Corporate Smart Content

Designs and Prototypes

Report II on the sub-project Smart Content Enrichment

Technical Report TR-B-15-02

Adrian Paschke, Ralph Schäfermeier, Kia Teymourian and
Alexandru Todor and Ahmad Hasan

Freie Universität Berlin
Department of Mathematics and Computer Science
Corporate Semantic Web

March 16, 2015

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Abstract

In this technical report, we present the results of the second milestone phase of the Corporate Smart Content sub-project "Smart Content Enrichment". We present concepts and first prototypes of tools and APIs addressing the problems described in our last report which were identified in the fields concerning the three working packages defined in the sub-project, which are aspect-oriented ontology development, complex entity recognition, and semantic event pattern mining. We describe solution designs and demo implementations.

Contents

Chapter 1

Introduction

This is the second report on the research efforts of the **“Smart Content Enrichment” (SCE)** sub-project of the InnoProfile-Transfer **“Corporate Smart Content (CSC)**[1] project funded by the German Federal Ministry of Education and Research (BMBF) and the BMBF Innovation Initiative for the New German Länder-Entrepreneurial Regions. This report covers the period after the publication of the first report [16].

Based on the requirements specified in the first report, this report describes concepts and first prototypes of tools and APIs addressing these problems.

The rest of this report is structured along the three main topics of the Smart Content Enrichment project: Aspect-Oriented Ontology Development, Complex Entity Recognition and Knowledge-Based Mining of Complex Event Patterns.

For the working package Aspect-Oriented Ontology Development, we present a prototypical aspect-oriented extension to an ontology API that permits transparent read and write access to aspect-oriented ontologies (e.g. ontology modules defined by aspect) using aspect-oriented programming techniques. We further present a formal description of aspects in OWL ontologies and RDF graphs using a framework of aspect ontologies and a named graph approach, respectively.

For the working package Complex Entity Recognition we present an approach based on relations between simple entities and experiments on optimization by relevance assessment of relations.

For the working package Complex Event Pattern Mining, we present a probability-based approach for mining of event detection patterns based on event frequency. Second, we present concepts for knowledge-based mining of event detection patterns based on external background knowledge.

[1] `http://corporate-smart-content.de/`

Chapter 2

Aspect-Oriented Ontology Development

2.1 Introduction

The modularization of ontologies may serve different goals, such as the improvement of reasoning and query result retrieval performance, scalability for ontology evolution and maintenance, complexity management, amelioration of understandability, reuse, context-awareness, and personalization [15]. As a consequence, a significant number of ontology modularization approaches exist, each of them addressing one or several of the above-mentioned goals.

As the goals differ in different ontology modularization techniques, so do the criteria that they employ in order to determine which parts of an ontology should be contained in a particular module and which ones should not. There exist approaches that focus on topical decomposition, gathering topically related concepts in the same module while ensuring that the resulting modules remain semantically consistent. Some techniques adopt a graph-based approach, interpreting domains and ranges of relations as graph vertices and following these vertices in order to gather related concepts. Other approaches exploit the networked structure of ontologies and employ metrics from the domain of social network analysis in order to generate clusters of related concepts. For a comprehensive study of the related work in the field, cf. [16].

What the majority of the existing approaches have in common is that they are algorithmic, and the criteria used for selecting an ontology module are determined by the respective algorithm. Some of the existing modularization techniques operate completely autonomous, requiring no user interaction, while others are parameterizable to a certain extent, permitting some degree of adaptation of the modularization criteria to the user's needs. However, the parameters often reflect the internal operational mode of the modularization algorithm rather than requirements concerning the expected outcome of the modularization process from a user's point of view.

Furthermore, relying on a particular modularization approach results in a lack of flexibility. However, requirements concerning the modularization may change, even within the context of the same application. An application backed by a large and complex ontology might require a module that contains the full set of declarations of concepts and only their subclass/superclass relations for browsing the concept hierarchy. Another part of the application might require only a small, but fully axiomatized module for (topically restricted) complex queries and reasoning tasks.

The problem of modularization is not new and not unique to the field of ontology research. In software development, in particular, modularization is a key concept and has been studied for decades. One approach that has proved suitable as a solution to the problem with multiple and possibly disjoint modularization requirements is the programming paradigm of *aspect-oriented programming* (AOP).

AOP introduces the notion of *cross-cutting concerns*. Cross-cutting concerns are concerns in a software system that emerge from requirements on different levels, pertain to the entire system or a significant part of it and are thus scattered across the system, preventing meaningful code modularization [7]. Prominent cross-cutting concerns, which are often cited as an example, are authentication and logging. They are omnipresent throughout the system and cannot easily be encapsulated in a separate module (the actual functionality can, of course, be encapsulated in a separate module, but references to that module would have to be retained throughout the system).

As a solution to this problem, AOP allows for moving these necessary references out of the application code into the respective modules and provides a mechanism for reconnecting them with the application code at runtime or compile time, leading to effective and flexible modularization of the entire system. The functionality encapsulated in such a module is referred to as an *aspect*, because its functionality stems from a requirement formulated by a particular stakeholder from his or her particular point of view.

In our last report, we identified a number of commonalities shared by modular ontologies and aspect-oriented software systems, and we provided preliminary insights in the prospective applicability of the notion of aspects in order to provide a flexible mechanism for specifying ontology modules.

The new contributions of this work are as follows:

1. We demonstrate that the underlying formalisms of aspect-oriented programming are compatible with triple based data models such as RDF and DL-based ontologies and can be employed as a solution to the problem of ontology modularization and modular ontology development.

2. We provide an approach for formalizing ontology aspects on the axiomatic level by using graph-based queries and an ontological aspect metamodel which can be connected to the axioms of the ontology using annotations.

3. We propose an approach for inferring aspects on entailed axioms using explanations.

4. We present an implementation that demonstrates the applicability of our approach to OWL2 ontologies, using SPARQL as a query language.

5. We demonstrate that our approach can, in some situations, serve as a means for metamodeling and prevent expensive refactoring in ontology reuse scenarios.

Parts of the results of this work have been published in [21] and [22].

2.2 A Formalism for the Specification of Aspects in Ontologies

We derive a formalism for Aspect-Oriented Ontology Development (AOOD) from the basic principles of Aspect-Oriented Software Development (AOSD) on the one hand, and from the list of requirements we have compiled in our last report [16] on the other hand (see Table 2.1).

Aspects in software are self-contained modules that comprise a well-defined fragment of a system's functionality and instructions on how to combine this functionality with the main system, the former being referred to as *advice* and the latter being referred to as *pointcuts*.

Advice is regular software code, in the majority of cases written in the same programming language as the main system. A pointcut is a collection of so called *join points*, points in the code of the main system where the advice of the aspect is supposed to be executed. A pointcut can be an exhaustive list of join points or an abstract description thereof by the means of *quantification* (cf. [16, page 12]).

The functionality in the advice normally reflects a cross-cutting concern, i.e. the implementation of a requirement that cross-cuts with other requirements of the system.

As described in [16], cross-cutting concerns also appear in ontology development, and it is desirable to encapsulate such concerns in self-contained modules and recombine them when needed. The following sections describe how we derive a formalism for aspects in ontologies from the above concepts.

2.2.1 An Aspect Vocabulary for Ontologies

We start by defining a vocabulary that is suitable for a sound and complete description of aspects in ontologies on an abstract level. Later on, we will demonstrate two concrete applications of the vocabulary in RDF and OWL 2, respectively.

Software aspects apply to lines of code where the execution flow is diverted due to a call to a method, function or routine (or whichever callable units the paradigm of the programming language at hand defines), i.e., each call is a candidate for a join point, and the system can be changed by advising the join points.

In the case of ontologies, the application of an aspect would mean the modification of factual (Tbox or Abox) knowledge. Hence, elements that can be used as join points in ontologies must be expressions that represent canonical units of factual knowledge. What these simple expressions exactly are is defined by the knowledge representation model of each specific language. In the case of RDF they are triples, while in the case of OWL, they are axioms.

Once the simple expressions of a language have been identified, we can map the relevant concepts from the AOSD domain to ontologies and come up with an abstract definition of ontology aspects:

Pointcut: A pointcut in an ontology is the set of simple expressions that will be affected by the facts that are contained in the aspect (and stem from a cross-cutting requirement).

Advice: An advice in an ontology is a set of facts that will be applied to the facts in the pointcut and, depending on their nature, extend or restrict the facts in the pointcut. A language aspect might, for example, add language specific information (a translation) to the facts in the pointcut, while a temporal aspect might impose a temporal restriction on them (with the consequence that they are only valid during a specified period in time).

Aspect: An aspect in an ontology is a compound entity consisting of a pointcut and an advice.

2.2.2 Embedding Aspects in RDF Graphs

The Resource Description Framework (RDF)[1] is a data interchange model for data on the semantic web. It has a simple triple based semantics to describe resources on the web, (typed) links between resources, and data attributes on resources.

An addition to RDF are Named Graphs[2], a formalism that allows to label each RDF triple with a URI which in turn establishes a context within which the triple exists. A triple can be part of multiple graphs at the same time.

We use the Named Graphs model in order to convey our notion of aspects and their constituents pointcut and advice. We conceptualize these terms and their relations in an aspect RDF ontology. The ontology is depicted in Figure 2.1.

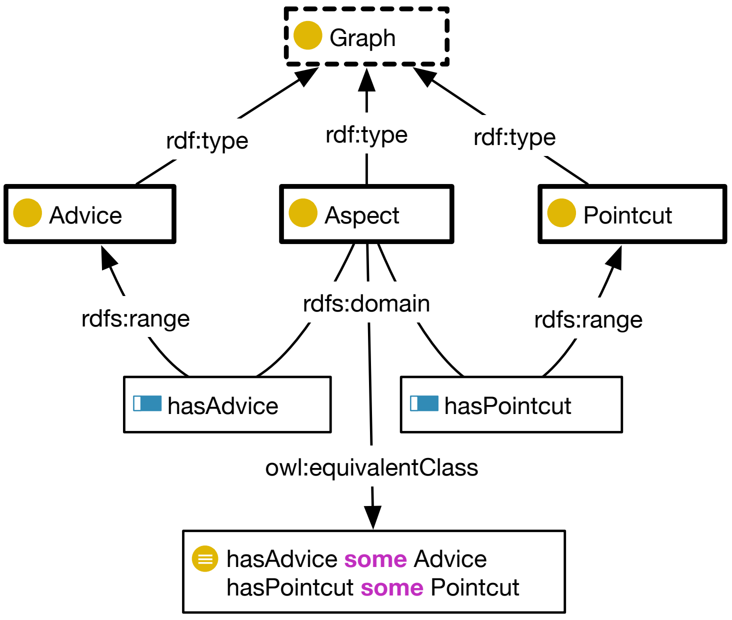


Figure 2.1: The aspect ontology for RDF graphs. Note that *Aspect*, *Pointcut* and *Advice* subclass *Graph* from the external RDF Named Graphs Vocabulary.

The ontology uses RDFS and OWL constructs in order to express more complex relations. However, the informative statements hold under an RDF

[1] http://www.w3.org/RDF/
[2] http://www.w3.org/2004/03/trix/

interpretation and are sufficient to convey the intended formal grounding of the approach.

The central resource in our model is *Aspect*. The Aspect concept has meronomic relationships to each of the concepts *Pointcut* and *Advice*. All three concepts are of `rdf:type` *Graph* from the Named Graph Vocabulary.

Each of the two concepts are intended to be instantiated in the form of named graphs, respectively. Each graph contains a set of RDF triples. The triples gathered in the pointcut graph are the target of the aspect's advice, which is the set of triples gathered in the corresponding Advice graph.

An aspect may contain arbitrarily many pointcuts. We define the semantics for cases where more than one pointcut is associated with the same aspect as follows: All pointcuts defined for the same aspects are affected by the aspect's advice in the same way.

Overlapping pointcuts, i.e., pointcuts with non-empty intersections, are allowed. In this case, the advice affects the duplicated triples as if they were only present once (note that the RDF specification allows for the arbitrarily often repetition of the same triple in an RDF graph).

In the same vein, we allow for multiple and overlapping advices in the same aspect. The intended semantics behind this is that every advice in the same aspect is applied to every triple in every pointcut of that aspect.

Examples

Listings 2.1 and 2.2 provide two examples of aspects in the DBpedia[3] dataset.

The first example demonstrates the use of an external vocabulary in an advice, in this case the W3C provenance vocabulary[4] in order to convey provenance information to the (DBpedia) fact that US President Barack Obama is married to his wife Michelle.

The second example used a custom vocabulary for expressing time constraints. The same fact is augmented with meta information about the time period in which the fact is valid.

2.2.3 Embedding Aspects in OWL Ontologies

In [26], Steimann argues that quantification in AOP involves higher-order quantification as the quantified variables range over functions. However, in AOP, the recombination of modules takes place in an additional compilation phase of the application, rather than at runtime. Therefore, the discourse domain is the set of method signatures in the system, of which at this point no instantiations exist. Since the set of signatures is finite, the second-order nature of the quantification can in fact be reduced to second-order quantification under a Henkin semantics, which is computationally equivalent to first-order quantification.

Aspects and Ontology Modules

Based on this insight, we propose our approach for ontology module selection by using higher-order statements about ontology parts on the level of ontology axioms.

[3] http://de.dbpedia.org/
[4] http://www.w3.org/ns/prov/

```
@prefix : <http://www.example.org/aspect123#> .
@prefix this: <http://www.example.org/aspect123> .
@prefix aspect: <http://www.corporate-semantic-web.de/
    ontologies/aspect/rdf#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix dbpedia_prop: <http://dbpedia.org/property/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/>.

this: {
    this: a aspect:Aspect .
        aspect:hasPointcut :pointcut .
        aspect:hasAdvice :advice .
}

:pointcut {
    dbpedia:Barack_Obama dbpedia_prop:spouse dbpedia:
        Michelle_Obama.
}

:advice {
    :pointcut prov:generatedAtTime "2004-07-29T10:38:00Z"
        ^^xsd:dateTime .
    :pointcut prov:entity <http://de.wikipedia.org/wiki/
        Barack_Obama>
    :pointcut prov:wasAttributedTo _:a .
    _:a foaf:homepage <http://de.wikipedia.org/wiki/user:
        wikipediauser123>
}
```

Listing 2.1: An example of an aspect using a third-party vocabulary. Provenance information is specified using the W3C provenance vocabulary and added to facts in the DBpedia dataset as an aspect.

```
@prefix : <http://www.example.org/aspect987#> .
@prefix this: <http://www.example.org/aspect986> .
@prefix aspect: <http://www.corporate-semantic-web.de/
   ontologies/aspect/rdf#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix dbpedia_prop: <http://dbpedia.org/property/> .
@prefix ex: <http://www.example.org/time#>.

this: {
    this: a aspect:Aspect .
        aspect:hasPointcut :pointcut .
        aspect:hasAdvice :advice .
}

:pointcut {
    dbpedia:Barack_Obama dbpedia_prop:spouse dbpedia:
       Michelle_Obama.
}

:advice {
    :pointcut ex:start "1992-10-18"^^xsd:date .
}
```
Listing 2.2: An example of a temporal aspect using own vocabulary about time instances.

We define an aspect-oriented ontology module as follows:

Definition 1 (aspect-oriented ontology module) *Given an ontology $\mathcal{O}$ that consists of a finite set of axioms* $\mathsf{Ax}_{\mathcal{O}}$*, an aspect ontology* $\mathcal{O}_A$ *with the set* $\mathcal{A} \subseteq \mathsf{Sig}(\mathcal{O}_{\mathcal{A}})$ *of all named aspect individuals in* $\mathcal{O}_{\mathcal{A}}$*, an aspect individual* $a \in \mathcal{A}$ *and a predicate* $\mathsf{hasAspect}$*. Then* $\mathcal{O}_a \subseteq \mathcal{O}$*, consisting of the set of axioms* $\mathsf{Ax}_{\mathcal{O}_a} := \{\mathsf{ax} \in \mathsf{Ax}_{\mathcal{O}} \mid \mathsf{hasAspect}(\mathsf{ax}, a)\}$ *is an ontology module defined by the aspect* a*.*

We make use of an external aspect ontology $\mathcal{O}_A$, which provides us with the necessary vocabulary for making statements that assign aspects to axioms of $\mathcal{O}$, using the $\mathsf{hasAspect}$ predicate. As can be seen, $\mathsf{hasAspect}$ is a higher-order predicate since its domain consists of ontological axioms. This imposes a strong restriction on the applicability of our approach, since real-world ontology languages such as OWL, which have a rigidly defined semantics based on Description Logics, do not allow for higher-order statements of this kind.

In order to circumvent this restriction, our approach involves interpreting the ontology under a less rigid semantics during the selection process. In the case of an OWL ontology, the selection process interprets the ontology under the RDF semantics. In this way, ontology axioms can be expressed in terms of sets of triples. Similar to the selection process in AOP, no instantiation or interpretation takes place at this stage, but for the purpose of module selection, this is obviously not necessary. Since an ontology is a finite set of axioms, the quantification over the set of axioms can also be interpreted as second-order quantification under a Henkin semantics, as long as the axioms are not interpreted.

By this means, every axiom of an ontology can be related to an arbitrary number of aspects which can then be used as a selection criterion for module extraction.

Module Extraction

Once the relations between ontology axioms and aspect individuals are established, modules can be extracted accordingly. In the simplest case, individuals serve as a named identifiers for an aspect. However, more complex scenarios are feasible. For example, different aspects could be defined by instances of a time interval class. It is then possible to define a further aspect the description of which consists of all time individuals between a given start and end time (see Figure 2.5 for an example).

This way, it is possible to define arbitrary aspects, and hence ontology modules, using the features of the ontology language at hand. This provides a high degree of flexibility. Aspect-oriented modules can, for example, be used for metamodeling, circumventing the barriers imposed by the rigid semantics of ontology languages necessary in order to keep reasoning decidable. As in the above example, aspects can define a set of facts which are valid under certain conditions (in this case a time interval), extract only those facts and then perform DL-based reasoning on this module.

Since aspects identifiers are ontological entities, it is possible to define aspects in terms of other aspects. For example, an ontology developer could use aspects in order to extract an ontology module that is relevant in a particular project (using a project affiliation aspect) and containing only facts that are valid during a specific period of time (using a time interval aspect). The combined aspect can be defined as the intersection of the two other aspects, and the resulting module would be the intersection of both sets of axioms, defined by the two aspects.

Aspects and Entailment

If an ontology contains asserted axioms that are intended to hold only under particular aspects, then the question arises under which aspects the logical consequences of these axioms hold. In order to determine that, we can employ justifications for entailments as defined by Horridge et al. in [12].

Definition 2 (Justification) *For an ontology $\mathcal{O}$ and an entailment η where $\mathcal{O} \models \eta$, a set of axioms $\mathcal{J}$ is a justification for η in $\mathcal{O}$ if $\mathcal{J} \subseteq \mathcal{O}$, $\mathcal{J} \models \eta$ and if $\mathcal{J}' \subsetneq \mathcal{J}$ then $\mathcal{J}' \not\models \eta$.*

As also noted in [12], there may be multiple, potentially overlapping, justifications for a given entailment. It is obvious that if there exists an explanation for an entailment that only contains axioms without any associated aspects, then the entailment is also free of aspects.

In cases where multiple explanations for an entailment exist, one of which contains an axiom with an aspect and another one contains an axiom with a different aspect, then the entailment has both aspects. Algorithm 1 demonstrates the carrying over of aspects to inferred axioms.

Algorithm 1 Carrying over aspects from asserted to inferred axioms using justifications.

```
Input: An ontology $\mathcal{O}$, an aspect ontology $\mathcal{O}_A$ an entailment $\eta$ with $\mathcal{O} \models \eta$ and
    the set $\{\mathcal{J}_1, \ldots, \mathcal{J}_n\}$ of all justifications for $\eta$.
 1: $\mathrm{A} \leftarrow \emptyset$
 2: for all $\mathcal{J} \in \{\mathcal{J}_1, \ldots, \mathcal{J}_n\}$ do
 3:     $\mathrm{flag} \leftarrow false$
 4:     for all $\mathsf{ex} \in \mathcal{J}$ do
 5:         for all $\mathsf{a} \in \mathcal{O}_A, \mathsf{asp}(\mathsf{ex}\ \mathsf{a}), \mathsf{a} \leq \mathsf{hasAspect}$ do
 6:             $\mathrm{A}.add(\mathsf{asp}(\eta\ \mathsf{a}))$
 7:             $\mathrm{flag} \leftarrow true$
 8:         end for
 9:         if $\mathrm{flag} == false$ then return
10:         end if
11:     end for
12: end for
13: $\mathcal{O} \leftarrow \mathcal{O} \cup \mathrm{A}$
```

The set A and the boolean flag are needed for determining whether there exists a justification $\mathcal{J}$ containing only axioms that are free of aspects. For each $\mathcal{J} \in \{\mathcal{J}_1, \ldots, \mathcal{J}_n\}$, where any of the axioms ex is assigned to an aspect a using the property asp, which is a sub-property of a property $\mathsf{hasAspect}$, the aspect assigning axiom is added to a set A replacing ex with η (line 6) and the flag is set to *true* (line 7). Once the inner loop terminates, and the flag still has the value *false*, then the inferred axiom can safely be considered aspect free, and the algorithm can terminate prematurely (line 9). Otherwise, all aspect assigning axioms are collected in A and eventually added to $\mathcal{O}$ (line 13).

It must be noted that if aspects are used for modularization of an ontology, and it is explicitly allowed to have contradictions in the original ontology (which are meant to be resolved during the module selection stage), if the above algorithm is applied to the original ontology, it needs to be assured that conflicting information is not carried over to the selected modules later.

Given, e.g., an ontology consisting of the following axioms:
capitalOf(Rio_De_Janeiro, Brazil), capitalOf(Brasília, Brazil),
hasTemporalAspect((capitalOf(Rio_De_Janeiro, Brazil)), timeInterval1),
hasTemporalAspect((capitalOf(Brasília, Brazil)), timeInterval2),
startDate(temporalAspect1, `"1889-11-15T00:00:00"^^xsd:dateTime"`),
endDate(temporalAspect1, `"1960-04-21T00:00:00"^^xsd:dateTime"`),
startDate(temporalAspect2, `"1960-04-21T00:00:00"^^xsd:dateTime"`),
InverseFunctionalObjectProperty(capitalOf).

The ontology represents the fact that the republican Brazil had different capitals throughout its history, namely Rio de Janeiro from 1889 to 1960, and Brasília from 1960 onwards. The time intervals are represented using aspects, with the consequence that the original ontology contains both the axioms capitalOf(Bonn, Germany) and capitalOf(Berlin, Germany). The additional InverseFunctionalObjectProperty(capitalOf) axiom will lead to the unwanted entailment sameAs(Rio_De_Janeiro, Brasília). In such cases, it is necessary to make the intended semantics explicit, e.g., as in the above example, by adding an axiom differentFrom(Rio_De_Janeiro, Brasília).

2.2.4 Implementation

In what follows, we demonstrate the validity of our approach described in Section 2.2.3 by presenting a concrete implementation using OWL 2 as ontology language and SPARQL as a triple-based query language for expressing quantified statements over OWL 2 axioms in the module selection step.

2.2.5 Specification of Aspects in OWL 2

As mentioned in Section 2.2.3, the OWL 2 language has a rigid semantics based on the Description Logic $\mathcal{SROIQ}$. Under this semantics, OWL 2 does, for example, not permit to make statements about axioms. The OWL 2 language, however, does permit this kind of statements in the form of annotations, with the limitation that such annotations are not interpreted under the OWL 2 semantics. This property of OWL 2 is advantageous with respect to the obliviousness property of AOP, since adding annotation axioms to an OWL 2 ontology does not alter the semantics of the ontology.

Furthermore, OWL 2 allows the object of an annotation axiom to be any kind of OWL 2 construct, which makes it very flexible with regard to the definition of complex aspects as described in Section 2.2.3. This way, it is possible to use all features of the language in order to define aspects, as well as inference on aspects.

2.2.6 Aspect Ontology

In order to permit a formal specification of aspects, we developed an ontology with an initial classification schema of aspects (see Figure 2.2e) and an annotation property hasAspect. This property can either be used in order to assign aspects to axioms, or more specialized subproperties can be added. The range of the hasAspect property is defined as the base class of all aspects, Aspect.

2.2.7 Module Selection

We have implemented our approach in the form of a plug-in[5] to the widely-known Protégé ontology editor[6].

As described in Section 2.2 , the connections between aspects and their targets may either established by using an extensional description (by providing a complete set of targets), or intensionally (by specifying the properties of the targets using an abstract query or quantification). Functionality for the first variant involves manually creating the appropriate annotation axioms (see Figure 2.4, middle).

The intensional specification by quantification is realized through a SPARQL interface. For the reasons described in Section 2.2.3, the ontology needs to be temporarily transformed to its triple-based representation in RDF. Using SPARQL CONSTRUCT queries, the subgraph of the triple based ontology representation that corresponds to the desired axioms is selected. The resulting subgraph is transformed back into an OWL 2 model, which then contains the desired set of axioms. For each axiom in the result set, an aspect annotation axiom is then added to the ontology, with the axiom as annotation subject and the specified aspect as the annotation object (see Figure 2.4, to the left, and Figure 2.5a).

2.2.8 Module Extraction

The extraction process involves providing a set of aspects and results in the extraction of the corresponding ontology modules (see Figure 2.4, to the right). The provided aspects can be the result of manual selection or of a query, e.g., by using the DL-query facility provided by Protégé.

2.2.9 Aspects and Entailment

The checking for aspects of entailed axioms has been implemented as described in Section 2.2.3. Since OWL 2 permits the creation of subproperties of annotation properties, and the aspect ontology encourages users to do so, it is necessary to transitively check for all subproperties of the hasAspect property. Inferred axioms with aspect annotations are presented in a special Protégé view (see Figure 2.5b).

2.2.10 The Ontology Reuse Case

To demonstrate the applicability and usefulness of our approach to aspect-oriented ontology modules using annotations, we depict a scenario where an external ontology is imported and extended by additional knowledge. The case involves modeling knowledge about territories and their status wrt. to international recognition as a sovereign state. As an example, we picked the

[5] http://page.mi.fu-berlin.de/schaef/aspectowl/de.fuberlin.csw.aspectowl.protege.jar
[6] http://protege.stanford.edu/

Republic of Kosovo, which, at the time this writing, has been recognized by 108 UN member countries, leaving its status disputed.

The geopolitical ontology[7], published by the Food and Agriculture Organization of the United Nations (FAO), contains a conceptualization of regions of the world as well as their status concerning international recognition.

In our scenario, we want to use the geopolitical ontology and extend it to include facts about the date since when particular countries recognize a territory as a sovereign state, i.e., we want to include a cross-cutting temporal aspect. We intend to use the W3C time ontology[8] for this purpose.

International recognition is asserted using an object property recognizedBy. The fact that the United Kingdom has recognized the Republic of Kosovo is, e.g., asserted by the predicate recognizedBy(Kosovo, United_Kingdom). Adding temporal information to that predicate is not directly possible in OWL 2, since OWL only supports binary predicates, while adding temporal information would require a ternary predicate. For these cases, the W3C proposes a refactoring pattern [9]. The refactoring required in the context of our scenario is depicted in Figure 2.2d. It involves the creation of a new class and an instance of that class, which will now represent the relation. An additional object property needs to be created in order to connect the three participants in the ternary relation via the instance. Using this pattern leads to a number of undesirable effects. For example, the range of the recognizedBy changes from Country to the synthetic Recognition class. Class restrictions relying on this property become invalid and need to be repaired. Finally, if an updated version of the FAO ontology became available, the refactoring would need to be repeated.

Including the temporal aspect using an aspect annotation is possible without any modifications (see Figure 2.2c). Using our aspect-oriented approach for module selection, it is now possible to perform reasoning or querying on the temporal part of the ontology, e.g., for selecting all countries that have recognized the Kosovo before a specific date, and then extract these as a module to perform further reasoning with.

2.3 A Java API for Aspect-Oriented Access to Ontologies

The previous sections covered our formal approach to the description of aspects in ontologies and prototypical tool support for ontology developers in the form of Protégé plugins. In addition to that, developers of semantic applications (i.e., software developers) need programmatic access to ontologies.

The most notable APIs for web ontologies are Apache Jena [10] and the OWL API [11]. With both being Java APIs, Jena's scope are RDF graphs while the OWL API is tailored to the OWL language[12], providing an object model for OWL entities and axioms and interfaces for DL-based reasoning and explanation support.

[7] http://www.fao.org/countryprofiles/geoinfo/en/
[8] www.w3.org/TR/owl-time/
[9] http://www.w3.org/TR/swbp-n-aryRelations/#pattern1
[10] http://jena.apache.org/
[11] http://owlapi.sourceforge.net/
[12] At the time of the writing of this report, the target language of the OWL API was OWL 2.

Listing 2.3: An AspectJ pointcut that captures all calls to the OWL API's model getters that return OWLAxioms.

```
pointcut accessToAxioms() :
  call(public java.util.Set<OWLAxiom>
    org.semanticweb.owlapi.model.*.*(..));
```

In what follows, we describe an extension of the OWL API by programmatic access to OWL aspects as defined in the previous sections.

2.3.1 OWL Aspects Implemented as Generic AspectJ Aspects Supplementing the OWL API

According to feedback from our industrial consortium partners we could gather the following requirements for programmatic access to OWL aspects:

- The approach should be usable, i.e. comprehendible and not overly complicated.
- The approach should be backwards-compatible with existing systems.

While the first requirement (or, more precisely, its measurable manifestation in the real world) is arguably of a somewhat subjective nature, it can still be argued that simplicity is a desired property in the design of the approach.

Particularly with regard to the second requirement an aspect-oriented approach is a natural candidate for a viable solution since uninvasiveness is one of the goals of AOP.

We designed an aspect-oriented extension of the OWL API using the Java aspect language AspectJ[13]. We use Java aspects in order to intercept read/write access to the OWL API's Java object model of a loaded OWL ontology and advice the calls by code responsible for extracting ontology modules affected by an OWL aspect.

With regard to requirement 2 we decided to use Java annotations as a means to convey the selected aspects from the Java side.

Listing 2.3 shows one of the pointcuts in the OWL API necessary to intercept read access to an ontology's axioms.

Listing 2.4 demonstrates the advice for this pointcut. Note that the aspect URIs are passed from outside (in the Java code) in the form of annotations of a custom annotation type `OWLAspect` that accepts as parameter an array of URIs (see Listing 2.5).

Listing 2.6 shows an example call with the URIs of the two example aspects from Section 2.2.2.

2.4 Outlook

The project phase covered by this report comprised the design and prototypical implementation of the approach to aspect-orientation in ontologies and the

[13] https://eclipse.org/aspectj/

Listing 2.4: The corresponding AspectJ advice that extracts the modules related to the aspects specified with the @OWLAspect Java annotation.

```
Set<OWLAxiom> around(): accessToAxioms() {
 Set<OWLAxiom> axioms = proceed();
 MethodSignature callerSig = (MethodSignature)
     thisEnclosingJoinPointStaticPart.getSignature();
 OWLAspect[] aspectAnnotations = callerSig.getMethod().
     getAnnotationsByType(OWLAspect.class);
 for(OWLAspect aspectAnnotation : aspectAnnotations) {
     ... // return only axioms associated with given
         aspects
 }
}
```

Listing 2.5: The @OWLAspect Java annotation type. Calls to OWL API methods in an application developer's code annotated with this annotation type will be intercepted and the OWL API objects will be modified according to the aspect URIs passed as parameters to the annotation.

```
@Target(value={METHOD, TYPE})
@Retention(RetentionPolicy.RUNTIME)
public @interface OWLAspect {
         String[] value();
}
```

Listing 2.6: Example client code using the OWL API. Note that aspect references are added transparently and uninvasively as method annotations. Client code itself does not need to be changed.

```
@OWLAspect({"http://www.fu-berlin.de/csw/ontologies/aood/
     ontologies/aspect123", "http://www.fu-berlin.de/csw/
     ontologies/aood/ontologies/myAspect456"})
public void doSomething () {
 Set<OWLAxiom> allAxioms = myOntology.getAxioms();
 ...
}
```

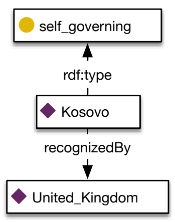
design of an aspect-oriented extension to an API for ontologies. According to our working plan, the next phase will consist of finishing a prototypical implementation of the API extension (which is work in progress at the time of writing this report). While the implementation is expected to be straightforward, special attention has to be paid when constructing the Java pointcuts to the OWL API that all relevant calls are captured.

The next step after finishing the implementation will be an evaluation with our industrial partners with regard to the requirements and the integration into our general architecture and demo implementations that fulfill our vision of the Corporate Smart Content process chain.

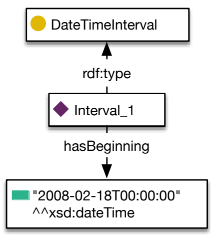
Functional Requirements		
#	**Req. ID**	**Description**
1	decomposition of cross-cutting concerns	The formalism should provide means for decomposing ontologies based on cross-cutting concerns by using aspects in order to attribute certain parts of the ontology to such concerns.
2	flexibility	The formalism must be flexible enough to express all kinds of (functional or non-functional) aspects.
3	self-descriptiveness	Aspect descriptions should be ontological entities (either in the same or a different ontology language as the base-ontology).
4	isolation	Aspects should not interfere with the semantics of the ontology they are added to. They should reside on a meta-level. They should **not** be first-class citizens.
5	combination	In order to embrace the problem of cross-cutting requirements, aspects must be combinable. It must be possible to assign to each axiom of an ontology an arbitrary number of aspects. During the ontology module selection stage, it must also be possible to select arbitrarily many aspects at the same time.
6	decidability	Should the approach be used in conjunction with an ontology language that is designed to only allow for decidable reasoning problems, then the aspect-oriented formalism should only introduce decidable reasoning problems as well.

Non-Functional Requirements		
#	**Req. ID**	**Description**
7	compatibility	The formalism must be compatible with existing knowledge modeling formalisms, i.e., ontology languages.
8	main module	The formalism must allow for aspect-oriented modularization independently of whether there exists a main module or not. If a main module exists, then it is identified by the fact that is not associated with any aspect.

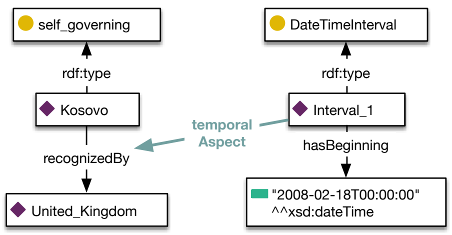
Table 2.1: Requirements for aspect-oriented ontology development as formulated in our previous report[16].



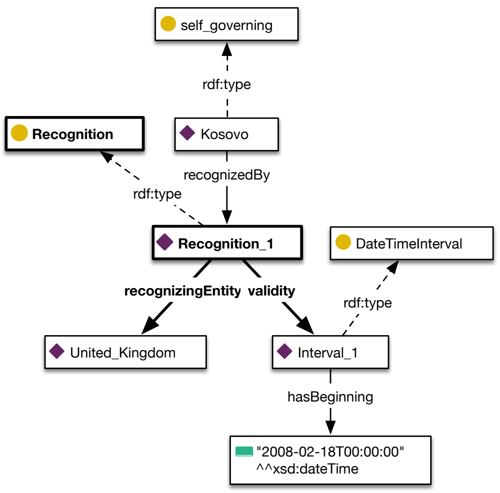
(a) Original FAO geopolitical ontology selected for reuse.



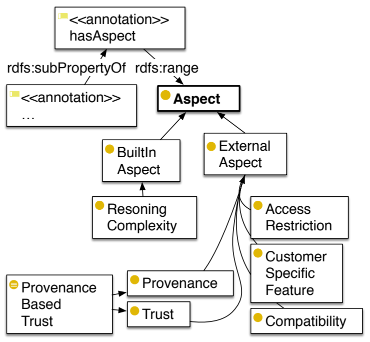
(b) Custom extension of the original ontology for representing a temporal aspect.



(c) Modeling of the extension as an aspect by using an axiom annotation.



(d) Necessary refactoring of the reused ontology in order to allow for the extension, following the W3C n-ary relations pattern



(e) An ontology of aspects providing an intitial classification schema for aspects and a hasAspect annotation property intended to be extended by specialized subproperties.

Figure 2.2: An example involving the integration of an external ontology for reuse using aspects and the n-ary relations pattern and the aspect ontology.

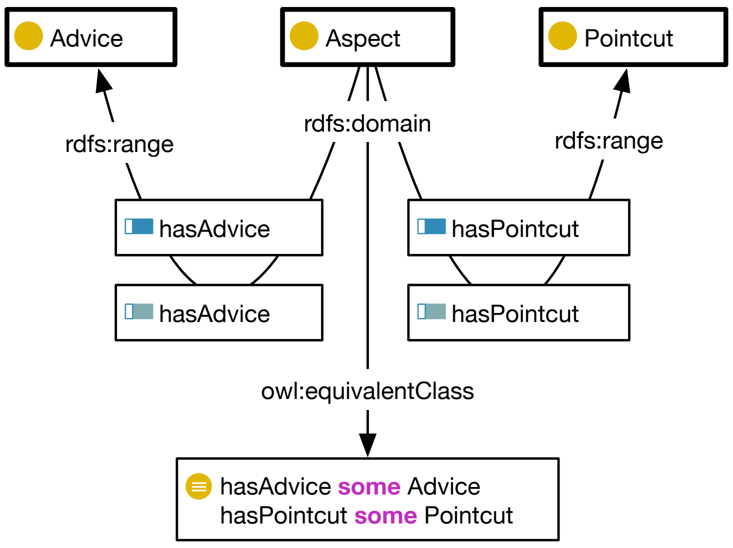


Figure 2.3: The aspect ontology for OWL 2 ontologies. Note that the common superclass *Graph* has been eliminated. Instead, the two object properties *hasAdvice* and *hasPointcut* have (punned) annotation property counterparts.

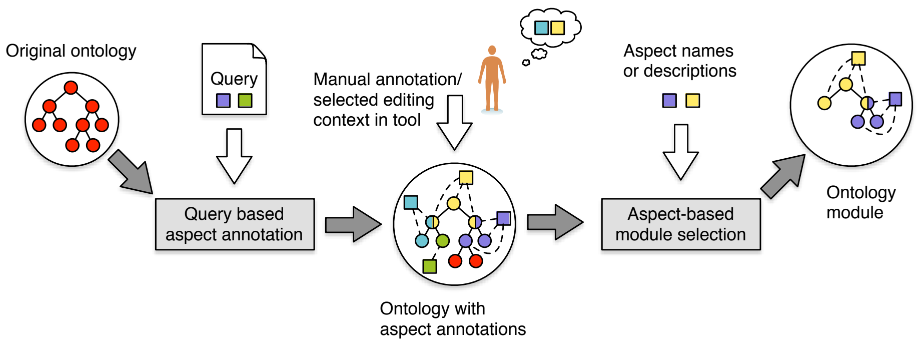
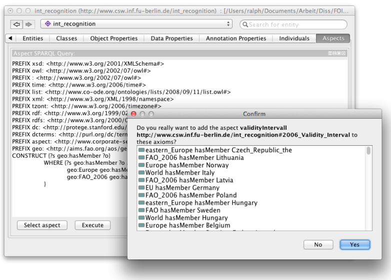


Figure 2.4: Our approach to aspect-oriented ontology modularization. Axioms of the original ontology (left) are annotated with entities from an external aspect ontology (center). Axiom selection is based on queries or is performed manually. Module extraction happens dynamically, as a particular part of the ontology is requested (right). The parameters of the request are one or several aspect names. The result is an ontology module including exactly those axioms that belong to the given aspects.



int_recognition (http://www.csw.inf.fu-berlin.de/int_recognition) : [/Users/ralph/Documents/Arbeit/Diss/FOI...

Aspect SPARQL Query:

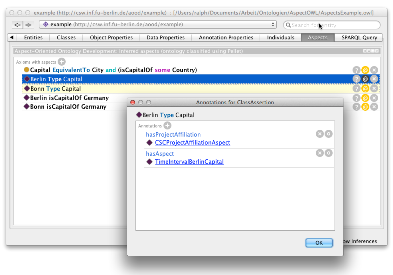
```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX : <http://www.w3.org/2002/07/owl#>
PREFIX time: <http://www.w3.org/2006/time#>
PREFIX list: <http://www.co-ode.org/ontologies/lists/2008/09/11/list.owl#>
PREFIX xml: <http://www.w3.org/XML/1998/namespace>
PREFIX tzont: <http://www.w3.org/2006/timezone#>
PREFIX rdf: <http://www.w3.org/1999/02
PREFIX rdfs: <http://www.w3.org/2000/0
PREFIX dc: <http://protege.stanford.edu/
PREFIX dcterms: <http://purl.org/dc/term
PREFIX aspect: <http://www.corporate-se
PREFIX geo: <http://aims.fao.org/aos/ge
CONSTRUCT {?s geo:hasMember ?o}
          WHERE {?s geo:hasMember ?o .
                 geo:Europe geo:hasMe
                 geo:FAO_2006 geo:ha
          }
```

Confirm

Do you really want to add the aspect **validityIntervall http://www.csw.inf.fu-berlin.de/int_recognition#2006_Validity_Interval** to these axioms?

- eastern_Europe hasMember Czech_Republic_the
- FAO_2006 hasMember Lithuania
- Europe hasMember Norway
- World hasMember Italy
- FAO_2006 hasMember Latvia
- EU hasMember Germany
- FAO_2006 hasMember Poland
- eastern_Europe hasMember Hungary
- FAO hasMember Sweden
- World hasMember Hungary
- Europe hasMember Belgium

(a) An aspect defined using a SPARQL query. The aspect definition consists of the query, an aspect annotation property, and an aspect entity. All axioms selected by the query will be annotated with the selected aspect entity, using the selected aspect annotation property.



example (http://csw.inf.fu-berlin.de/aood/example) : [/Users/ralph/Documents/Arbeit/Ontologien/AspectOWL/AspectsExample.owl]

Aspect-Oriented Ontology Development: Inferred aspects (ontology classified using Pellet)

Axioms with aspects

- Capital EquivalentTo City and (isCapitalOf some Country)
- Berlin Type Capital
- Bonn Type Capital
- Berlin isCapitalOf Germany
- Bonn isCapitalOf Germany

Annotations for ClassAssertion

Berlin Type Capital

Annotations

hasProjectAffiliation
CSCProjectAffiliationAspect

hasAspect
TimeIntervalBerlinCapital

(b) Inferred axioms with inferred aspect annotations.

Figure 2.5: Example screenshots.

Chapter 3

Complex Entity Recognition

3.1 Introduction

Most large companies today have to deal with increasingly large amounts of data, which is one of the reasons that lead to the establishment of the term "Big Data". In a "Big Data" scenario recognizing named entities in text becomes more and more important for companies, because it enables them to focus on certain aspects of their internal documents such as people, places and organizations.

In the last years a lot of named entity recognition services and libraries have become available on the market. These tools enable the identification of named entities in text and delimit them from other entities with similar names. This semantically unique identification of named entities enables new kinds of applications in the fields of *Knowledge Management*, *Information Retrieval* and *Business Intelligence*. However, one of the particularities of corporate data is the large amount of implicit entities, that are not explicitly named in the text but can be deduced trough the analysis of the relationships extracted from the text document corpora and those stored in the various corporate and external knowledge bases.

One of the problems that arises is that these implicit entities are not only expressed in a complex language form but also have a difficult structure themselves. This means that in most cases we need to analyze the contextual clues present in the text documents and the background knowledge in order to be able to recognize an implicit entity and its meaning.

3.2 Approaches for the semantic recognition, annotation and resolution of complex entities.

Examples of complex entities in the corporate context can be such entities that are indirectly referenced but are not directly mentioned by name in the text, for example "The German subsidiary of General Motors". In this context this would refer to the German automobile manufacturer Adam Opel AG, because General Motor has one one related company in Germany with the name Opel in it. Other examples can be entities(like time, place, process, role etc.), whose meaning changes based on the context. Let's take for example the role of CEO, this role will change based on the time context and can refer to different people(implicit entities) inside a company. Identifying this entity requires finding out which documents and what version of them is used in the specific context and determining what background knowledge is required in order to infer the right identity for the entity. The existence of an implicit entity in a document network(this means the entire, context relevant, corporate document corpora) also comprises the entity complexity. In our approach we focus on the different semantic relationships that form between the existing explicit entities in the documents and the semantic background knowledge, this allows us to infer as to the existence of implicit entities.

3.2.1 Concepts for complex entity recognition

The foundations of complex entities can be traced to Jerry R. Hobbs [1], one of the most prominent researchers in Computational Linguistics in his work on deep lexical semantics [11] and encoding commonsense knowledge [2].Jerry R. Hobbs defines a general theory of entities that are composed out of different things, the definition he gives also stands behind some of the concepts behind complex entities

> A composite entity is characterized by a set of components, a set of properties of these components, and a set of relations among the components and between the components and the whole. With this theory we can talk about the structure of an entity by explicating its components and their relations, and we can talk about the environment of an entity by viewing the environment as composite and having the entity among its components. [11]

In our project we are developing and testing tools and approaches for the acquisition and goal-driven processing of knowledge from corporate data. Trough the analysis of the semantic relationships between existing, explicitly named entities in the documents, new more complex implicit entities can be learned and identified.

We proposed a knowledge-based approach for the recognition of complex, composed or implicit entities. In our approach we first pre-process the document corpus by using existing named entity recognition approaches in order to identify simple, implicitly named entities. In the following steps, the recognized entities are seen as a set of entities and the relationships between them are analyzed. These relationships are extracted from an existing knowledge base as well as from the document corpus. From the analysis of the relationships between the recognized entities as well as the relations to other resources in the knowledge bases we can infer the probability of the existence of a complex entity.

One of the particularities of our approach is the ability to recognize abstract entities that are composed out of different simple entities like for example events. For example one could improve a diagnosis by recognizing and annotating a symptom description in medical texts. The relations between the individual

[1] http://www.isi.edu/~hobbs
[2] http://www.isi.edu/~hobbs/csk.html

recognized simple entities like "headaches", "fever" and "rash" in combination with background knowledge about symptoms as well as patient records let us delimit possible diseases as the triggers for these symptoms. Another example would be the description of events on the stock market trough the recognition of entities that belong or can be grouped into certain patterns, like the high volume sale of stocks and the related loss in stock value.

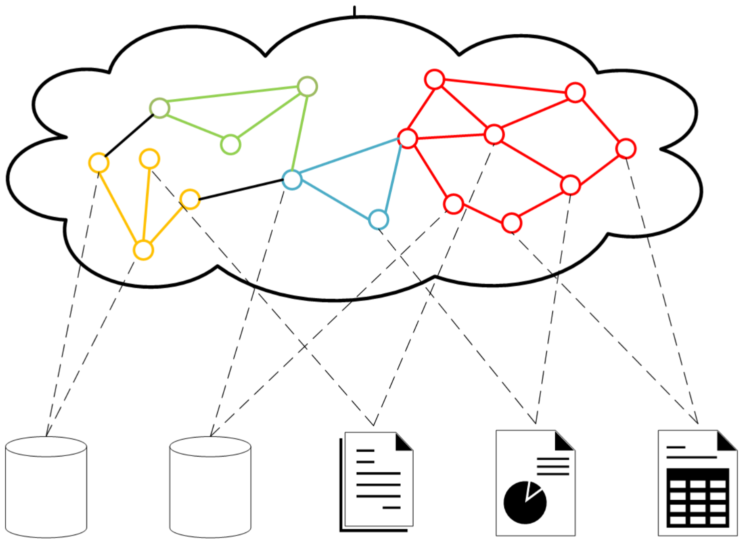


Figure 3.1: Knowledge Base formed by combining extracted information with internal and external data sources

In figure 3.1 we represent the entity cloud which results after our previous corporate information extraction process. The entity cloud contains a series of entities extracted from text documents as well as entities from existing corporate and external knowledge bases. The entities are represented by nodes in a graph, interconnected with each other by edges that represent the different properties that tie them together. By analyzing the strength and type of the connection between entities, as well as the type of the entities we can discover clusters of entities and relations that can represent a complex entity.

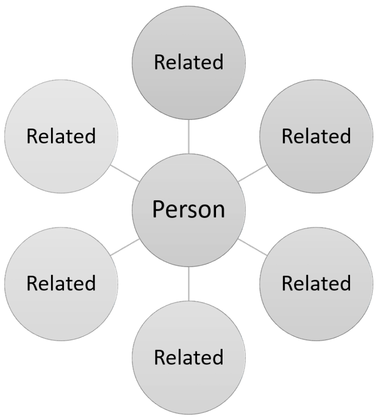


Figure 3.2: CE formed by the relations between a person and other people or concepts

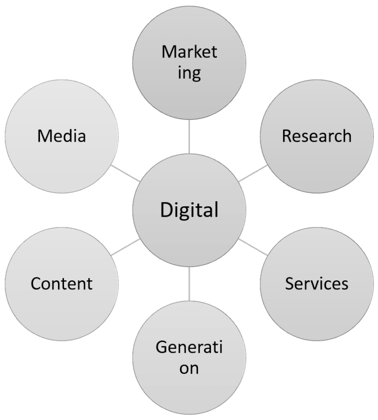


Figure 3.3: CE showing a topic formed by the relation between strongly connected concepts

Figure 3.2 shows an example of a complex person entity (CPE). A CPE can be formed by a person and other people that are strongly related to him by context, as well as the topics he is working on or other strongly related concepts. For example in a corporate context a CPE could be a project manager and his subordinates, the project they are working on as well as the topics/domain on which they are working. Mining CPEs can help companies better understand their own company structure as well as that of competitors. In figure 3.3 we see an example of a complex entity which we call a topic entity (TE). A topic entity is formed by strongly interconnected and co-occurring concepts that can be observed throughout a document collection. In order to mine such entities we combine topic modeling approaches with a graph based approach.

3.2.2 Approaches for Relation Ranking in CER

A complex entity is defined as a being formed by a set of components that are described by properties and a set of relationships between these components. In our case these components are entities in a KB (Knowledge Base) , the properties are the attributes of the entity in the KB and the set of relationships is formed by the relationships between entities in the KB.

An important part of CER is to determine the meaningful relationships between entities, and based on those to determine the other component entities of a CE (complex entity). The approach that we took was to start with a set of one or multiple centroid entities and then determine the most meaningful related entities to the centroid entity, given a specific context which in our case is a document. This approach can be reduced to an exploratory semantic search approach, where the centroid entities are the input entities for the search query. Given a set of search entities our devised algorithm [8] finds all related entities in a context and ranks them based on the strength of their relationship, in order to form an entity-relationship subgraph which in our case is a complex entity.

Related Work

Some more prominent approaches towards document search based on annotated named entities have been developed by Kiryakov et al. [13] and Castells et al. [4]. Both of these publications transfer established term based weighting metrics to entities, but at the same time they do not exploit further semantic relationships between detected entities in the same document as we do in our work.

In the approach by Aleman-Meza et al. [1] the strength of a relation between entities is based on manually assigned weights for all existing properties in the taxonomy. However, this requires domain experts need to assign one of the categories low, medium or high to each of the defined properties. While this approach of manual assignments might work well on ontologies with a small t-box, problems will arise with general purpose ontologies like Freebase, DBpedia or Wikidata containing hundreds of properties and with continuous t-box modifications. Another drawback results from the static weighting, which does not allow individual strengths of relationship between entities connected by the same property.

In [8], we address these problems by using statistical metrics that adapt almost without any manual adjustments to the data of a given knowledge base. In the next subsection we present these metrics.

Link-based ranking approaches

The analysis of link structures constitutes an important characteristic of our algorithm. Not only the entities in the search query but also the related entities should be considered when computing the relevance score. To utilize such semantic relations, it is not enough to know about the existence or absence of links. In many cases it would be useful to have a more informative statement about the strength of a connection in order to be able to assign weights to the edges in the knowledge graph. We have identified a series of ranking metrics from the literature that haven been evaluated in different semantic-based applications [20] [18] [25]. These metrics can be combined in order to provide a more flexible ranking approach that can be adapted to different scenarios.

Entity Specificity: Determining the entity specificity based on the degree of interlinkage inside ontologies is a known metric that can be compared with determining the inverse document frequency of a term inside a text corpus: relations to strongly connected entities should be ranked lower than relations to entities with a lower number of links. For instance, in an ontology that describes places people are born or live in large cities have far more links than locations with smaller populations. This metric can be computed over incoming edges: In the case of a person born in a city with a large population, the relation between this person and the city is relatively unspecific. Such a *specificity-score* is defined in [20] for the exploration of paths in a knowledge graph in order to find related entities. Let a be an entity of the knowledge base and $\{b|b \mapsto a\}$ the set of incoming links that connect other entities to a in the knowledge base. The entity specificity for a would be computed according to this formula:

$$Specificity(a) = \frac{1}{\sqrt{\|\{b|b \mapsto a\}\|}} \tag{3.1}$$

In this case we need to note that all edges/ontology properties are taken into account equally when computing the specificity score. Individual properties cannot be weighted higher or lower than others in this algorithm.

The *Clusterscore* approach by Rocha et al. [20] follows a different idea: let a and b be two entities and $a \mapsto b$ a link between a and b and $\{c\}$ a set of entities with the cardinality $|\{c\}|$:

$$Clusterscore(a, b) = \frac{|\{c|a \mapsto c \bigcap b \mapsto c\}|}{|\{c|a \mapsto c\}|} \tag{3.2}$$

The general idea with this approach is that entities are more similar to each other if they are connected via paths with many common intermediate entities.

Ranking Entity Relations

The link based ranking metrics introduced in the preceding section will be used in our ranking algorithm to weight the relatedness of entities that co-occur in documents together with the searched entities. These metrics will be combined with traditional corpora based ranking techniques applied to the annotated entities.

Utilization of Link-Based Ranking Approaches

For our work we chose to combine the specificity score with the clusterscore, since both metrics cover different and supplementary aspects. We extend the clusterscore by also taking into account the number of direct links between entities:

$$Related(a, b) = Links(a, b) \cdot Clusterscore(a, b) \tag{3.3}$$

This extension is designed to improve the ranking results when using very large KBs with a high number of properties, for smaller KBs there is no noticeable improvement.

Corpora-based Ranking Approaches

Classical term-based ranking approaches such as *TF-IDF* can also be applied for our use case, with the difference that we do not rank keyword terms but entities that have been annotated in the document corpora. The Term Frequency or in our case the *Entity Frequency (Ef)* is essential for obtaining good ranking results. The final score of the document reflects the frequency of the searched entities as wells as that of their related entities . For an entity e and a document d, where the frequency count of e is *ef*, the formula for the *Entity Frequency* is as follows:

$$Ef(e, d) = 1 + ln(ef) \tag{3.4}$$

In the next step we normalize the text documents based on the number of annotations in the documents. We use the *pivoted normalization weighting* [24] adapted for the entity frequency:

$$Ef_{norm}(e, d) = \frac{1 + ln(ef)}{(1 - s) + s\frac{dl}{avdl}} \tag{3.5}$$

The constant s adjusts the level of normalization, dl is the length of document d which is composed of a set of annotated entities, and $avdl$ is the average entity count computed from all the document in the corpora. For search queries that are composed of multiple terms and/or entities, the inverse document frequency idf helps us to determine the more specific entities from the corpora (rank rarer entities higher than more frequent ones).

$$Idf(e) = log \frac{N}{df(e)} \tag{3.6}$$

N is the number of documents in the corpora, $df(e)$ is the document frequency for an entity e (the number of documents where e has been annotated).

Computing the document ranking score

The document-score for a random search-entity e_s is composed of the normalized entity frequency 3.5 and the sum of the scores of its related entities and their product which is composed of 8 and the link-based scores 6 and 4:

$$\begin{aligned} Score(d, e_s) = Ef_{norm}(e_s, d) + k \cdot \sum_{e_V \in V} Ef_{norm}(e_s, d) \\ \cdot Related(e_s, e_v) \cdot Specificity(e_v) \end{aligned} \tag{3.7}$$

$V = \{e_v | e_s \mapsto e_v\}$ is the set of entities related with e_s. The constant k allows us to fine tune the influence of the related entities over the final score. The values of the link-based metrics are dependent on the structure of the particular KB, so the value of k must be experimentally determined for each KB. If we want to score the documents based on a set of search entities E (entities from the search query), we also need to compute the inverse document frequency:

$$Score(d, E) = \sum_{e_S \in E} Idf(e_S) \cdot Score(d, e_S) \tag{3.8}$$

This formula ensures that the score of a document increases monotonously when adding new search entities. Search entities that have common related entities, all partially influence the total score based on the strength of their relationship. This leads to a stronger influence on the score of entities that are related to multiple search entities.

Extending the Idf score of an entity to that of its related entities has the purpose of keeping the proportion between a search entity and that of its related entities.

3.2.3 Entity linking and resolution for corporate data

Entity resolution is the task of finding the appropriate real world objects or entries in a database to mentions in text. This is achieved by taking into consideration a series of textual clues around the words we are interested in. Examples of cases where entity resolution is needed are cases where we have corporate documents that mention multiple people with the same name, that address the same person in different ways, that use different names/acronyms for organizations and places. For example, we have a text that mentions a person by the name Max Mustermann. our task in this case would be to find the appropriate entry in the company employee database in order to associate the mention in text to the entry in the database. Since we have multiple people with the name Max Mustermann in the company we need to take contextual clues as to their position in the company. Since the Max Musterman in this case appears together with terms such as sales, sales strategy, CEO of sales and sales targets, we can associate this mention to Max Mussterman, CEO of sales department in the company database. Entity resolution can also be described as "the process of identifying and merging records judging to represent the same real-world entity" [2] .

In the corporate context however we deal with large quantities of data which leads to a series of new problems. As the amount of data and frequency with which it changes grows, the number of semantic relationships between entities grows in a disproportionate manner.

In the case of complex/composite entities, entity resolution poses a different challenge. Since a complex entity is formed out of different components we need to compute the probability that in the case some of the components of a CE from out KB are identified in a document, the entire CE can be identified. In order to achieve this we used a combination of string similarity and graph traversal approaches.

The approach we implementer for CE resolution is described in [5]. The main idea behind approach is the analysis of the entity-relationship graph constructed from the underlying dataset.

3.2.4 Selection and Development of an annotation approach for corporate data.

After years of work from the semantic web community, in the past year new developments have come to light that try to standardize text and multimedia annotations. From the work of the "Open Annotation" community, together with various industrial partners a new standards body has been formed called the Web Annotation Working Group [3]. This standard builds upon previous standards such as the Open Annodation Data Model and the Annotation Ontology, but has the full backing of the W3C and the Industry, thereby ensuring it will be the official and de facto standard.. Due to this developments and our familiarity with the used data model we chose to select this data model as the basis for annotations in our approach.

Companies produce a lot of internal data such as financial reports, charts, competition analysis reports, market analysis reports etc. This data can become very valuable if data mining and business intelligence methods are applied to it. One of the problems such companies face is the unstructured nature of this data with which traditional business intelligence tools can't cope with directly. By using data annotations standards such as the W3C Open Annotation initiative which we presented in our previous report[16] or the Web Annotations data model, businesses can greatly improve the day the data is stored, searched, analyzed and enriched by external tools.

One of the large hurdles to using business intelligence tools in companies is the lack of existing annotations. Producing annotations by hand can be costly

[3] http://www.w3.org/annotation/

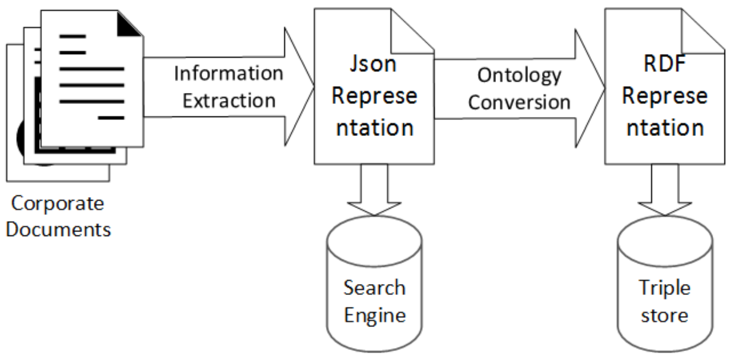


Figure 3.4: Automatic document information extraction and annotation workflow

and time consuming. Furthermore, a large amount of the internal documents in companies contain unstructured data and are stored in proprietary formats that make traditional tools unable to process them.

An expressive annotation ontology is needed in order to extract more valuable information from company documents. In many types of documents (presentations, reports, brochures etc.) the layout of the document encodes the structure and meaning of the information inside. This means that, while extracting information from documents, we not only need to extract the unstructured text and process it, but also need to take into account and preserve the layout and structure of the document at the same time.

The structural annotation of corporate documents documents can be achieved by using 2 different approaches. One is The Document Components Ontology (DoCO) [6] and the other approach is by using the Web Annotation Standard mentioned before.

The Document Components Ontology (DoCO) is an OWL 2 DL ontology that which aims to provide a structured vocabulary in order to annotate different structural document elements. The document representation in DoCO is based on the Pattern Ontology for describing structural patterns,the Discourse Elements Ontology(DEO) [4] in order to describe rhetorical elements and additional hybrid classes such as paragraph, section or list are added.

The Web Annotation Ontology is based on the Open Annotation Data Model which was presented in the previous report [16]. Since it is a more general annotation ontology it allows not only for the structural annotation of documents but also the capturing of provenance information and user annotations. However, due to it's larger scope, it is also more complex.

Figure 3.4 shows the general workflow of our document information extraction and annotation system. In a first step different types of documents are analyzed regarding their structure and information content (named entities, acronyms, topics) and then this information is stored in an application specific JSON format. The JSON format allows us to imput this information is a search engine component such as Elastic Search [5]. In a second step the JSON representation is converted into an RDF representation by employing the DoCO Ontology and the Web Annotation Ontology. The resulting RDF representation is stored in a

[4]http://www.essepuntato.it/lode/http://purl.org/spar/deo
[5] https://www.elastic.co

triplestore and enable further semantically enriched functionality.

3.2.5 CER System Architecture and Prototype

The CER system architecture is custom designed for some of the unique requirements that many corporations have. Unlike normal news sites or Wikipedia, making sense of corporate documents requires complex semantic understanding of who wrote a document, who collaborated with the author of the document, in what role, department or workflow step the document was created. In order to capture this background information we can use annotations as we discussed in the previous section. However, in many cases such annotations can only be added after the fact in a manual process that is to laborious in order to be performed.

In order to address these requirements we implemented a series of components that address the recognition of more complex entities such as collaboration networks between authors, document structure, corporate specific acronym detection and extraction, topic detection, author-topic networks etc.

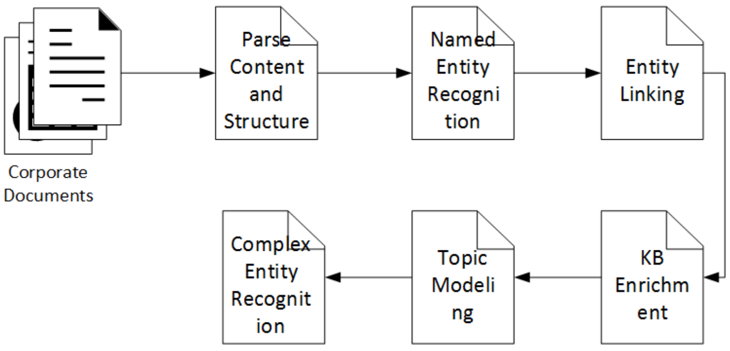

Corporate Documents
Parse Content and Structure
Named Entity Recogni tion
Entity Linking
KB Enrichm ent
Topic Modeli ng
Complex Entity Recognit ion

Figure 3.5: NLP Pipeline for the CER Prototype System

Figure 3.5 shows the NLP pipeline architecture for our CER system. The first step in the pipeline includes a complex preprocessing component that performs semantic and structural analysis of different document types. The documents are converted into a uniform format and then a NER component is executed. After the NER component and additional entity linking step is performed in order to further disambiguate the recognized entities and link them to internal KB entires. The entity linking and KB enrichment steps are complementary, the newly extracted information from the documents is added to the kb thereby enriching the existing KB. The topic modeling step further enriches the KB by mining documents from the document collection and adding new topic entities to the KB. A graph based CER approach is then performed on the extended KB.

In this section we will describe the functionality of *Knowledge Manager* and show it on the example of *PDF* document format (other text formats, such as *DOC(X)* would just require a different text extractor during the first processing step).

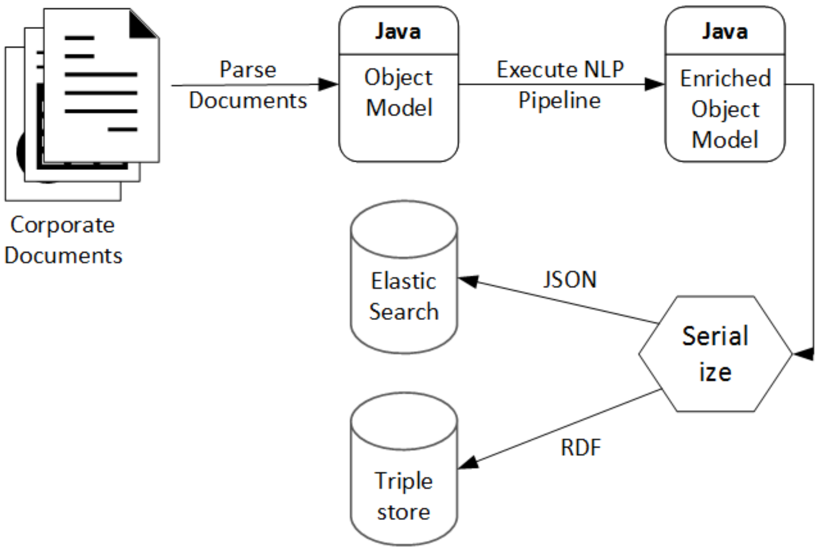


Figure 3.6: The process of extracting the data from the document and storing it in the backend.

Figure 3.2.5 shows a simplified data flow of the CER system prototype. At first a preprocessing step is executed where different types of documents in different formats are converted into a standardized format and then into a java object model for internal processing. In the next step the NLP pipeline is executed and an enriched model is generated, this model is then serialized in different formats such as JSON or RDF/XML for the different storage components of our system.

Preprocessing: In the preprocessing step needed to extract the actual text from the enterprise documents. Combining XML-formatted output of our document parser with regular expressions allowed us to build a precise model which contained all relevant information (a document which was divided into pages, which in turn consisted of paragraphs containing actual important data, such as title of the research, short summary, involved people, etc.).

Document Annotation: In the document annotation step we went through all paragraphs and passed their texts to an NER component in order to perform *Named Entity Recognition and Disambiguation*. This way, all people, places or job titles that have been mentioned within one document can be clustered across all the processed documents. Moreover, the Named Entity Disambiguation gives us the URIs to the Linked Data Knowledge Bases as long as the entity can be matched to a resource. This way we could for example easily investigate and quickly answer the following question: *What is the leading industry in the most frequently mentioned country?*

Search engine: A basic and important feature of our system is the possibility to search the documents. We do not only offer the full text search but additionally allow the user to browse the recognized entities or their types.

To avoid being bound to a certain schema and make our system more flexible in terms of data we can store, we decided to make use of Elastic Search [6] as the backend where we store our annotated documents. This service offers clear and

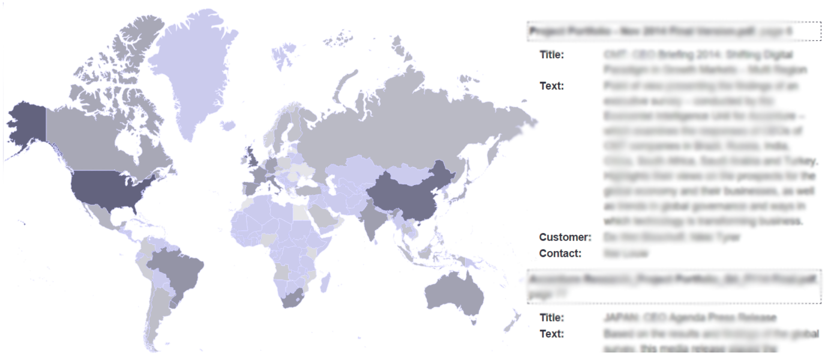
[6] https://www.elastic.co/

easy-to-use API which we use to obtain the data.

Acronyms: Our software also shows the acronyms used throughout the documents. We decided to apply this feature in two ways. The first, rather naïve way was to make use of regular expressions and extract uppercased strings of length greater than 2. On the other hand we utilized the algorithm by Schwartz and Hearst[23]. It is a computationally inexpensive method which finds pairs <short form, long form>of the abbreviation and it's matched definition. This process consists of two parts: extraction of the pair candidates from the text and identifying the correct long form from among the candidates in the sentence that surrounds the short form.

Visualizing the Knowledge: Apart from the described text-based information we also provide the visualization part.

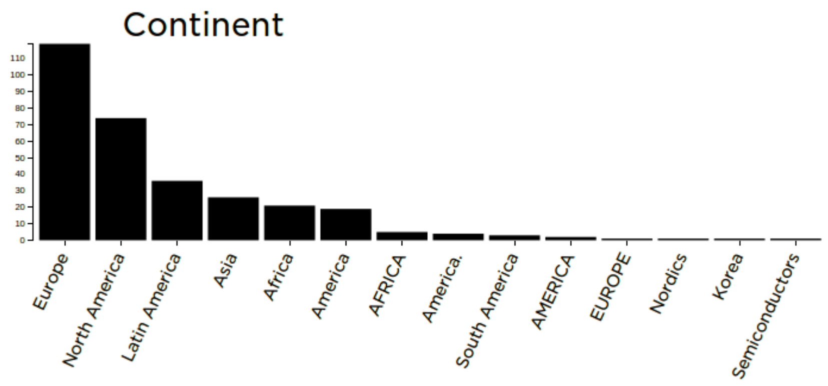
- *Map*: From all entities with type Country we created a heat map, showing how many times a country has been mentioned. Clicking on a selected country fetches the relevant paragraphs.

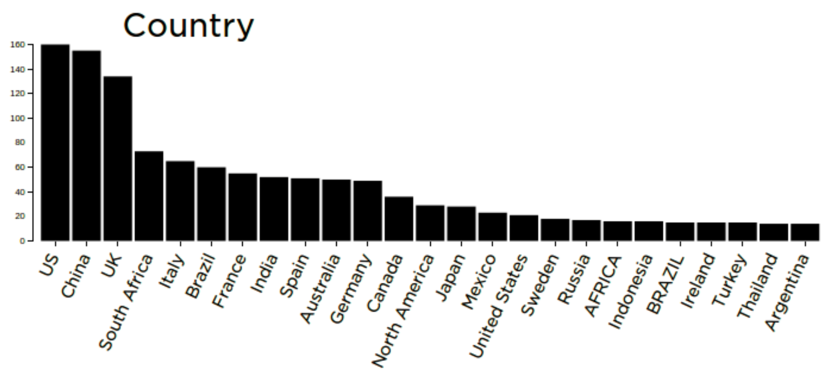


[redacted]
Title: [redacted]
Text: [redacted]
Customer: [redacted]
Contact: [redacted]
[redacted]
Title: [redacted]
Text: [redacted]

Figure 3.7:

- *Charts*: We offer a view, where we generate bar charts for selected types of entities. This gives a quick overview of the most frequently used entities within a type. Clicking a bar fetches the relevant paragraphs.





- *Timeline*: We incorporated an advanced NLP component to extract the dates from the text. Based on them and taking the document creation date as the reference point we visualize all paragraphs containing dates (or other time related expressions, e.g. *next year* or *previous month*) on the timeline.

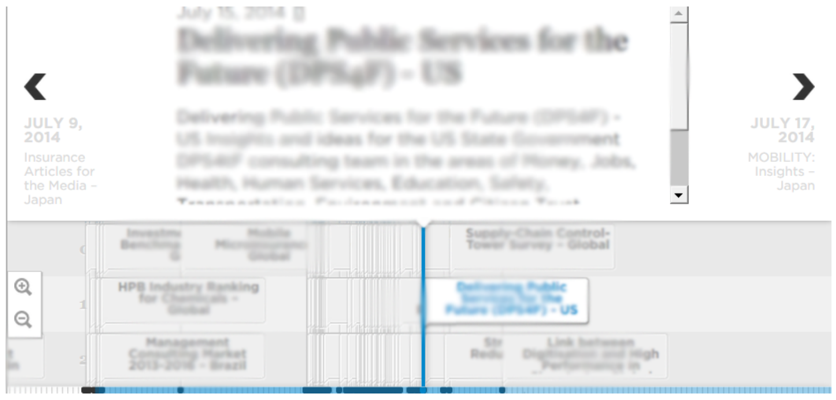

JULY 9, 2014
Insurance Articles for the Media – Japan
JULY 17, 2014
MOBILITY: Insights – Japan

Figure 3.8: The Timeline component of our visualization system. Specific extracts from corporate documents can be visually browsed based on their creation date, or the dates mentioned in the extract.

- *Complex Entity Visualization:* Our CER component produces serializes the recognized complex entities in a graph format that can then be visualized with specialized graph visualization tool such as GraphInsight [14]

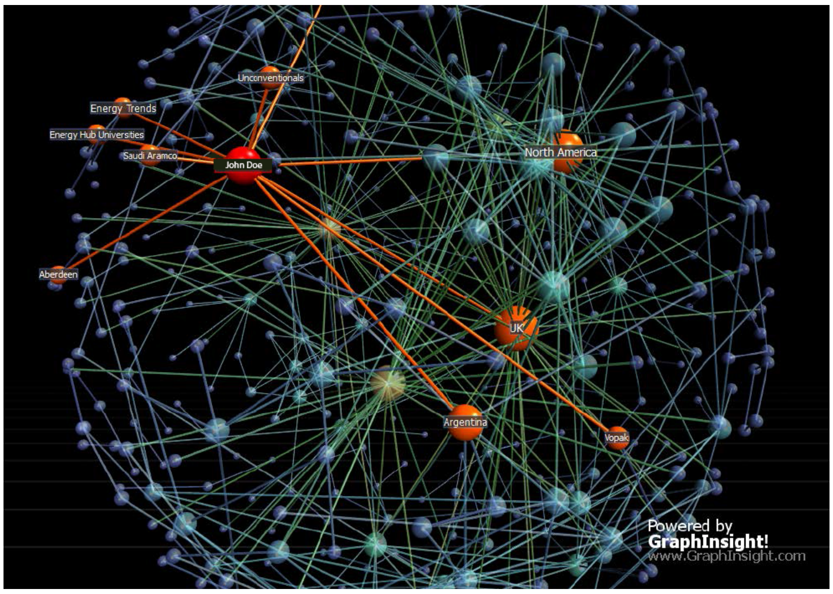


Figure 3.9: Example of a Complex Person Entity mined with our CER System Prototype and visualized with GraphInsight

3.2.6 Evaluation

The evaluation of the CER system was performed under two different aspects, scientific and industrial.

Scientific Evaluation

The scientific evaluation was performed by thoroughly evaluation different components of the system based on performance and recall as well as an evaluation of the annotation process. This evaluation has been published and will be shortly presented in this report.

The evaluation of the Ranking approach in [**?**] focuses on two main aspects:

- Comparison of the developed annotation-based approach with an established full text ranking approach
- Interpretation of the link-based metrics within the developed approach

Comparison with the full text ranking

Table 3.1 presents the results of the ranking approach $A1$ alongside with the overall score (calculated as in 3.8) and the full text ranking approach V, based on Solr/Lucene using the Lucene TFIDFSimilarity score function [7]. As evaluation

[7] https://lucene.apache.org/core/4_6_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html

metrics we calculated the MAE (mean absolute error) and NDCG (normalized discounted cumulative gain) measures for each of the 6 test queries from the golden standard (see [8]).

ID	Approach	MAE	NDCG
A1	Annotation-based	2.71	0.98
V	Full text	3.83	0.89

Table 3.1: Comparison of the full-text ranking with our annotation-based approach

Form our experimental results we can state that our ranking approach performs better regarding to the both evaluation measures:

- MAE: The deviation of the positions of the documents from the ideal position is on average smaller than using the full text ranking.
- NDCG: Both approaches are close to the optimum of 1. The annotation-based approach manages to sort out the relevant documents even better.

The advantage over a fulltext-based solution can also be noticed for individual search scenarios. Figures 3.10 and 3.11 show the values for MAE and NDCG, respectively. Only in one case the annotation-based ranking performs significantly worse (MAE for Scania). Apart from that our approach outperforms Solr in both measures.

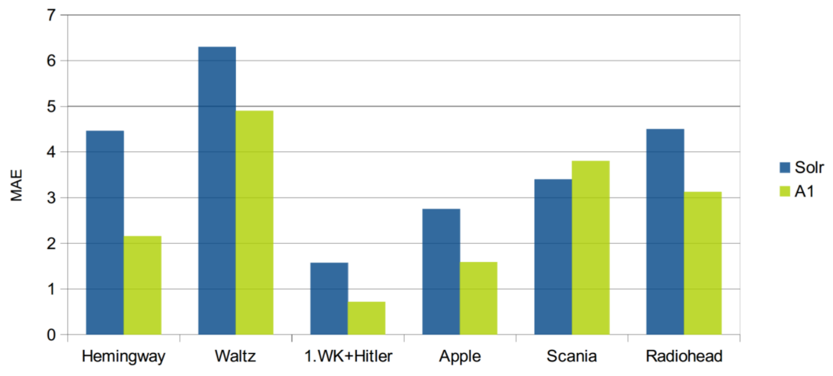


Figure 3.10: Comparison of full text ranking with the annotation-based ranking in regard to the *Mean Absolute Error*

Industrial Evaluation

An industrial evaluation has been performed with a startup company from Berlin and one of their clients which represents one of the largest financial consulting companies worldwide. The industrial evaluation was performed with two internal departments of the client company, for two different scenarios. The first scenario focuses on the management of internal company documents. Our application

[8] http://www.csw.inf.fu-berlin.de/ranking2015/evaluation.html

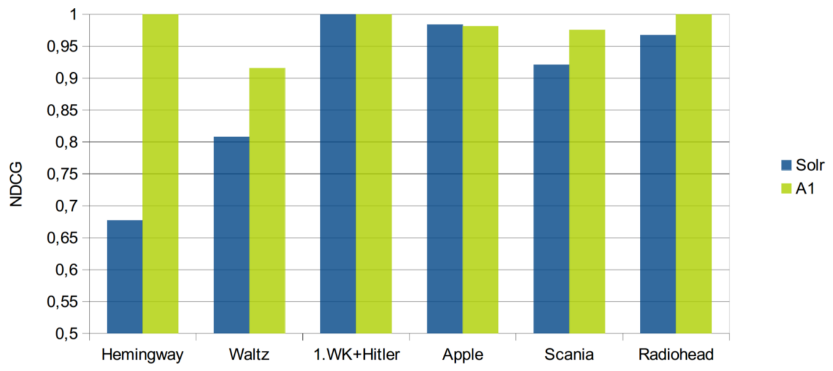


Figure 3.11: Comparison of full text ranking with the annotation-based ranking in regard to the *Normalized Discounted Cumulative Gains*

allowed the client to extract knowledge from unstructured internal documents. The CER process included document structure analysis, named entity recognition, document categorization, date recognition, acronym recognition and extraction, topic modeling and trend detection.

The other use case as to which the software was evaluated focused more on analyzing documents produced by competing companies. The complex entities we analyzed were internal company structure, collaboration structure, topic models and trending topics. The industrial evaluation currently resulted in high interest by the client company expressed trough frequent meetings with multiple departments, assessment by the client company that the system offers valuable insights that can be used by them, commitment towards further collaboration for the development of the software and concrete financial commitment in adherence with the rules imposed by the funding body.

3.3 Conclusion and Outlook

This chapter described out approach for complex entity recognition based on different graph-based approaches applied on an enriched knowledge base. We presented different aspects of our systems such as the data model used, a detailed document/entity/relation ranking approach as well as concepts and approaches for graph-based complex entity recognition and resolution.

In our future work we will focus on improving the cer approach by including other graph-based measures as well as topic modeling in the cer computation score.

Chapter 4

Knowledge-based Mining of Complex Event Patterns

Complex Event Processing provides means for handling a flow of events and answering questions about the current state of the stream. Enriching simple events with background knowledge from the application domain allows for a richer set of queries and for a wider range of applications.

Our work on mining patterns of complex events based on background knowledge resulted in two approaches that target different levels of the mining process. On the raw-event level, we developed a probability-based approach for mining of event pattern based on the distribution of their frequencies. On a higher level, we present our concept for knowledge-based mining of event patterns which fuses background knowledge from the domain to enable more complex patterns to be detected.

4.1 Probability-based Retrieval of Complex Event Patterns

In prediction-oriented applications, the goal is to identify the event that will occur next on the source. Our ability to estimate this probability depends on the amount of information we have about the past.

To achieve a good quality of prediction, we need to collect as much information as possible about the past of the event source, i.e., to consider as many event types as possible and to track patterns as long as possible.

However, the longer the patterns we detect, the more memory and computation power we have to provide. This makes detecting all actual patterns with no restriction on pattern length challenging or even impossible.

On the other hand, detecting short patterns is straightforward, especially when considering only a handful of frequent event types. But such patterns do not contain enough information to make a decision about an observed phenomenon. Yet they might provide useful information if the right heuristics were applied to extend them to longer patterns that efficiently utilize available history information.

The probability-based approach we propose estimates the frequencies of event patterns without having to detect all of them directly in the event source. The procedure starts with an initial set of patterns of a specific length n_0 representing event sequences whose frequencies are directly calculated from the source. Using this initial set, assuming we have the distribution of its items, the frequencies of longer patterns (of length $n' > n_0$) are estimated on multiple step until a target length n is reached.

For evaluating the method, we applied measures of Information Retrieval on the sets of most frequent/infrequent patterns. The tests were performed on a real-world dataset as well as on synthetic data. In the later case, where events are randomly generated, the quality of the results was also measured under various levels of entropy. The content of this section has been meanwhile published in our paper entitled Probabilistic Event Pattern Discovery [9].

4.1.1 Probabilistic Event Pattern Discovery

We start our discovery process by calculating the frequencies of all sequential patterns of a given initial length n_0. Given a set of E of primitive events, patterns of the form $\{A_0; A_1; ...A_{n_0}\}$ are detected and their frequencies are calculated.

To extend event patterns of the initial length n_0 resulted from a previous discovery phase, we apply conditional probability computations on their frequencies. We regard the target pattern as two overlapped patterns as shown below:

$$\begin{aligned} P &= \{A_1;A_2;A_3;\ldots A_{n_0-1};A_{n_0};A_{n_0+1}\} \\ P_0 &= \{A_1;A_2;A_3;\ldots A_{n_0-1};A_{n_0} \qquad\quad \} \\ \acute{P_0} &= \{ \qquad A_2;A_3;\ldots A_{n_0-1};A_{n_0};A_{n_{0+1}}\} \end{aligned} \tag{4.1}$$

We want now to estimate the frequency of the pattern P based on the frequencies of its subpatterns. We don't have enough information to calculate the probability of the intersection $A_{n_0+1} \cap P_0$ accurately, but we do have two pieces of the puzzle whose combination will give a good approximation of the actual value. Indeed, the probabilities of the patterns P_0 and $\acute{P_0}$ can be used to estimate the probability of the target pattern.

We can estimate this frequency as follows:

$$Fr(P) \approx Fr(P_0) \times Pr(\acute{P_0}) \tag{4.2}$$

In other words, the probability for the new event A_{n_0+1} to follow P_0 approximately equals the probability for it to follow the longest postfix of P_0. Estimating the frequency based on this equation lead to a loss of information which we will discuss thoroughly in the next section.

4.1.2 Evaluation

In order to evaluate our approach, the patterns resulting from extension have to be compared to the real patterns in the event source. The actual frequencies, calculated by running a naive algorithm on the whole dataset for once, make up our *gold standard* that serves as a control set.

For experiments on real-world data, we used the dataset provided by a Dutch academic hospital [1]. The event log contains data of treatments received by cancer patients and was distributed in XES format with the history of each patient listed within a *trace*. Each of those traces contains *event* tags that correspond to treatments. Events are marked with timestamps that indicate the date on which a treatment was performed.

For the purpose of our evaluation, we are particularly interested in temporal relations between successive treatments received by each patient. A patient usually visits the hospital regularly and receives multiple treatments in one *session*. Sessions do not have dedicated structural elements in XES format, but they can be recognized by the timestamps of the treatments where a session consists of events that have the same timestamp and are logged within the same trace.

Information Retrieval for Complex Event Patterns

For evaluating the quality of the retrieval results, we applied *Recall* and *Precision* [19] by ordering the resulted patterns according to their estimated frequencies and comparing the most frequent among them to the most frequent patterns of the gold standard.

By measuring the quality of retrieval by examining the set of most frequent patterns, we are no more concerned about the exact value of the pattern's frequency, but about its ranking among other patterns of the same length.

Accumulated MAPE

Our experiments of accumulated error rates consist of multiple iterations with different start conditions. In each iteration, we detect all patterns of length L and keep them as our gold standard. Patterns of length L_0 are then detected again and extended to patterns of length L. Each time, we compare the estimated frequencies resulting from extension with the gold standard in order to calculate the Mean Absolute Percentage Error, or MAPE.

The experiment, summarized in Figure 4.1, has shown that the error rate increases with each extension step. We reach the worst results when the initial set of patterns doesn't provide enough information about the event source.

Indeed, the highest error rate results when we start with patterns of length 2. After extending this set 7 times to get patterns of length 9, the MAPE value even exceeds one hundred percent.

However, when we start with longer patterns, the MAPE error rate starts low and grows more slowly. When patterns of length 5 are provided as initial set, it seems possible to double the length of the initial patterns with negligible error rates.

Retrieval of Frequent and Infrequent Patterns

In this test, we apply the technique of accumulated error rates we used for calculating *MAPE* values in the previous section to calculate information retrieval

[1]Real-life log of a Dutch academic hospital, originally intended for use in the first Business Process Intelligence Contest (BPIC 2011) `http://data.3tu.nl/repository/uuid:d9769f3d-0ab0-4fb8-803b-0d1120ffcf54` published by Eindhoven University of Technology.

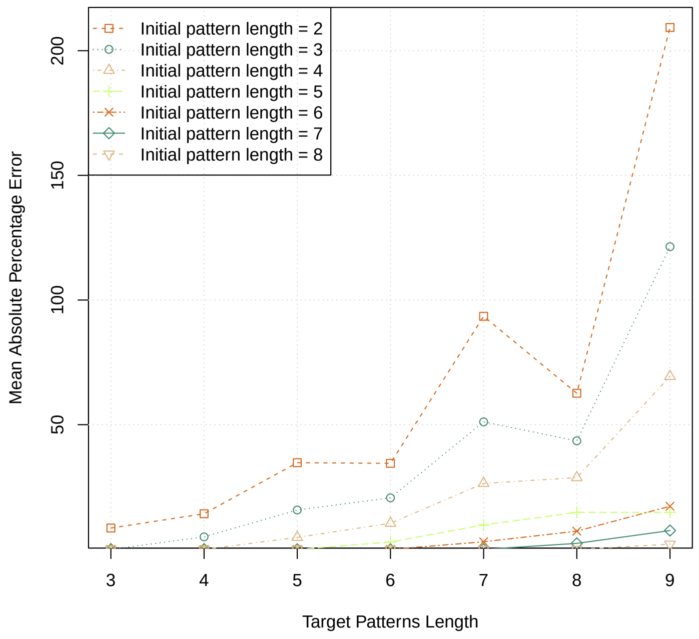


Figure 4.1: MAPE error rates resulting from pattern extension multiple times with various starting conditions.

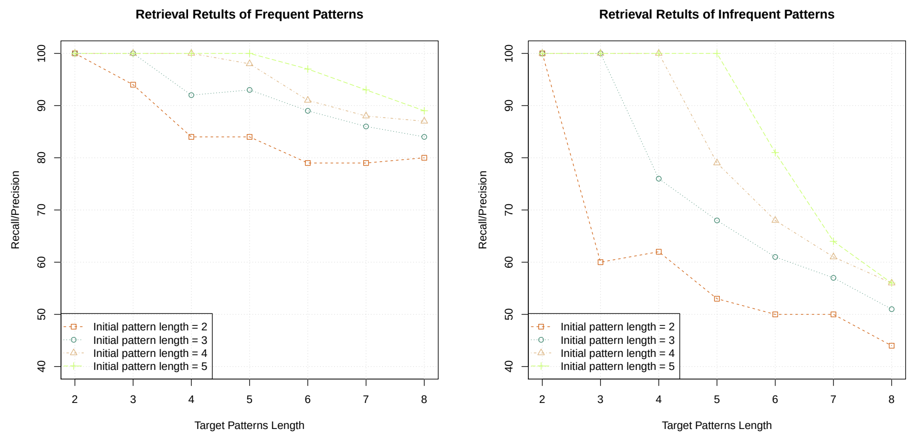
measures of recall and precision. Our aim is to study the effect of initial pattern set and the desired length of target patterns on the quality of retrieval.

The patterns come again from the dataset of the Dutch academic hospital and were submitted to two experiments. In the first one, we evaluated the retrieval of frequent event patterns, then we repeated the same test for infrequent patterns. The retrieval measures were calculated for 10 % of the most frequent patterns. The greater the intersection between the sets of most frequent patterns for extended patterns the gold standard, the higher values of recall and precision we get.

Figure 4.2 shows retrieval results for both sets. As the values prove, the more information provided as the basis for extension, the better the results that can be expected. In the tests that started with patterns of length 5, we have retrieved almost 80% of the most frequent patterns correctly.

We notice that recall and precision have the same value when the set of relevant patterns, i.e., the set of most frequent patterns in the gold standard, has the same size as the set of retrieved patterns, i.e., the most frequent patterns after extension, which is the case in our experiment where every pattern happens at least once.

As we will see later, the quality of the results is determined by the entropy of the event source and this value can differ from one level on the pattern tree to another. Thus, if the entropy of the patterns on some level is lower than that of the parent level, i.e. we can tell with more confidence what events will follow



(a) Evaluation of discovery results of the **most** frequent patterns in the Dataset.

(b) Evaluation of discovery of the **least** frequent patterns in the Dataset.

Figure 4.2: Retrieval results for frequent and infrequent event patterns

the patterns of this level, the extension results will be better.

Pattern Extension under Controlled Entropy

For observing the behavior of our procedure under various grades of entropy, we ran a test in which a synthetic event stream was generated using only two event types. The distribution of the two event types was controlled to achieve different levels of entropy in the generated stream. The result was twenty values of entropy ranging from 0 to 1 bit.

In our test, a randomizer generates, for each level of entropy, event instances for 30 seconds and sends them to a consumer. The later builds two pattern trees:

- *Gold standard Tree* containing patterns of length 12.
- *Initial Tree* with patterns of length 10 only.

When a generation burst is done, the consumer extends the patterns of the second tree to length 12. Recall and precision are then measured by comparing 10% of the most frequent patterns from both trees.

Figure 4.3 shows how recall and precision change under various entropy values. The obvious trend confirms that the quality of retrieval always gets worse with the increasing entropy.

When the entropy is zero, i.e. there is only one pattern in the whole stream, perfect recall and precision of 100% are achieved. However, the greater the entropy, the worse the results get.

4.1.3 Discussion

Probability-based retrieval of complex event patterns is an efficient way to estimate the frequencies of long patterns that are otherwise difficult to retrieve.

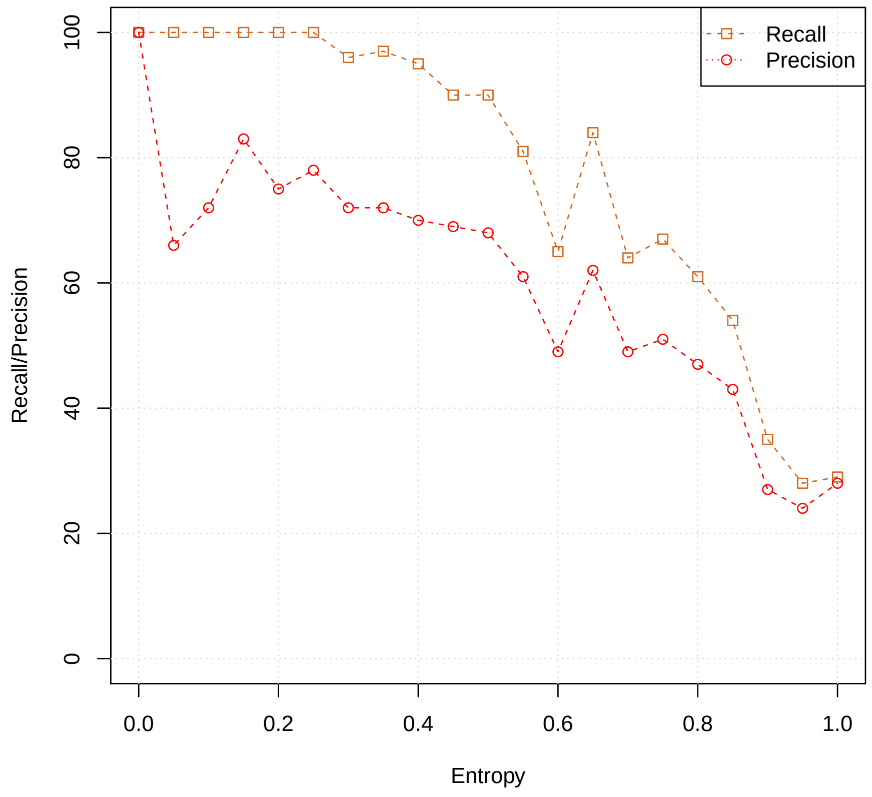


Figure 4.3: Retrieval results of the most frequent patterns extended from length 10 to 12 under controlled entropy.

The method saves processing time by computing the frequencies of whole patterns instead of processing single events and reduced memory by keeping a minimal amount of information about the past.

The quality of retrieval results depends particularly on the entropy of the event source. However, measuring the entropy, which is a property of the stream, might help estimate the reliability of the resulted ranking.

4.2 Parallel Relational Concept Analysis Framework

The main idea of the Parallel Relational Concept Analysis (PRCA) framework relies on the parallelization of the scaling step of object-object relations to relational properties and the integration of basic and relational properties into one concept lattice. The aim is to learn positive and negative hypotheses. PRCA does not generate a complete lattice with all relational properties, but only finds suitable relational properties that are good for dividing the positive from the negative examples. Therefore, the PRCA framework iteratively generates new relational properties from the relational information given in the data set and combines them with the basic properties in one lattice until sufficient hypotheses are found.

Figure 4.4 depicts the basic steps of the PRCA framework. The input of the framework consists of *relational contexts*. We define a relational context C as a pair (K, R) consisting of a set of formal contexts $K = \{K_i\}$, whereby each context $K_i = (O_i, P_i, I_i)$ has objects O_i, properties P_i and a relationship I_i between these objects and properties; and object-object relations $R = \{R_i\}$, with $R_i \subseteq O_1^i \times O_2^i$, associating objects from two contexts.

Each basic relation R_j has a source (relational) context C_i and a target (relational) context C_k. Source and target contexts can be identical. The main (relational) context is a learning problem with multiple contexts. One context is the main context. This is the context that contains the positive, negative (and unknown) labeled objects of the learning problem.

The learning algorithm consists of several steps.

In the *Generation of Relations* step, relations are generated based on the basic relations and properties of the relational contexts. The generator yields basic relations as well as new *composite relations*. A composite relation is the result of composing two relations or one relation and an additional post-condition. Different generation operators like joins, intersections and conditional joins exist. For instance, the relation join is defined as $R_{j.k} := R_j.R_k$. Applying a post-condition creates a new relation by applying filters based on properties in the target context.

In the following *Relational Scaling* step, these relations are scaled to *relational properties*. Different Scaling operators exist: existential, universal and cardinality restricted. There are also different scaling directions: left and right direction (has/is).

In the *Integration into Lattice* step, the relational properties are integrated with the basic properties into one lattice to check for new positive hypotheses, i.e. intentions of concepts that contain only positive examples). All new formal concept intentions being hypotheses are selected and stored.

In the *Learning* step, we verify whether all positive and negative examples are covered by at least one positive or negative hypothesis respectively. If all positive examples of the main context are covered by at least one hypothesis, the best hypotheses are selected and returned as result of the learning process.

4.2.1 Discussion

Our application scenario is in learning positive concepts of normality for detecting abnormal (i.e. negative) behaviors in the context of smart monitoring environments in ambient assisted living. Our event data sets don't only contain object-property relations but also more complex information relating objects to other objects which have properties. We therefore extended standard FCA-based learning on the basis of RCA for parallel learning on top of data with object-object relations. Due to the amount of data, high scalability of the learning method is relevant and the proposed parallel learner addresses this problem. The proposed approach is configurable and extensible which allows us to further study

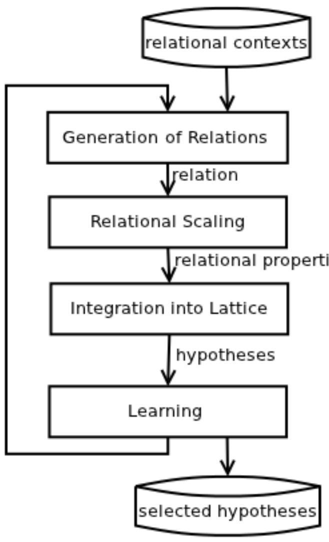


Figure 4.4: The basic steps of the Parallel Relational Concept Analysis (PRCA) framework

and evaluate relational concept analysis in different parallel configurations. We conducted experiments that have shown that we can handle data sets with one or multiple relational contexts and that PRCA outperforms DL Learner on the used data sets: PRCA finds a solution on straight data sets, where DL Learner doesn't find a correct solution. On data sets where both find solutions PRCA learns the definitions faster and achieves similar results for testing accuracy.

4.3 Semantic Mining of Complex Event Patterns

Events are semantic entities that can be further described using background knowledge. Therefore, semantic patterns can be used to extract complex events from a stream of events [28, 17, 27, 29]. Semantic mining of complex events is done based on relevant background knowledge about related resources.

Semantic Enrichment is the extension of events with background knowledge prior to complex event detection with new derived event attributes. In an Event Processing Network (EPN), several Event Mapping Agents (EMAs) are responsible for enriching raw events by querying external knowledge bases. Mapping agents can be replicated to achieve better scalability. In the following processing step, the enriched event stream can be monitored by multiple event processing agents. EMAs can be replicated and deployed in parallel to achieve efficient scalability with respect to throughput.

Definition 4.1 ***(Event Mapping Agents (EMA))*** *An Event Mapping Agent (EMA) is a software system responsible for receiving events and generating newly derived ones by querying external KBs and mapping incoming events to new events. Derived events are either completely new or raw events received with updated attributes (adding new fields or removing old ones).*

In order to enrich the event stream, new events can be derived from raw event instances. Such derived events can contain attributes that are *inferred* from external knowledge bases. A raw event stream can be enriched by one or many EMAs resulting in an enriched outbound event stream processed by a set of EPAs in order to detect complex events.

4.3.1 Types of Event Enrichment

Event stream enrichment can be categorized into two types. The first type is where the system generates new events and adds them to the output event stream in addition to raw instances. The second type is the fat event generation, i.e., generation of events with more attributes/values.

An event can be considered as a tuple of $\langle \bar{a}, \bar{t} \rangle$, where $\bar{a}$ is a multiset of fields $\bar{a} = (a_1, ..., a_n)$, and is defined by schema $\mathbb{S}$. In the process of event enrichment we will add new attributes to the multiset of fields of event instances or generate new event instances.

Enrichment of event streams is possible in the following two cases:

- **New Event Generation:** In the case of new event generation, an EMA can derive new event instances from each singular event instance and add them to the event stream. From a single event notification tuple, several event instances can be produced and sent downstream. The generated events might have the same type as the incoming stream, but they can be as well generated with new event types.

- **Fat Event Generation:** In the fat event generation approach, EMAs derive new attributes for each event instance and add them to their existing set of attributes. For each event $\langle \bar{a}, t_s, t_e \rangle$ the multiset of fields $\bar{a} = (a_1, ..., a_n)$ are extended to new fields and a is extend to $\bar{a} = (a_1, ..., a_(n+m))$ and m is the number of added fields.

4.3.2 Event Enrichment Prior to Pattern Mining

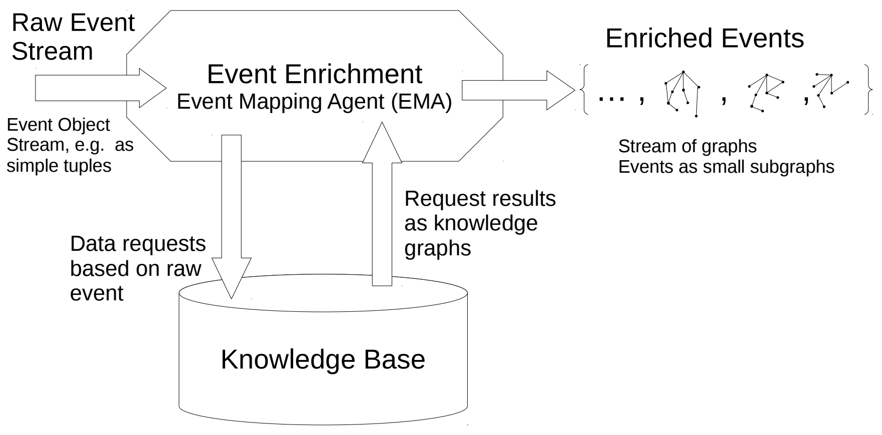
Regarding events as semantic entities, we need to retrieve background knowledge about them to understand more about their relations. By adding background knowledge retrieved from an external knowledge base, raw event logs can be used to generated new data types like graphs.

Our event enrichment approach looks for identifiers of raw events that can be used to point to resources in the background knowledge. Identifiers can be unique attributes of events that can build relations to resources in knowledge base. Fig. 4.5 depicts our approach for event enrichment.

Examples of Event Enrichment

For example, consider the case in which we have log files of all patient treatments in a hospital. A petition ID or a treatment ID can be used link the raw data to background knowledge about the patients or about each treatment. Such identifiers from event attributes can be used to generate queries for the extraction of related background knowledge about the events.

With this enrichment approach, it is possible to map the raw event objects to small-sized graphs extracted from the knowledge base. This enrichment step



Raw Event Stream
Event Object Stream, e.g. as simple tuples
Event Enrichment
Event Mapping Agent (EMA)
Enriched Events
{ … , , , }
Stream of graphs
Events as small subgraphs
Data requests based on raw event
Request results as knowledge graphs
Knowledge Base

Figure 4.5: Enrichment of Raw Event Stream.

results in a mapping of the raw event object (also known as event tuples), to a stream of small-sized graphs. An example is to map the patient treatments logs to a sequence of graphs that describes the treatment history of the patient and all of the treatments side effects.

Depending on the following mining process, this enrichment can vary to enrich events with much deeper knowledge or only a small subset of the existing knowledge about the events. As an example, consider the use case of patients treatment monitoring in a hospital (as it is used in the previous report), a monitoring system can have access to a stream of patients treatments as well as to a knowledge base containing information about a set of diseases, treatments and treatment history of each patients (assuming that this knowledge is prepared and stored in advance). We might enrich the raw treatment stream with only a subset of knowledge about the side effects and its relevance to the disease without considering the complete knowledge related diseases.

We can find more interesting use cases in scenarios of Internet of Things and ubiquitous computing. A relevant use case is the Ambient Assisted Living (AAL) [3] which aims at improving the quality of life, especially in accordance with the requirements of elderly people. AAL can be understood as age-appropriate assistance systems for a healthy and independent live.

Let's take for example Mr. Smith, a senior citizen who lives in a smart home well-equipped with various sensors signaling event streams from different areas of the house, e.g., movement in different areas of the house, or from household appliances, e.g., a coffee machine, a refrigerator or a washing machine. The refrigerator can register all its content and each time when one of the items is taken by Mr. Smith, it can signal the ID number of the food item. There are also medical sensors in the house which send blood pressure, blood sugar and pulse measurements of Mr. Smith in regular time intervals.

Mr. Smith's family doctor advises that some food ingredients are not good for Mr. Smith' health condition because of chronic diseases he has like hypertension (high blood pressure) and diabetes. In the case that Mr. Smith has abnormal

Ambient Assisted Living	
Property	**Description**
Raw event source	Event streams containing activities of elderly people
Complex event type	Detection of critical situations in elderly people
Background knowledge	Background knowledge about foods, drugs, allergies, diseases and doctor preferences
Event Enrichment	Enrichment of further relations, e.g., about foods, drugs, diseases, user/doctor preferences and the relations to other resources

Table 4.1: Overview of Information Sources for Ambient Assisted Living Use Case

blood pressure over a period of time, he should not eat certain foods, or in the case that Mr. Smith has already eaten one particular type of food, he should not take another type within a certain time period.

An intelligent system should monitor for drug and food interactions and combine this with all of Mr. Smith's activities. The system should send warnings to him and, in critical situations, send alarms to his family doctor. The system will receive a stream of raw event signals from different household sensors and medical sensors, and have access to a knowledge base which can store all the background knowledge about food types, Mr. Smith's diseases, and their relations.

The task of a monitoring system is to process the event stream close to real-time and integrate background knowledge from external knowledge bases. Background knowledge bases also include a logic reasoner for inference on the existing knowledge about, for example, food types, diseases and their relations. All the drug interactions, alcohol/food interactions with drugs, disease interactions with drugs can be monitored by a monitoring system if huge amounts of knowledge about drug, food and disease interactions are available in a knowledge base. Different warnings and alarms with different classifications like *"Major"*, *"Moderate"* or *"Minor"* as highly/moderately/minimally clinically significant can be triggered in the case that one of the event detection patterns are evaluated to be true.

Table 4.1 provides an overview of event stream sources and background knowledge used in ambient assisted living use case. Our approach of event enrichment adds further relations to the raw events like about foods, drugs, diseases, user/doctor preferences so that the final pattern mining system can extract patterns based on these relations. How deep the knowledge about different resources should be enriched depends highly on the specific use case and on *how deep the requested patterns relate to the depth of knowledge about each resource.*

4.3.3 Pattern Mining on Enriched Event Stream

As described before, after the enrichment step, an event mapping agent can produce a stream of small subgraphs generated from relations of event objects to resources in background knowledge. The enriched stream is a sequence of graphs (subgraphs of the knowledge stored in KB) built from a set of vertices (nodes, in this case resources) and edges (relations between resources). The knowledge bases used to enrich the events store the complete graph, when adding the event resources to it, so that it includes all of the streaming vertices and edges.

The patterns of complex events that we aim to mine are based on the graph patterns and the sequence of graph instances. Event detection operators (Like those described and reviewed in [10]) can be used to define patterns based on sequence of events. Our event detection patterns are specified based on combination of event detection operators and graph patterns that match graphs to each other. Previously, we have proposed [28, 17, 27, 29] that such patterns can be used to detect complex events based on the background knowledge about them.

Algorithms from sequential pattern mining or similar algorithms might be used to analyze the sequence of streaming subgraphs. Such algorithms should be customized to be able to work on graphs streams and its combination with event detection operators.

One initial idea might be to extract small patterns like triangles or rectangular patterns on each of the subgraphs (graph instances created from events) which makes a special meaning like an inheritance or inverse relations. Then we map again the stream to another form, a stream of simple patterns, that can be used to mine patterns of complex events. Consider the case in which we observe the frequencies of triangle patterns.

One further idea is to connect the subgraphs to each other because nodes (knowledge resources) that are used in the subgraphs can be connected with other subgraphs (they are resources in KB). In this way we can build a link between subgraphs. Considering a data window in which we have a set of subgraphs, we might only consider the subgraphs that have resources in common for further analysis.

Our future task is to finalize our basic ideas for event pattern mining based on a stream of graphs based on relations of event sequences and graphs.

4.4 Conclusion and Outlook

This chapter described our approach for event pattern mining based on syntactic processing of event sequences (probability based approach in Sec. 4.1) and its extension to semantic pattern mining (Sec. 4.3) by enriching the event stream in advance of pattern mining.

Our main future work is to finalize our concept for pattern mining from stream of subgraphs so that we can extract patterns based on different event matching operations in combination with graph structures.

Chapter 5

Conclusion

Success of content distributors and consumers is dependent on the delivery of relevant content tailored to the needs of the recipients. In this project, we envision a process chain for the creation of Corporate Smart Content, which we define as content enriched with the necessary corporate knowledge that enables applications for needs-based content delivery.

The process includes the in-house creation of content and/or importation of content from external sources, such as (linked) data repositories, audio/video archives, or news feeds as well as the creation of own or imported external ontologies, semi-automated, recommendation-based annotation of content and population of ontologies, and recommendation based enrichment with conceptual knowledge from the ontologies, as well as process knowledge mined from activity and event patterns.

Corporate Smart Content as the outcome of this process will enable the construction of smart applications, that allow for situation-aware and context-sensitive access to corporate content, that help employees or end-customers finding the content they need in order to get their work done, and that fits the current project they work in, the role they assume in the project, the current step in their process, and the information needs resulting from this situative context.

The sub-project of Freie Universität Berlin *Smart Content Enrichment* tackles three activities in this process:

Aspect-oriented ontologies: The project phase covered by this report comprised the design and prototypical implementation of the approach to aspect-orientation in ontologies and the design of an aspect-oriented extension to an API for ontologies. According to our working plan, the next phase will consist of finishing a prototypical implementation of the API extension, followed by an evaluation with our industrial partners with regard to the requirements and the integration into our general architecture and demo implementations that fulfill our vision of the Corporate Smart Content process chain.

Complex Entity Recognition: One of the key elements in corporate documents are named entities that give an idea about the main topics of those documents. Complex entities are classes or instances that have dependencies to other classes or instances. In this report, we described out approach for complex entity recognition based on different graph-based approaches applied on an enriched knowledge base. We discussed the different aspects of our systems such as the data model used, a detailed document/entity/relation ranking approach as well as concepts and approaches for graph-based complex entity recognition and resolution. We also presented a detailed description of the prototype system we implemented for complex entity recognition and corporate knowledge management. Our future work will focus on adding additional graph-based and topic modeling approaches in order to improve the insights gained from our complex entity recognition approach.

Knowledge-Based Mining of Complex Event Patterns: Identifying the most frequent patterns in a flow of events is a challenging task. In this report, we presented our probabilistic approach that reduces the memory and processing power required for solving the problem. Another mining problem is learning positive and negative hypothesis for which we also proposed a solution in Section 4.2. For mining more sophisticated patterns, streams of primitive events can be enriched with semantic elements from background knowledge. Our approach, described in Section 4.3, addresses this problem and provides an efficient solution for it. Future works will combine those approaches in a comprehensive mining system.

Appendix A

Work Packages

Work package FU 1	**Aspect-Oriented Ontology Development (AOOD)**	
Work package FU 1.2	**Development of a formalism for the expression of aspects in ontologies**	03/14-08/14
WP FU 1.2 Task 1.2.1	State of the art study on meta information formalisms in ontologies	03/14-08/14
WP FU 1.2 Task 1.2.2	Design of a meta data vocabulary for aspects in ontologies	03/14-08/14
WP FU 1.2 Task 1.2.3	Design of a mechanism for embedding aspect-oriented meta data in existing ontologies	03/14-08/14
WP FU 1.2 Milestone 1.2.1	Scientific validation (workshop paper)	03/14-08/14
Work package FU 1.3	**Design of an API for aspect-oriented access to ontologies**	09/14-02/15
WP FU 1.3 Task 1.3.1	State of the art study on ontology APIs	09/14-02/15
WP FU 1.3 Task 1.3.2	Requiements analysis	09/14-02/15
WP FU 1.3 Task 1.3.3	Design of an API for aspect-oriented access to ontologies	09/14-02/15
WP FU 1.3 Task 1.3.4	Technical evaluation of the design with the industrial partners	09/14-02/15
WP FU 1.3 Milestone 1.3.1	Technical evaluation of the design with the industrial partners	09/14-02/15

Work package FU 2	**Semantic Complex Entity Recognition and Annotation in corporate data**	
Work package FU 2.2	**Konzeption und Entwicklung von Verfahren fur die Semantische Erkennung, Annotation und Resolution von komplexen Entitaten**	03/14-08/14
WP FU 2.2 Task 2.2.1	Entwicklung und Auswahl eines Annotationsverfahren mageschneidert fur Unternehmensdaten.	03/14-08/14
WP FU 2.2 Task 2.2.2	Entwicklung von Konzepten fur die automatische Erkennung von komplexen Entitaten durch Einbeziehung der semantischen Relationen von einfachen Entitaten.	03/14-08/14
WP FU 2.2 Task 2.2.3	Entwicklung eines Verfahrens fur die Bewertung von Relationen anhand ihrer Relevanz in einem bestimmten Kontext durch inferenzbasierte Algorithmen.	03/14-08/14
WP FU 2.2 Task 2.2.4	Entwicklung von Konzepten fur die Automatische Verlinkung von Entiaten und internen und externen Wissensquellen. (Complex Entity Resolution)	03/14-08/14
WP FU 2.2 Task 2.2.5	Konzeption der Systemarchitektur fur die Analyse von Unternehmensdaten, Erkennung von Entitaten und Verlinkung der Entitaten mit Ontologien und semantischen Wissensbasen.	03/14-08/14
WP FU 2.2 Task 2.2.6	Entwicklung und Implementierung des ersten Prototyps des Complex Entity Recognition Systems (CER)	03/14-08/14
WP FU 2.2 Milestone 2.2.1	Erster Prototyp des Complex Entity Recognition Systems	03/14-08/14
Work package FU 2.3	**Erste Evaluation der Konzepte fur das Complex Entity Recognition System**	09/14-02/15
WP FU 2.3 Task 2.3.1	Technische Evaluation des Complex Entity Recognition Systems basierend auf verschiedenen qualitativen und quantitativen Kriterien mit den Industriepartnern	09/14-02/15
WP FU 3.3 Task 2.3.2	Evaluation der Qualitat der erkannten komplexen Entitaten sowie der Trefferrate des Systems.	09/14-02/15
WP FU 2.3 Task 2.3.3	Identifikation von konzeptionellen Verbesserungen des Complex Entity Recognition Systems	09/14-02/15
WP FU 2.3 Milestone 2.3.1	Wissenschaftliche und technische Validierung durch Veroffentlichungen	09/14-02/15

Work package FU 3	**Semantic mining of event data in corporate data for knowledge acquisition**	
Work package FU 3.2	**Konzeption und Entwicklung von Verfahren fur das semantische Prozess-Mining zur Wissensgewinnung**	03/14-08/14
WP FU 3.2 Task 3.2.1	Auswahl eines Anwendungsszenarios und eines Informationsportals fur die Datenextraktion und die Entwicklung des Mining-Systems	03/14-08/14
WP FU 3.2 Task 3.2.2	Entwicklung von Methoden zur Wissensgewinnung basierend auf den Ergebnisses aus AP FU 3.1	03/14-08/14
WP FU 3.2 Task 3.2.3	Spezifikation der Datenformate fur die Erfassung von Ereignisdaten und des Prozesskontextes	03/14-08/14
WP FU 3.2 Task 3.2.4	Konzeption der Systemarchitektur fur die Verarbeitung der Log-Datenflusse und deren Fusion mit existierendem Hintergrundwissen	03/14-08/14
WP FU 3.2 Task 3.2.5	Entwicklung und Implementierung des ersten Prototyps des Mining-Systems	03/14-08/14
WP FU 3.2 Milestone 3.2.1	Erster Prototyp des Mining-Systems	03/14-08/14
Workpackage FU 3.3	**Erste Evaluation der Konzepte fur das Mining-System**	09/14-02/15
WP FU 3.3 Task 3.3.1	Technische Evaluation des Mining-Systems basiert auf qualitativen und quantitativen Kriterien mit den Industriepartnern	09/14-02/15
WP FU 3.3 Task 3.3.2	Evaluation der Qualitat der durch Mining der Logdaten gewonnenen Metadaten	09/14-02/15
WP FU 3.3 Task 3.3.3	Identifikation von konzeptionellen Verbesserungen des Mining-Systems nach der ersten Evaluation	09/14-02/15
WP FU 3.3 Milestone 3.3.1	Wissenschaftliche und technische Validierung durch Veroffentlichungen	09/14-02/15

Appendix B

Acknowledgment

This work has been partially supported by the "InnoProfile-Transfer Corporate Smart Content" project funded by the German Federal Ministry of Education and Research (BMBF) and the BMBF Innovation Initiative for the New German Länder - Entrepreneurial Regions.

Bibliography

[1] Boanerges Aleman-Meza, Ismailcem Budak Arpinar, Mustafa V Nural, and Amit P Sheth. Ranking documents semantically using ontological relationships. In *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on*, pages 299–304. IEEE, 2010.

[2] Omar Benjelloun, Hector Garcia-Molina, David Menestrina, Qi Su, Steven Euijong Whang, and Jennifer Widom. Swoosh: a generic approach to entity resolution. *The VLDB JournalThe International Journal on Very Large Data Bases*, 18(1):255–276, 2009.

[3] José Bravo, Ramón Hervás, and Marcela Rodríguez, editors. *Ambient Assisted Living and Home Care - 4th International Workshop, IWAAL 2012, Vitoria-Gasteiz, Spain, December 3-5, 2012. Proceedings*, volume 7657 of *Lecture Notes in Computer Science*. Springer, 2012.

[4] Pablo Castells, Miriam Fernández, and David Vallet. An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 19:261–272, 2007.

[5] Zhaoqi Chen, Dmitri V Kalashnikov, and Sharad Mehrotra. Adaptive graphical approach to entity resolution. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 204–213. ACM, 2007.

[6] Alexandru Constantina, Silvio Peronib, Steve Pettiferd, David Shottone, and Fabio Vitalib. The document components ontology (doco).

[7] R.E. Filman and D.P. Friedman. Aspect-Oriented Programming Is Quantification and Obliviousness. *Workshop on Advanced Separation of Concerns, OOPSLA*, 2000.

[8] Benjamin Großmann, Alexandru Todor, and Adrian Paschke. Improving semantic search through entity-based document ranking. In *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics*, WIMS '15, pages 9:1–9:12, New York, NY, USA, 2015. ACM.

[9] Ahmad Hasan, Kia Teymourian, and Adrian Paschke. Probabilistic event pattern discovery. In Nick Bassiliades, Georg Gottlob, Fariba Sadri, Adrian Paschke, and Dumitru Roman, editors, *Rule Technologies: Foundations, Tools, and Applications - 9th International Symposium, RuleML 2015, Berlin, Germany, August 2-5, 2015, Proceedings*, volume 9202 of *Lecture Notes in Computer Science*, pages 241–257. Springer, 2015.

[10] Annika Hinze and Agnès Voisard. EVA: an event algebra supporting complex event specification. *Inf. Syst.*, 48:1–25, 2015.

[11] Jerry R Hobbs. Deep lexical semantics. In *Computational Linguistics and Intelligent Text Processing*, pages 183–193. Springer, 2008.

[12] Matthew Horridge, Bijan Parsia, and Ulrike Sattler. Laconic and Precise Justifications in OWL. In Amit Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy Finin, and Krishnaprasad Thirunarayan, editors, *The Semantic Web - ISWC 2008*, number 5318 in Lecture Notes in Computer Science, pages 323–338. Springer Berlin Heidelberg, January 2008.

[13] Atanas Kiryakov, Borislav Popov, Ivan Terziev, Dimitar Manov, and Damyan Ognyanoff. Semantic annotation, indexing, and retrieval. In *Web Semantics: Science, Services and Agents on the World Wide Web*, volume 2, pages 49–79. December 2004.

[14] Carlo Nicolini and Michele Dallachiesa. Graphinsight: An interactive visualization system for graph data exploration, 2013.

[15] Christine Parent and Stefano Spaccapietra. An Overview of Modularity. In Heiner Stuckenschmidt, Christine Parent, and Stefano Spaccapietra, editors, *Modular Ontologies*, number 5445 in Lecture Notes in Computer Science, pages 5–23. Springer Berlin Heidelberg, January 2009. DOI: 10.1007/978-3-642-01907-4.

[16] Adrian Paschke, Ralph Schäfermeier, Kia Teymourian, Alexandru Todor, and Ahmad Hassan. Corporate Smart Content: Requirements and Use Cases. Report I on the sub-project Smart Content Enrichment. Technical Report TR-B-14-02, Freie Universität Berlin, 2014.

[17] Adrian Paschke and Kia Teymourian. Rule based business process execution with bpel+. In *In Proc. 5th International International Conference on Semantic Systems (i-Semantics 2009)*, 2009.

[18] Cartic Ramakrishnan, William H Milnor, Matthew Perry, and Amit P Sheth. Discovering informative connection subgraphs in multi-relational graphs. *ACM SIGKDD Explorations Newsletter*, 7(2):56–63, 2005.

[19] C. J. Van Rijsbergen. *Information Retrieval.* Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.

[20] Cristiano Rocha, Daniel Schwabe, and Marcus Poggi Aragao. A hybrid approach for searching in the semantic web. In *Proceedings of the 13th international conference on World Wide Web*, pages 374–383. ACM, 2004.

[21] Ralph Schäfermeier and Adrian Paschke. Towards a Unified Approach to Modular Ontology Development Using the Aspect-Oriented Paradigm. In *7th International Workshop on Modular Ontologies (WoMO) 2013*, pages 73–78, 2013.

[22] Ralph Schäfermeier and Adrian Paschke. Aspect-Oriented Ontologies: Dynamic Modularization Using Ontological Metamodeling. In Pawel Garbacz and Oliver Kutz, editors, *Proceedings of the 8th International Conference on Formal Ontology in Information Systems (FOIS 2014)*, volume 267, pages 199 – 212. IOS Press, 2014.

[23] A.S. Schwartz and M.A. Hearst. A simple algorithm for identifying abbreviation definitions in biomedical text. In *In Proceedings of Pacic Symposium on Biocomputing*, volume 4, pages 451–462, November 2003.

[24] Amit Singhal. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.

[25] Tadej Štajner and Dunja Mladenić. Entity resolution in texts using statistical learning and ontologies. In *The Semantic Web*, pages 91–104. Springer, 2009.

[26] Friedrich Steimann. Domain Models Are Aspect Free. In Lionel Briand and Clay Williams, editors, *Model Driven Engineering Languages and Systems*, number 3713 in Lecture Notes in Computer Science, pages 171–185. Springer Berlin Heidelberg, January 2005.

[27] Kia Teymourian and Adrian Paschke. Towards semantic event processing. In *DEBS '09: Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*, pages 1–2, New York, NY, USA, 2009. ACM.

[28] Kia Teymourian and Adrian Paschke. Plan-based semantic enrichment of event streams. In *11th Extended Semantic Web Conference (ESWC 2014)*, Crete, Greece, May 2014.

[29] Kia Teymourian, Malte Rohde, and Adrian Paschke. Fusion of background knowledge and streams of events. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, DEBS '12, pages 302–313, New York, NY, USA, 2012. ACM.