007/978-3-642-19400-9_30. (Cited on page 51.)

[320] R. Remus, U. Quasthoff, and G. Heyer. SentiWS a publicly available German-language resource for sentiment analysis. In *Proceedings of the 7th International Language Resources and Evaluation*, LREC'10, 2010. URL http://www.lrec-conf.org/proceedings/lrec2010/pdf/490_Paper.pdf. (Cited on pages 171 and 172.)

[321] Repustate.com. Product Website: Repustate Sentiment Analysis API. URL https://www.repustate.com/sentiment-analysis/. [Online; accessed 12/2012]. (Cited on page 26.)

[322] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, volume 6323 of *Lecture Notes in Computer Science*, pages 148–163. Springer Berlin Heidelberg, 2010. URL http://dx.doi.org/10.1007/978-3-642-15939-8_10. (Cited on page 30.)

[323] Ellen Riloff and Janyce Wiebe. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1119355.1119369. (Cited on page 13.)

[324] Ellen Riloff, Janyce Wiebe, and Theresa Wilson. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the 7th Conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 25–32, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1119176.1119180. (Cited on pages 34 and 173.)

[325] Ellen Riloff, Siddharth Patwardhan, and Janyce Wiebe. Feature subsumption for opinion analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 440–448, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1610075.1610137. (Cited on pages 12, 214 and 217.)

[326] Juho Rousu, Craig Saunders, Sandor Szedmak, and John Shawe-Taylor. Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research*, 7, December 2006. URL http://dl.acm.org/citation.cfm?id=1248547.1248606. (Cited on page 147.)

[327] Victoria L. Rubin. Stating with certainty or stating with doubt: Intercoder reliability results for manual annotation of epistemically modalized statements. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, NAACL-Short '07, pages 141–144, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1614108.1614144. (Cited on page 51.)

[328] Miguel E. Ruiz and Padmini Srinivasan. Hierarchical text categorization using neural networks. *Information Retrieval*, 5:87–118, 2002. URL http://dx.doi.org/10.1023/A:1012782908347. (Cited on page 324.)

[329] J. A. Russell. Pancultural aspects of the human conceptual organization of emotions. *Journal of Personality and Social Psychology*, 45(6):1281, 1983. URL http://psycnet.apa.org/doi/10.1037/0022-3514.45.6.1281. (Cited on page 53.)

[330] Salesforce.com Inc. Product Website: Salesforce Marketing Cloud. URL http://www.salesforce.com/marketing-cloud/overview/. [Online; accessed 12/2012]. (Cited on page 26.)

[331] SAS Institute Inc. Technical Info: SAS Sentiment Analysis. URL http://www.sas.com/resources/whitepaper/wp_27999.pdf. [Online; accessed 12/2012]. (Cited on page 26.)

[332] Christopher Scaffidi, Kevin Bierhoff, Eric Chang, Mikhael Felker, Herman Ng, and Chun Jin. Red opal: product-feature scoring from reviews. *Proceedings of the 8th ACM Conference on Electronic Commerce*, pages 182–191, 2007. URL http://dx.doi.org/10.1145/1250910.1250938. (Cited on pages 22, 96 and 99.)

[333] Robert E. Schapire and Yoram Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39:135–168, 2000. URL http://dx.doi.org/10.1023/A:1007649029923. (Cited on pages 148 and 323.)

[334] Klaus R. Scherer. What are emotions? And how can they be measured? *Social Science Information*, 44(4):695–729, 2005. URL http://ssi.sagepub.com/content/44/4/695.abstract. (Cited on page 13.)

[335] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7), 2001. URL http://dx.doi.org/10.1162/089976601750264965. (Cited on page 226.)

[336] F. Sclano and P. Velardi. TermExtractor: A web application to learn the shared terminology of emergent web communities. In *Enterprise Interoperability II*, chapter 32, pages 287–290. Springer London, London, 2007. URL http://dx.doi.org/10.1007/978-1-84628-858-6_32. (Cited on page 101.)

[337] Sam Scott and Stan Matwin. Text classification using WordNet hypernyms. In *Workshop on the Usage of WordNet in Natural Language Processing Systems*, Coling-ACL, August 1998. URL http://acl.ldc.upenn.edu/W/W98/W98-0706.pdf. (Cited on pages 212 and 217.)

[338] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34:1–47, March 2002. URL http://doi.acm.org/10.1145/505282.505283. (Cited on pages 19, 149 and 209.)

[339] Y. Seki, D. K. Evans, L. W. Ku, H. H. Chen, N. Kando, and C. Y. Lin. Overview of opinion analysis pilot task at NTCIR-6. In *Proceedings of 6th NTCIR Workshop*, pages 265–278, 2007. URL http://research.nii.ac.jp/ntcir/ntcir-ws6/OnlineProceedings/NTCIR/81.pdf. (Cited on pages 59 and 70.)

[340] Y. Seki, D. K. Evans, L. W. Ku, L. Sun, H. H. Chen, N. Kando, and C. Y. Lin. Overview of multilingual opinion analysis task at NTCIR-7. In *Proceedings of the 7th NTCIR Workshop*, 2008. URL http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings7/pdf/NTCIR7/C2/MOAT/01-NTCIR-OV-MOAT-SekiY.pdf. (Cited on pages 59 and 70.)

[341] Yohei Seki, Lun-Wei Ku, Le Sun, Hsin-Hsi Chen, and Noriko Kando. Overview of multilingual opinion analysis task at NTCIR-8: A step toward cross lingual opinion analysis. In *Proceedings of the 8th NTCIR Workshop*, pages 209–220, 2009. URL http://research.nii.ac.jp/~ntcadm/workshop/OnlineProceedings8/NTCIR/01-NTCIR8-OV-MOAT-SekiY.pdf. (Cited on page 70.)

[342] Sentiment140. Product Website: Sentiment 140 Twitter Polarity. URL http://www.sentiment140.com/. [Online; accessed 12/2012]. (Cited on page 26.)

[343] Burr Settles. Active learning literature survey. Technical Report 1648, University of Wisconsin–Madison, 2010. URL http://research.cs.wisc.edu/techreports/2009/TR1648.pdf. (Cited on pages 29, 34 and 157.)

[344] Mostafa A. Shaikh, Helmut Prendinger, and Mitsuru Ishizuka. Sentiment assessment of text by analyzing linguistic features and contextual valence assignment. *Applied Artificial Intelligence*, 22(6):558–601, 2008. URL http://www.tandfonline.com/doi/abs/10.1080/08839510802226801. (Cited on page 203.)

[345] Kazutaka Shimada and Tsutomu Endo. Seeing several stars: A rating inference task for a document containing several evaluation criteria. In *Advances in Knowledge Discovery and Data Mining*, volume 5012 of *Lecture Notes in Computer Science*, pages 1006–1014. Springer Berlin / Heidelberg, 2008. URL http://dx.doi.org/10.1007/978-3-540-68125-0_106. (Cited on pages 19 and 207.)

[346] Hyun J. Shin, Dong-Hwan Eom, and Sung-Shick Kim. One-class support vector machines - an application in machine fault detection and classification. *Computers & Industrial Engineering*, 48(2), March 2005. URL http://dx.doi.org/10.1016/j.cie.2005.01.009. (Cited on page 226.)

[347] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. Cheap and fast – but is it good? Evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, Hawaii, October 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D08-1027.pdf. (Cited on pages 29 and 34.)

[348] Benjamin Snyder and Regina Barzilay. Multiple aspect ranking using the good grief algorithm. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 300–307, Rochester, New York, April 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N/N07/N07-1038.pdf. (Cited on pages 19, 31 and 207.)

[349] Catherine Soanes and Angus Stevenson. *Oxford Dictionary of English*. Oxford University Press, August 2005. (Cited on pages 280 and 285.)

[350] Socialmention.com. Product Website: Socialmention Twitter Polarity. URL http://www.socialmention.com/. [Online; accessed 12/2012]. (Cited on page 26.)

[351] S. Somasundaran. *Discourse-level Relations For Opinion Analysis*. PhD thesis, University of Pittsburgh, 2010. URL http://scicomp.pitt.edu/~wiebe/pubs/papers/somasundaranThesis.pdf. (Cited on page 44.)

[352] S. Somasundaran, T. Wilson, J. Wiebe, and V. Stoyanov. QA with attitude: Exploiting opinion type analysis for improving question answering in on-line discussions and the news. In *Proceedings of the International Conference on Weblogs and Social Media*, ICWSM '07, 2007. URL http://icwsm.org/papers/2--Somasundaran-Wilson-Wiebe-Stoyanov.pdf. (Cited on page 2.)

[353] Swapna Somasundaran and Janyce Wiebe. Recognizing stances in online debates. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, ACL '09, pages 226–234, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1687878.1687912. (Cited on pages 2 and 14.)

[354] Swapna Somasundaran and Janyce Wiebe. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, CAAGET '10, pages 116–124, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1860631.1860645. (Cited on page 203.)

[355] Swapna Somasundaran, Josef Ruppenhofer, and Janyce Wiebe. Discourse level opinion relations: An annotation study. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, SIGdial '08, pages 129–137, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1622092. (Cited on page 54.)

[356] Swapna Somasundaran, Galileo Namata, Janyce Wiebe, and Lise Getoor. Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, EMNLP '09, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1699510.1699533. (Cited on page 208.)

[357] Mohammad Sorower. A literature survey on algorithms for multi-label learning. Technical report, Oregon State University, 2010. URL http://people.oregonstate.edu/~sorowerm/pdf/Qual-Multilabel-Shahed-CompleteVersion.pdf. (Cited on page 323.)

[358] Valentin I. Spitkovsky, Daniel Jurafsky, and Hiyan Alshawi. Profiting from mark-up: Hyper-Text annotations for guided parsing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1278–1287, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P10-1130. (Cited on page 32.)

[359] Adam Stepinski and Vibhu Mittal. A fact/opinion classifier for news articles. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, New York, NY, USA, 2007. ACM. URL http://doi.acm.org/10.1145/1277741.1277919. (Cited on pages 13 and 172.)

[360] P. J. Stone, D. C. Dunphy, M. S. Smith, D. M. Ogilvie, and Others. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press Cambridge, MA, 1966. (Cited on pages 24, 171, 175 and 211.)

[361] Veselin Stoyanov and Claire Cardie. Annotating topics of opinions. In *Proceedings of the 6TH International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2008/pdf/813_paper.pdf. (Cited on page 69.)

[362] Veselin Stoyanov and Claire Cardie. Topic identification for fine-grained opinion analysis. In *Proceedings of the 22nd International Conference on Computational Linguistics*, volume 1 of *COLING '08*, pages 817–824, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1599081.1599184. (Cited on page 12.)

[363] Veselin Stoyanov, Claire Cardie, and Janyce Wiebe. Multi-Perspective question answering using the OpQA corpus. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1220575.1220691. (Cited on pages 2, 13 and 14.)

[364] C. Strapparava and A. Valitutti. WordNet-affect: An affective extension of WordNet. In *Proceedings of LREC*, volume 4, pages 1083–1086, 2004. URL http://www.lrec-conf.org/proceedings/lrec2004/pdf/369.pdf. (Cited on page 171.)

[365] Carlo Strapparava and Rada Mihalcea. Learning to identify emotions in text. In *Proceedings of the 2008 ACM symposium on Applied Computing*, SAC '08, New York, NY, USA, 2008. ACM. URL http://doi.acm.org/10.1145/1363686.1364052. (Cited on pages 13 and 171.)

[366] Olga Streibel and Malgorzata Mochol. Trend ontology for knowledge-based trend mining in textual information. In *Proceedings of the 2010 7TH International Conference on Information Technology: New Generations*, ITNG '10, pages 1285–1288, Washington, DC, USA, 2010. IEEE Computer Society. URL http://dx.doi.org/10.1109/ITNG.2010.232. (Cited on page 145.)

[367] Fangzhong Su and Katja Markert. Subjectivity recognition on word senses via semi-supervised mincuts. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Boulder, Colorado, June 2009. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N/N09/N09-1001.pdf. (Cited on page 173.)

[368] Qi Su, Xinying Xu, Honglei Guo, Zhili Guo, Xian Wu, Xiaoxun Zhang, Bin Swen, and Zhong Su. Hidden sentiment association in chinese web opinion mining. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 959–968, New York, NY, USA, 2008. ACM. URL http://doi.acm.org/10.1145/1367497.1367627. (Cited on pages 22, 96, 97 and 99.)

[369] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. YAGO: A large ontology from Wikipedia and WordNet. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3):203–217, 2008. URL http://www.sciencedirect.com/science/article/pii/S1570826808000437. World Wide Web Conference 2007 (Semantic Web Track). (Cited on page 30.)

[370] M. Surdeanu, D. McClosky, J. Tibshirani, J. Bauer, A. X. Chang, V. I. Spitkovsky, and C. D. Manning. A simple distant supervision approach for the TAC-KBP slot filling task. In *Proceedings of the 3rd Text Analysis Conference*, TAC '10. National Institute of Standards and Technology, November 2010. URL http://www.nist.gov/tac/publications/2010/participant.papers/Stanford.proceedings.pdf. (Cited on page 30.)

[371] J. Surowiecki. *The Wisdom of Crowds: Why the Many Are Smarter Than the Few*. Random House, Inc., 2004. (Cited on page 34.)

[372] Sysomos Inc. Product Website: Sysomos Social Media Monitoring Dashboard. URL http://www.sysomos.com/products/overview/heartbeat/. [Online; accessed 12/2012]. (Cited on page 26.)

[373] M. Taboada and J. Grieve. Analyzing appraisal automatically. In *Proceedings of AAAI Spring Symposium on Exploring Attitude and Affect in Text (AAAI Technical Report SS-04-07)*, pages 158–161, 2004. URL http://www.aaai.org/Papers/Symposia/Spring/2004/SS-04-07/SS04-07-029.pdf. (Cited on page 55.)

[374] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 1(August):1–41, 2010. URL http://dx.doi.org/doi:10.1162/COLI_a_00049. (Cited on page 203.)

[375] Oscar Täckström and Ryan McDonald. Discovering fine-grained sentiment with latent variable structured prediction models. In *Advances in Information Retrieval*, volume 6611 of *Lecture Notes in Computer Science*, pages 368–374. Springer Berlin / Heidelberg, 2011. URL http://dx.doi.org/10.1007/978-3-642-20161-5_37. (Cited on pages 207 and 208.)

[376] Hiroya Takamura, Takashi Inui, and Manabu Okumura. Extracting semantic orientations of words using spin model. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 133–140, Morristown, NJ, USA, 2005. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1219840.1219857. (Cited on page 172.)

[377] Hiroya Takamura, Takashi Inui, and Manabu Okumura. Extracting semantic orientations of phrases from dictionary. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 292–299, Rochester, New York, April 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N/N07/N07-1037.pdf. (Cited on page 172.)

[378] Chade-Meng Tan, Yuan-Fang Wang, and Chan-Do Lee. The use of bigrams to enhance text categorization. *Information Processing and Management*, 38(4), July 2002. URL http://dx.doi.org/10.1016/S0306-4573(01)00045-0. (Cited on page 149.)

[379] David M. Tax. *One-class Classification — Concept-learning in the Absence of Counter-examples*. PhD thesis, TU Delft, 2001. URL http://homepage.tudelft.nl/n9d04/thesis.pdf. (Cited on pages 225, 226 and 264.)

[380] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12), 2010. URL http://dx.doi.org/10.1002/asi.21416. (Cited on page 203.)

[381] Matt Thomas, Bo Pang, and Lillian Lee. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1610075.1610122. (Cited on pages 12, 32 and 205.)

[382] Ivan Titov and Ryan McDonald. Modeling online reviews with multi-grain topic models. In *Proceeding of the 17th International Conference on World Wide Web*, WWW '08, pages 111–120, New York, NY, USA, 2008. ACM. URL http://dx.doi.org/10.1145/1367497.1367513. (Cited on pages 57, 314 and 316.)

[383] Ivan Titov and Ryan McDonald. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL-08: HLT*, pages 308–316, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P08/P08-1036.pdf. (Cited on pages 23, 26 and 315.)

[384] Ryoko Tokuhisa, Kentaro Inui, and Yuji Matsumoto. Emotion classification using massive examples extracted from the Web. In *Proceedings of the 22nd International Conference on Computational Linguistics*, COLING '08, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1599081.1599192. (Cited on page 13.)

[385] Takashi Tomokiyo and Matthew Hurst. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, MWE '03, pages 33–40, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1119282.1119287. (Cited on page 101.)

[386] Cigdem Toprak, Niklas Jakob, and Iryna Gurevych. Sentence and expression level annotation of opinions in user-generated discourse. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 575–584, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1858681.1858740. (Cited on pages 13, 24, 39, 44, 49, 55, 70 and 73.)

[387] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1073445.1073478. (Cited on page 102.)

[388] G. Tsoumakas and I. Katakis. Multi-Label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3), 2007. URL http://dx.doi.org/10.4018/jdwm.2007070101. (Cited on page 148.)

[389] Peter Turney. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. In *Machine Learning: ECML 2001*, volume 2167 of *Lecture Notes in Computer Science*, pages 491–502. Springer Berlin / Heidelberg, 2001. URL http://dx.doi.org/10.1007/3-540-44795-4_42. (Cited on page 95.)

[390] Peter D. Turney. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Morristown, NJ, USA, 2002. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1073083.1073153. (Cited on pages 11, 19, 172 and 173.)

[391] Peter D. Turney and Michael L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315–346, 2003. URL http://dx.doi.org/10.1145/944012.944013. (Cited on pages 171, 172, 174 and 175.)

[392] Twitrratr.com. Product Website: Twitrratr Twitter Polarity. URL http://www.twitrratr.com/. [Online; accessed 12/2012]. (Cited on page 26.)

[393] Adrian Ulges, Christian Schulze, Daniel Keysers, and Thomas Breuel. Identifying relevant frames in weakly labeled videos for training concept detectors. In *Proceedings of the 2008 International Conference on Content-Based Image and Video Retrieval*, CIVR '08, pages 9–16, New York, NY, USA, 2008. ACM. URL http://doi.acm.org/10.1145/1386352.1386358. (Cited on page 29.)

[394] Melissa L-H Võ, Markus Conrad, Lars Kuchinke, Karolina Urton, Markus J Hofmann, and Arthur M Jacobs. The Berlin Affective Word List Reloaded (BAWL-R). *Behavior Research Methods*, 41(2):534–538, 2009. URL http://www.ncbi.nlm.nih.gov/pubmed/19363195. (Cited on page 171.)

[395] A. Valitutti, C. Strapparava, and O. Stock. Developing affective lexical resources. *PsychNology Journal*, 2(1), 2004. URL http://www.psychnology.org/File/PSYCHNOLOGY_JOURNAL_2_1_VALITUTTI.pdf. (Cited on pages 171, 172 and 198.)

[396] Jeroen van der Meer and Flavius Frasincar. Automatic review identification on the Web using pattern recognition. *Software: Practice and Experience*, 2012. URL http://dx.doi.org/10.1002/spe.2152. (Cited on pages 13 and 206.)

[397] Baptist Vandersmissen. Automated detection of offensive language behavior on social networking sites. Master's thesis, Universiteit Gent, 2012. URL http://lib.ugent.be/fulltxt/RUG01/001/887/239/RUG01-001887239_2012_0001_AC.pdf. (Cited on page 13.)

[398] Paola Velardi, Roberto Navigli, and Pierluigi D'Amadio. Mining the web to create specialized glossaries. *IEEE Intelligent Systems*, 23(5):18–25, 2008. ISSN 1541-1672. URL http://dx.doi.org/10.1109/MIS.2008.88. (Cited on pages 100 and 101.)

[399] Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. The viability of Web-derived polarity lexicons. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 777–785, Los Angeles, California, June 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N10-1119. (Cited on pages 24, 171, 172, 173, 174 and 175.)

[400] L. von Ahn, R. Liu, and M. Blum. Peekaboom: A game for locating objects in images. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 55–64. ACM, 2006. URL http://dx.doi.org/10.1145/1124772.1124782. (Cited on page 34.)

[401] Luis von Ahn. Games with a purpose. *Computer*, 39(6):92–94, 2006. ISSN 0018-9162. URL http://dx.doi.org/10.1109/MC.2006.196. (Cited on page 34.)

[402] Hongning Wang, Yue Lu, and Chengxiang Zhai. Latent aspect rating analysis on review text data: A rating regression approach. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 783–792, New York, NY, USA, 2010. ACM. URL http://doi.acm.org/10.1145/1835804.1835903. (Cited on page 31.)

[403] Hongning Wang, Yue Lu, and ChengXiang Zhai. Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 618–626, New York, NY, USA, 2011. ACM. URL http://doi.acm.org/10.1145/2020408.2020505. (Cited on page 31.)

[404] Andreas S. Weigend, Erik D. Wiener, and Jan O. Pedersen. Exploiting hierarchy in text categorization. *Information Retrieval*, 1:193–216, 1999. URL http://dx.doi.org/10.1023/A:1009983522080. (Cited on page 147.)

[405] G. M. Weiss, K. McCarthy, and B. Zabar. Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs. In *International Conference on Data Mining*, 2007. URL http://storm.cis.fordham.edu/~gweiss/papers/dmin07-weiss.pdf. (Cited on page 157.)

[406] D. S. Weld, R. Hoffmann, and F. Wu. Using Wikipedia to bootstrap open information extraction. *ACM SIGMOD Record*, 37(4):62–68, 2009. URL http://dx.doi.org/10.1145/1519103.1519113. (Cited on page 34.)

[407] Joachim Wermter and Udo Hahn. Finding new terminology in very large corpora. In *Proceedings of the 3rd International Conference on Knowledge Capture*, K-CAP '05, pages 137–144, New York, NY, USA, 2005. ACM. URL http://dx.doi.org/10.1145/1088622.1088648. (Cited on page 101.)

[408] C. Whissell. The dictionary of affect in language. *Emotion: Theory, Research, and Experience*, 4, 1989. (Cited on page 171.)

[409] Peter R. R. White and J. R. Martin. *The Language of Evaluation: Appraisal in English*. Palgrave Macmillan, London/New York, 2005. (Cited on page 54.)

[410] Matthew Whitehead and Larry Yaeger. Building a general purpose cross-domain sentiment mining model. *Computer Science and Information Engineering, World Congress on*, 4:472–476, 2009. URL http://dx.doi.org/10.1109/CSIE.2009.754. (Cited on page 264.)

[411] Casey Whitelaw, Navendu Garg, and Shlomo Argamon. Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 625–631, New York, NY, USA, 2005. ACM. URL http://dx.doi.org/10.1145/1099554.1099714. (Cited on page 55.)

[412] J. Wiebe, E. Breck, C. Buckley, C. Cardie, P. Davis, B. Fraser, D. Litman, D. Pierce, E. Riloff, and T. Wilson. NRRC summer workshop on multiple-perspective question answering — final report. Technical report, Northeast Regional Research Center, 2002. URL http://www.cs.cornell.edu/Info/People/cardie/papers/mpqa-finalreport-02.pdf. (Cited on pages 19 and 69.)

[413] Janyce Wiebe. The MPQA opinion corpus. URL http://mpqa.cs.pitt.edu/. [Online; accessed 12/2012]. (Cited on pages 53, 70, 175, 208 and 264.)

[414] Janyce Wiebe. Learning subjective adjectives from corpora. In *Proceedings of the 17th National Conference on Artificial Intelligence*, pages 735–740. AAAI Press, 2000. URL http://dl.acm.org/citation.cfm?id=647288.721121. (Cited on pages 170, 172 and 173.)

[415] Janyce Wiebe and Rada Mihalcea. Word sense and subjectivity. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1220175.1220309. (Cited on page 170.)

[416] Janyce Wiebe, Theresa Wilson, Rebecca Bruce, Matthew Bell, and Melanie Martin. Learning subjective language. *Computational Linguistics*, 30:277–308, September 2004. URL http://dx.doi.org/10.1162/0891201041850885. (Cited on pages 53 and 207.)

[417] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2):165–210, May 2005. ISSN 1574-020X. URL http://dx.doi.org/10.1007/s10579-005-7880-9. (Cited on pages 13, 14, 39, 53, 55, 69 and 70.)

[418] Janyce M. Wiebe. *Recognizing Subjective Sentences: A Computational Investigation of Narrative Text*. PhD thesis, State University of New York at Buffalo, Buffalo, NY, USA, 1990. (Cited on page 53.)

[419] Janyce M. Wiebe. Identifying subjective characters in narrative. In *Proceedings of the 13th Conference on Computational Linguistics - Volume 2*, COLING '90, pages 401–406, Stroudsburg, PA, USA, 1990. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/997939.998008. (Cited on page 53.)

[420] Janyce M. Wiebe. Tracking point of view in narrative. *Computational Linguistics*, 20:233–287, June 1994. URL http://portal.acm.org/citation.cfm?id=972525.972529. (Cited on page 53.)

[421] Janyce M. Wiebe and William J. Rapaport. A computational theory of perspective and reference in narrative. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, ACL '88, Stroudsburg, PA, USA, 1988. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/982023.982039. (Cited on page 11.)

[422] Michael Wiegand and Dietrich Klakow. The role of knowledge-based features in polarity classification at sentence level. In *Proceedings of the 22nd International Florida Artificial Intelligence Research Society Conference*. AAAI Press, May 2009. URL http://aaai.org/ocs/index.php/FLAIRS/2009/paper/viewFile/24/304. (Cited on pages 24, 207, 211, 212, 213, 215, 216, 217, 230 and 235.)

[423] Michael Wiegand and Dietrich Klakow. Topic-Related polarity classification of blog sentences. In *Progress in Artificial Intelligence*, volume 5816 of *Lecture Notes in Computer Science*, pages 658–669. Springer Berlin / Heidelberg, 2009. URL http://dx.doi.org/10.1007/978-3-642-04686-5_54. (Cited on page 207.)

[424] Michael Wiegand and Dietrich Klakow. Convolution kernels for opinion holder extraction. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1857999.1858120. (Cited on page 14.)

[425] Michael Wiegand, Alexandra Balahur, Benjamin Roth, Dietrich Klakow, and Andrés Montoyo. A survey on the role of negation in sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, NeSp-NLP '10, pages 60–68, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1858970. (Cited on pages 216 and 217.)

[426] Wikipedia. Crowdsourcing — Wikipedia, The Free Encyclopedia, 2012. URL http://en.wikipedia.org/w/index.php?title=Crowdsourcing&oldid=503317581. [Online; accessed 12/2012]. (Cited on page 29.)

[427] Wikipedia. Affective Computing — Wikipedia, The Free Encyclopedia, 2012. URL http://en.wikipedia.org/w/index.php?title=Affective_computing&oldid=528531138. [Online; accessed 12/2012]. (Cited on page 13.)

[428] Wikipedia. Sentiment Analysis — Wikipedia, The Free Encyclopedia, 2012. URL http://en.wikipedia.org/w/index.php?title=Sentiment_analysis&oldid=527200479. [Online; accessed 12/2012]. (Cited on page 12.)

[429] R. R. Wilcox. *Fundamentals of Modern Statistical Methods: Substantially Improving Power and Accuracy*. Springer, 2010. URL http://books.google.de/books?id=uUNGzhdxk0kC. (Cited on page 112.)

[430] S. S. Wilks. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics*, 9(1):60–62, 1938. URL http://dx.doi.org/10.1214/aoms/1177732360. (Cited on page 110.)

[431] Yorick Wilks and Janusz Bien. Beliefs, points of view, and multiple environments. *Cognitive Science*, 7(2):95–119, 1983. URL http://www.sciencedirect.com/science/article/pii/S036402138380007X. (Cited on page 11.)

[432] T. Wilson and J. Wiebe. Annotating opinions in the world press. In *Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue (SIGdial-03)*, pages 13–22, 2003. URL http://www.aclweb.org/anthology-new/W/W03/W03-2102.pdf. (Cited on pages 14, 53 and 69.)

[433] Theresa Wilson. Annotating subjective content in meetings. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA). URL http://lrec-conf.org/proceedings/lrec2008/pdf/693_paper.pdf. (Cited on pages 24, 44 and 70.)

[434] Theresa Wilson. *Fine-grained Subjectivity and Sentiment Analysis: Recognizing the Intensity, Polarity, and Attitudes of Private States*. PhD thesis, University of Pittsburgh, 2008. URL http://d-scholarship.pitt.edu/7563/. (Cited on pages 13, 24 and 70.)

[435] Theresa Wilson and Janyce Wiebe. Annotating attributions and private states. In *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*, CorpusAnno '05, pages 53–60, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1608829.1608837. (Cited on pages 53, 54, 69 and 271.)

[436] Theresa Wilson, Janyce Wiebe, and Rebecca Hwa. Just how mad are you? Finding strong and weak opinion clauses. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, pages 761–767. AAAI Press, 2004. URL http://portal.acm.org/citation.cfm?id=1597148.1597270. (Cited on pages 13, 53 and 208.)

[437] Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. OpinionFinder: A system for subjectivity analysis. In *Proceedings of HLT/EMNLP 2005 Demonstration Abstracts*, HLT-Demo '05, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1225733.1225751. (Cited on page 14.)

[438] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1220575.1220619. (Cited on pages 13, 24, 49, 53, 54, 69, 171, 175, 190, 208 and 211.)

[439] Theresa Wilson, Janyce Wiebe, and Rebecca Hwa. Recognizing strong and weak opinion clauses. *Computational Intelligence*, 22(2), 2006. URL http://dx.doi.org/10.1111/j.1467-8640.2006.00275.x. (Cited on page 13.)

[440] Wilson Wong. Determination of unithood and termhood for term recognition. In *Handbook of Research on Text and Web Mining Technologies*. IGI Global, 2009. URL http://dx.doi.org/10.4018/978-1-59904-990-8.ch030. (Cited on pages 100 and 101.)

[441] Fei Wu and Daniel S. Weld. Autonomously semantifying Wikipedia. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, CIKM '07, pages 41–50, New York, NY, USA, 2007. ACM. URL http://doi.acm.org/10.1145/1321440.1321449. (Cited on page 30.)

[442] Fei Wu and Daniel S. Weld. Automatically refining the Wikipedia infobox ontology. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 635–644, New York, NY, USA, 2008. ACM. URL http://doi.acm.org/10.1145/1367497.1367583. (Cited on page 30.)

[443] Fei Wu and Daniel S. Weld. Open information extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P10-1013.pdf. (Cited on page 30.)

[444] G. Wu and E. Y. Chang. Class-boundary alignment for imbalanced dataset learning. In *ICML 2003 Workshop on Learning from Imbalanced Data Sets II*, 2003. URL http://www.site.uottawa.ca/~nat/Workshop2003/Wu-final.pdf. (Cited on page 157.)

[445] Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. Phrase dependency parsing for opinion mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, EMNLP '09, pages 1533–1541, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1699648.1699700. (Cited on pages 22, 95, 96, 98 and 99.)

[446] Changhua Yang, Kevin H. Lin, and Hsin-Hsi Chen. Emotion classification using web blog corpora. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, WI '07, Washington, DC, USA, 2007. IEEE Computer Society. URL http://dx.doi.org/10.1109/WI.2007.51. (Cited on page 13.)

[447] Y. Yang, Y. Xia, Y. Chi, and R. R. Muntz. Learning naive Bayes classifier from noisy data. Technical Report 030056, University of California Los Angeles, 2003. URL ftp://webarchive.cs.ucla.edu/tech-report/2003-reports/030056.pdf. (Cited on page 264.)

[448] Yiming Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1:69–90, 1999. URL http://dx.doi.org/10.1023/A:1009982220290. (Cited on page 323.)

[449] Shiren Ye and Tat-Seng Chua. Learning object models from semistructured Web documents. *IEEE Transactions on Knowledge and Data Engineering*, 18:334–349, 2006. URL http://doi.ieeecomputersociety.org/10.1109/TKDE.2006.47. (Cited on page 98.)

[450] Ainur Yessenalina and Claire Cardie. Compositional matrix-space models for sentiment analysis. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D11-1016.pdf. (Cited on page 171.)

[451] Ainur Yessenalina, Yisong Yue, and Claire Cardie. Multi-level structured models for document-level sentiment classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1870658.1870760. (Cited on page 208.)

[452] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack. Sentiment Analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Third IEEE International Conference on Data Mining*, pages 427–434, Washington, DC, USA, 2003. IEEE Comput. Soc. URL http://dx.doi.org/10.1109/ICDM.2003.1250949. (Cited on pages 11, 22, 96, 99, 110, 113, 114, 208, 305 and 306.)

[453] Jeonghee Yi and Wayne Niblack. Sentiment mining in WebFountain. In *Proceedings of the 21st International Conference on Data Engineering*, ICDE '05, pages 1073–1083, Washington, DC, USA, 2005. IEEE Computer Society. URL http://dx.doi.org/10.1109/ICDE.2005.132. (Cited on pages 12, 102, 103 and 203.)

[454] Bei Yu, Stefan Kaufmann, and Daniel Diermeier. Exploring the characteristics of opinion expressions for political opinion classification. In *Proceedings of the 2008 International Conference on Digital Government Research*, pages 82–91. Digital Government Society of North America, 2008. URL http://portal.acm.org/citation.cfm?id=1367832.1367848. (Cited on page 2.)

[455] Hong Yu and Vasileios Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the*

2003 Conference on Empirical Methods in Natural Language Processing, EMNLP '03, pages 129–136, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1119355.1119372. (Cited on pages 11, 13, 14 and 207.)

[456] Hwanjo Yu, Jiawei Han, and Kevin C. Chang. PEBL: Web page classification without negative examples. *IEEE Transactions on Knowledge and Data Engineering*, 16(1), January 2004. URL http://dx.doi.org/10.1109/TKDE.2004.1264823. (Cited on pages 226 and 264.)

[457] Jianxing Yu, Zheng-Jun Zha, Meng Wang, Kai Wang, and Tat-Seng Chua. Domain-assisted product aspect hierarchy generation: Towards hierarchical organization of unstructured consumer reviews. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 140–150, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D11-1013.pdf. (Cited on pages 98 and 99.)

[458] Uri Zernik. *Lexical Acquisition: Exploiting On-line Resources to Build a Lexicon*. Lawrence Erlbaum, 1991. URL http://books.google.de/books?id=Gj50bzndWhwC. (Cited on pages 95 and 101.)

[459] Lei Zhang, Bing Liu, Suk H. Lim, and Eamonn O'Brien-Strain. Extracting and ranking product features in opinion documents. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 1462–1470, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1944566.1944733. (Cited on page 97.)

[460] Min Zhang and Xingyao Ye. A generation model to unify topic relevance and lexicon-based sentiment for opinion retrieval. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, New York, NY, USA, 2008. ACM. URL http://doi.acm.org/10.1145/1390334.1390405. (Cited on page 14.)

[461] Min-Ling Zhang and Zhi-Hua Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18: 1338–1351, 2006. URL http://doi.ieeecomputersociety.org/10.1109/TKDE.2006.162. (Cited on page 148.)

[462] Min-Ling Zhang and Zhi-Hua Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007. URL http://www.sciencedirect.com/science/article/pii/S0031320307000027. (Cited on page 148.)

[463] Zhu Zhang and Balaji Varadarajan. Utility scoring of product reviews. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM '06, pages 51–57, New York, NY, USA, 2006. ACM. URL http://doi.acm.org/10.1145/1183614.1183626. (Cited on page 20.)

[464] Lili Zhao and Chunping Li. Ontology based opinion mining for movie reviews. In *Knowledge Science, Engineering and Management*, volume 5914 of *Lecture Notes in Computer Science*, pages 204–214. Springer Berlin / Heidelberg, 2009. URL http://dx.doi.org/10.1007/978-3-642-10488-6_22. (Cited on page 146.)

[465] Wayne X. Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 56–65, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1870658.1870664. (Cited on pages 23, 56 and 315.)

[466] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University, 2002. URL http://www.cs.cmu.edu/~zhuxj/pub/CMU-CALD-02-107.pdf. (Cited on page 176.)

[467] X. Zhu, X. Wu, and Q. Chen. Eliminating class noise in large datasets. In *Proceedings of the 20th International Conference on Machine Learning*, volume 20 of *ICML-2003*, page 920, 2003. URL http://www.aaai.org/Papers/ICML/2003/ICML03-119.pdf. (Cited on page 264.)

[468] Li Zhuang, Feng Jing, and Xiao-Yan Zhu. Movie review mining and summarization. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM '06, pages 43–50, New York, NY, USA, 2006. ACM. URL http://doi.acm.org/10.1145/1183614.1183625. (Cited on pages 14, 70, 72, 146 and 203.)

[469] Cäcilia Zirn, Mathias Niepert, Heiner Stuckenschmidt, and Michael Strube. Fine-grained sentiment analysis with structural features. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing. URL http://www.aclweb.org/anthology/I11-1038.pdf. (Cited on pages 24 and 208.)