Part I

The Field of Information Extraction

2 Information Extraction

Most of the information stored in digital form is hidden in natural language texts. Extracting and storing it in a formal representation (e.g. in form of relations in databases) allows efficient querying and easy administration of the extracted data. The area of *information extraction* (IE) comprises techniques, algorithms and methods for performing two important tasks: finding (identifying) the desired, relevant data and storing it in appropriate form for future use.

Unlike *text understanding* and other related areas (cf. below), information extraction does not try to handle any potentially relevant information; instead it requires a predefined *target schema* that specifies which kinds of information should be extracted and how they should be stored. The reliance on a target schema also is a prerequisite for storing the extracted information in a way that allows *structured queries* (typically in a database).[1]

2.1 Information Retrieval, Text Mining, and Other Related Areas

A precursor of information extraction was the field of *text understanding* (or message understanding) which had the more ambitious aim of completely representing the contents of texts. To stimulate research in this area was the original goal of the *Message Understanding Conferences* (MUC) held from 1987 through 1998 under the auspices of the US government (ARPA/DARPA).

The field of *information retrieval* (IR) also has deeply influenced the development of modern IE systems, especially by pioneering the usage of statistical techniques and shallow (instead of deep) linguistic preprocessing. The goal of IR is to retrieve texts or texts segments that are most relevant for a given query.

Information extraction and information retrieval can be combined in various ways. IR can be used to select relevant documents for further analysis by IE. On the other hand, the structure filled by IE can also be utilized for more flexible IR (using a structured query language like SQL). Thus IE might be useful as a preparatory step for information retrieval as well as for postprocessing.

The term *text mining* (TM) is sometimes used almost synonymously to IE. It also denotes the application of data mining techniques to text with the goal of generating new knowledge by finding unknown patterns. TM in this second meaning aims farther than IE, which does not try to generate new knowledge, but only to represent facts

[1] Occasionally the term *information extraction* is defined in a broader sense and the term *fact extraction* is used to denote schema-based extraction, but we will continue to use *information extraction* in the more narrow sense since this seems to be the most common usage. Cf. Sec. 9.1 for more on the target schemas we will use.

Information Retrieval	Information Extraction	Text Mining
finding documents or text segments	filling predefined structures	discovering and filling unknown structures

Table 2.1: Applications of IR, IE, and Text Mining

explicitly expressed in a text in a more formal structure. But IE can be used as a first step in text mining, by extracting facts from the unstructured text to a database or other structured representation. In a second step, usual data mining techniques can be applied to the resulting database structure to discover interesting relationships in the data. This approach is utilized by [Nah00].

IE takes a middle position between IR (locating relevant texts) and Text Mining (generating new knowledge by finding unknown patterns). It does not try to generate new knowledge, but only to represent items of interest (facts) explicitly represented in a text in a more formal structure (cf. table 2.1).

The structure filled by IE can also be utilized for more flexible IR (using a structured query language like SQL). Thus IE might also be useful as a preparatory step for information retrieval. The extracted information can also be used in databases and ontologies for further processing—while IE extract only explicit facts, combining these extracted facts with knowledge encoded in ontologies or deductive databases allows deducing additional implicit knowledge (for example, if an extracted facts says that a sports stadium has a capacity of 50,000 spectators and another fact says that there were 50,000 spectators attending a specific event in that stadium, we can deduce that the stadium was sold out).

2.2 Overview and Classification of Approaches

The next chapter describes the architecture of a complete IE system and discusses how existing approaches fit into such a complete architecture. The remaining chapters of this part are dedicated to describe interesting approaches to IE. The focus is adaptive systems that can be customized for new domains by training or the use of external knowledge sources. Handcrafted systems that can only be adapted by elaborate rewriting are not considered. According to the observed origins and requirements of the examined IE techniques, a classification of different types of adaptive IE systems is established. The classification is significantly based on the essential methods and resources used for extraction such as learning techniques and models and central features. Therefore the approaches that belong to different classes are not necessarily completely orthogonal to each other since some techniques and features are not exclusive to an approach (e.g. rule-based approaches may use some statistical techniques for solving some subtasks in the extraction algorithm).

Table 2.2 lists the regarded systems and the approaches they represent.

Three main classes can be distinguished: rule learning, knowledge-based and statistical approaches. In Chapters 4 and 5 the approaches are presented according to the classification so as related subclasses are discussed in the common context. To make

Approach	System(s)	Section
Statistical Approaches		
Prob. Semantic Parsing	SIFT [Mil98, Mil00]	4.1 (p. 29)
Hidden Markov Models	Active HMMs [Sch01, Sch02] Stoch. Optimization [Fre99, Fre00b] (C)HHMMs [Sko03]	4.2 (p. 30)
Conditional Markov Models & Random Fields	MEMMs [McC00] CRF [Laf01, McC03b]	4.3 (p. 31)
Token Classification	MaxEnt [Chi02] MBL [Zav03] ELIE [Fin04a, Fin04b]	4.4 (p. 32)
Fragment Classification & Bayesian Networks	SNoW-IE [Rot01, Rot02] BIEN [Pes03]	4.5 (p. 33)
Rule Learners		
Covering Algorithms	Crystal [Sod95, Sod97a] Whisk [Sod99] $(LP)^2$ [Cir01, Cir02]	5.1 (p. 35)
Relational	Rapier [Cal98a, Cal03] SRV [Fre98b]	5.2 (p. 37)
Wrapper Induction	Stalker [Mus01, Mus03] BWI [Fre00a]	5.3 (p. 38)
Hybrid (Decision Trees)	IE^2 [Aon98]	5.4 (p. 40)
Knowledge-based Approaches		
Thesaurus-based	TIMES [Bag97, Cha99]	5.5 (p. 40)

Table 2.2: Overview of the Selected Approaches and Systems

the analysis of different approaches more systematic and establish a common base for their comparison and correlation we consider several qualitative criteria.

Used methods and algorithms: We focus on how relevant content is identified in texts and what techniques are used to match it to the target schema. Learning capabilities, learning models, the amount and role of human interaction are analyzed to infer advantages and weaknesses of the approach. These aspects form the basis of our classification and are mainly discussed in Chapters 4 and5 where the different types of approaches are presented.

Input and output features: Input characteristics involve the prerequisites that the processed texts should fulfill and requirements on used resources. These characteristics affect the domains where approaches can be employed (application range) and how easily they can be adapted to new domains and resources (adaptability). It is examined how much preparatory work and linguistic preprocessing is necessary, whether morphological and syntactic analysis is presupposed etc. Another important factor is whether the approaches rely on external resources such as semantic resources (e.g. thesauri or ontologies).

Output features define the accomplished tasks—which IE tasks have been solved completely or partially. We consider whether single attributes of target schema can be identified in text (single slot extraction) or complex facts consisting of several attributes (template unification) can be found. A résumé over these characteristics is given in Chap. 6.

The work presented in this part is based on previous joint work with Peter Siniakov, published as [Sie05b]. The text of this section as well as parts of the following chapter discussing system architectures and of the comparison presented in Chap. 6 have been co-authored by both of us. The approaches presented in Chapters 4 and 5 have been described by the author of this thesis.

In this work, we will keep the focus on general approaches instead of describing specific systems in detail; more detailed and specific descriptions can be found in the journal paper ([Sie05b]).