Extraction of Ontology Schema Components from Financial News

Dissertation
zur Erlangung des akademischen Grades eines
Doktors der Philosophie
der Philosophischen Fakultät
der Universität des Saarlandes

vorgelegt von
Mihaela Vela

aus
Timişoara (Rumänien)

Saarbrücken, 2012

Dekan: Prof. Dr. Erich Steiner
Erster Berichterstatter: Prof. Dr. Hans Uszkoreit
Zweite Berichterstatterin: Prof. Dr. Silvia Hansen-Schirra
Tag der letzten Prüfungsleistung: 20.12.2011

Abstract

In this thesis we describe an incremental multi-layer rule-based methodology for the extraction of ontology schema components from German financial newspaper text. By *Extraction of Ontology Schema Components* we mean the detection of new concepts and relations between these concepts for ontology building. The process of detecting concepts and relations between these concepts corresponds to the intensional part of an ontology and is often referred to as ontology learning[1]. We present the process of rule generation for the extraction of ontology schema components as well as the application of the generated rules.

Most of the research on ontology learning (Cimiano et al., 2005; Aguado de Cea et al., 2008) investigates the learning potential at sentential level, after the corpus has undergone a deep linguistic analysis[2]. In this thesis we present a bottom-up method for the extraction of ontology schema components, showing that the extraction process of new classes and relations can be initialized at a more "lower" level using shallow and robust linguistic analysis.

We start the investigation by extracting candidates for ontology classes and relations from plain text, by applying text-based and string-based patterns. Then we go one step further and apply the accumulated knowledge from the previous step on Part-of-Speech (PoS) and semantically annotated text, validating in this way

[1] Ontology learning is the process of semi-automatic support in ontology development (Buitelaar et al., 2005)

[2] By deep linguistic analysis we mean grammatical function analysis.

candidates from the first step. In the last step, we augment the already detected ontological knowledge with classes and relations identified on the basis of phrase structure, or more precisely, from grammatical functions.

Zusammenfassung

In dieser Arbeit beschreiben wir eine inkrementelle mehrschichtige regelbasierte Methode für die Extraktion von Ontologiekomponenten aus einer deutschen Wirtschaftszeitung. Die Arbeit beschreibt sowohl den Generierungsprozess der Regeln für die Extraktion von ontologischem Wissen als auch die Anwendung dieser Regeln. Unter *Extraktion von Ontologiekomponenten* verstehen wir die Erkennung von neuen Konzepten und Beziehungen zwischen diesen Konzepten für die Erstellung von Ontologien. Der Prozess der Extraktion von Konzepten und Beziehungen zwischen diesen Konzepten entspricht dem intensionalen Teil einer Ontologie und wird im Englischen *Ontology Learning* genannt. Im Deutschen enspricht dies dem Lernen von Ontologien.

Der Großteil der Forschung im Bereich Lernen von Ontologien (Cimiano et al., 2005; Aguado de Cea et al., 2008) untersucht das Lernpotenzial auf Satzebene, nachdem der Korpus einer tiefen linguistischen Analyse unterzogen wurde[3]. In dieser Arbeit präsentieren wir eine inkrementelle Methode für die Erkennung von ontologischen Konzepten und Relationen. Die hier vorgestellte Arbeit soll zeigen, dass die Extraktion von neuen Klassen und Relationen auf allen Ebenen der linguistischen Annotation möglich ist. Ausserdem wird aus der Arbeit ersichtlich, dass das auf „niedriger" Ebene extrahierte Wissen durch das Wissen auf den „höheren" Ebenen ergänzt wird.

[3] Durch linguistische Analyse verstehen wir die grammatischen Funktionen.

Wir beginnen die Untersuchung mit der Extraktion von Kandidaten für Konzepte und Klassen aus reinen (nicht-linguistisch annotierten) Texten. Die Ermittlung wird von selbst entwickelten Regeln durchgeführt, die auf wiederkehrenden Mustern im Text basieren. Im nächsten Schritt wenden wir das im vorherigen Schritt angesammelte Wissen an auf linguistisch (in diesem Schritt nur Wortklassen) und semantisch annotierten Text. Auf dieser Weise ist es möglich die Kandidaten aus dem ersten Schritt zu validieren und neues ontologisches Wissen zu erkennen. Im letzten Schritt wird (auf der Basis von grammatischen Funktionen) das schon extrahierte ontologische Wissen durch neues ontologisches Wissen erweitert.

In Kapitel 3 fassen wir den Stand der Forschung im Bezug auf die hier beschriebene Arbeit zusammen. Wir befassen uns mit der Definition der Ontologie, beschreiben existierende allgemeine Ontologien und bestehende Untersuchungen im Bereich Extraktion von ontologischem Wissen. Der Schwerpunkt dieses Kapitels ist die detaillierte Beschreibung der bestehenden Ansätze im Bereich Lernen von Ontologien und Bevölkerung von Ontologien. Das Kapitel endet mit einer Diskussion über die Vor- und Nachteile der bestehenden wissenschaftlichen Forschung auf diesem Gebiet und stellt die Verbindung zwischen den schon existierenden Untersuchungen und der hier vorgestellten Arbeit her.

In Kapitel 4 befassen wir uns mit der Beschreibung der Methodologie des hier dargestellten Ansatzes. In diesem Kapitel beschreiben wir sowohl die Art und Weise wie die Regeln entstanden sind als auch die Anwendung dieser Regeln. Basierend auf der Annahme, dass verschiedene Ebenen der linguistischen Verarbeitung unterschiedlich relevantes Wissen für die Ontologieextraktion enthalten, gehen wir auf die verschiedenen Verarbeitungsebenen ein. Obwohl die flache linguistische Analyse für die Ontologieextraktion nicht genügt, möchten wir zeigen, dass Ontologieextraktion aus verschiedenen Ebenen der linguistischen Verarbeitung möglich ist.

In Kapitel 5 präsentieren wir eine detaillierte Beschreibung der entwickelten

Regeln und ihre Anwendung für die Extraktion von ontologischem Wissen. Wir beschreiben die drei Ebenen, auf denen die Ontologieextraktion durchgeführt wird. Auf der ersten Ebene wird im Detail das Potenzial für die Extraktion von ontologischem Wissen aus nicht-annotierten Texten dargestellt. Auf der zweiten Ebene wird der Extraktionsprozess auf der Basis der semantischen und Wortklassenannotation durchgeführt. Die letzte Ebene befasst sich mit der Extraktion von ontologischem Wissen aus Prädikat-Argument-Strukturen. In diesem Kapitel beschreiben wir auch den Prozess der Ontologiebevölkerung als willkommener Nebeneffekt der Verwendung des Annotationstools SProUT[4].

In Kapitel 6 stellen wir die Formalisierung der extrahierten Konzepte und Klassen vor. Dafür bedienen wir uns der Formalisierungssprache OWL DL, einer Variante der Formalisierungssprache OWL (Web Ontology Language) und eine gängige Formalisierungssprache für Ontologien. Das Kapitel ist in zwei Teile aufgeteilt. Im ersten Teil beschreiben wir OWL und den Unterschied zwischen OWL Lite, OWL DL und OWL Full. Im zweiten Teil konzentrieren wir uns, entsprechend den W3C Empfehlungun für OWL, auf die Beschreibung der tatsächlichen Formalisierung.

In Kapitel 7 befassen wir uns mit der Möglichkeit, den hier dargestellten Ansatz auch auf andere Texte und andere Sprachen anzuwenden. Dies wird einerseits anhand eines medizinischen Korpus (Radiologie) erörtert. Wir zeigen, wie der hier präsentierte Ansatz auf einer völlig unterschiedliche Domäne Anwendung findet. Andererseits wird der Versuch dargestellt, den hier vorgestellten Ansatz auf die französische Sprache anzuwenden.

In Kapitel 8 befassen wir uns mit der Evaluierung des hier vorgestellten Ansatzes. Die Evaluierung wird auf zwei Arten durchgeführt. Zuerst vergleichen wir unsere Ergebnisse mit den Ergebnissen des MUSING[5] Projekts. In diesem Kontext

[4] http://sprout.dfki.de/

[5] http://musing.eu

zeigen wir, inwieweit die bestehende MUSING Ontologie durch unsere Ergebnisse erweitert werden kann. Desweiteren gibt es eine numerische Auswertung, basierend auf die Berechnung der F-measure. Dafür wurde ein Teil unseres Korpus manuell annotiert und als Referenz für die Berechnung des F-measure Wertes verwendet.

In Kapitel 9 geben wir einen Überblick über die hier vorgestellte Arbeit. Darüber hinaus diskutieren wir verschiedene Aspekte der Integration, der Anwendbarkeit und der Erweiterung des hier vorgestellten Ansatzes.

Danksagung

Meinem Doktorvater und erster Berichterstatter, Hans Uszkoreit, danke ich für die Möglichkeit das Thema dieser Arbeit umzusetzen, für sein Vetrauen in mir, für seine wissenschaftliche Unterstützung und seinen fachlichen Rat.

Meiner zweiten Berichterstatterin, Silvia Hansen-Schirra, möchte ich für ihre Unterstützung in fachlichen und nicht-fachlichen Belangen danken.

Ein besonderer Dank gilt meinem Betreuer, Thierry Declerck, für die konstruktive Kritik und die gute Zusammenarbeit.

Und nicht zuletzt möchte ich Hans-Ulrich Krieger, Alastair Burt, Diana Steffen und Jörg Steffen für die fachliche und sprachliche Beratung danken.

Leider läßt sich eine wahrhafte Dankbarkeit mit Worten nicht ausdrücken.

`Johann Wolfgang von Goethe (1749-1832)`

Contents

List of Figures

List of Tables

Glossary

A-Box

In Description Logics the A-Box (assertional box) describes the attributes of individuals, the roles between individuals, and other assertions about individuals regarding their class membership with the T-Box concepts

BACH

BACH ontology relies on the Bank for the Accounts of Companies Harmonised database scheme

CYC

CYC is a formalized knowledge base for the representation of a vast quantity of fundamental human knowledge

EuroWordNet

EuroWordNet is a multilingual database for several European languages (Dutch, Italian, Spanish, German, French, Czech and Estonian)

GENIA

The GENIA ontology is intended to be a formal model of cell signaling reactions in human

GermaNet

GermaNet is a lexical-semantic net for German, which relates nouns, verbs, and adjectives semantically by grouping lexical units that express the same concept into synsets and by defining semantic relations between these synsets

MeSH

Medical Subject Headings is the United States National Library of Medicine's controlled vocabulary thesaurus

MUSING

MUSING stands for Multi-Industry Semantic-Based Business Intelligence Solutions

NACE

NACE is an European industry standard classification system

Ontology learning

Ontology Learning is concerned with the intensional part of a domain (T-Box), more specific with the detection of concepts and the development of relations between these concepts

Ontology population

Ontology population is concerned with the extensional part (A-Box) of a specific domain. More precisely, ontology population instantiates the concepts and relations defined by the T-Box

Paraphrase

The term paraphrase as used in this work, is in fact the reformulation of a compound. The difference to the classical definition of a paraphrase is that the paraphrase, as it is used in this work, does not always preserve the essential meaning of the material being paraphrased

Roget

Roget is a thesaurus of English which groups words in synonym and antonym categories

SCHUG

SCHUG (Shallow and Chunk Based Unification Grammar) is a parser, implemented in Perl for German and English

SCRIBO

SCRIBO stands for Semi-automatic and Collaborative Retrieval of Information Based on Ontologies

SProUT

SProUT (Shallow Processing with Unification and Typed Feature Structures) is a platform for the development of multilingual shallow text processing and information extraction systems

T-Box

In Description Logics the T-Box (terminological box) contains the concepts and relations of a certain domain. The T-Box is the structural and intensional component of conceptual relationships

UMLS

UMLS is a collection of controlled vocabularies in the biomedical domain

WordNet

WordNet is a large lexical database of English, where nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept

List of Abbreviations

Adj

Adjective

BDM

Balanced Distance Metric

BI

Business Intelligence

Conj

Conjunction

DL

Description Logic

DOBJ

Direct Object

EBMT

Example-Based Machine Translation

GEN_Phrase

Genitive Phrase

GF

Grammatical Function

GN

GermaNet Semantic Class

IE

Information Extraction

IOBJ

Indirect Object

IR

Information Retrieval

KB

Knowledge Base

KR

Knowledge Representation

MUSING

Multi-Industry Semantic-Based Business Intelligence Solutions

N

Noun

NE

Named Entity

NL

Natural Language

NLP

Natural Language Processing

NP

Nominal Phrase

NUNA

Non-Unique Name Assumption

OBIE

Ontology-Based Information Extraction

OWA

Open World Assumption

OWL

Web Ontology Language

PoS

Part-of-Speech

PP

Prepositional Phrase

PP_ADJUNCT

Prepositional Adjunct

Prep

Preposition

RDF

Resource Description Framework

RDFS

Resource Description Framework Schema

SC

Semantic Class

SCHUG

Shallow and Chunk Based Unification Grammar

SCRIBO

Semi-automatic and Collaborative Retrieval of Information Based on Ontologies

SPROUT

Shallow Processing with Unification and Typed Feature Structures

SUBJ

Subject

UMLS

Unified Medical Language System

V

Verb

VG

Verb Group

XML

EXtensible Markup Language

XSD

XML Schema

Chapter 1

Introduction

Linguistic-based ontology extraction from unstructured text is a topic strongly connected to major themes in the area of Computational Linguistics and Computer Science such as Semantic Web, ontologies and Natural Language Processing (NLP).

In this work we concentrate on unsupervised linguistic-based ontology learning from unstructured text. Purely linguistic approaches for ontology extraction are nowadays performed either with supervised methods (the user introduces the ontological knowledge into an existing form) or on the basis of phrase structure and grammatical function information. The method presented in this thesis investigates the ontology extraction potential at different levels of linguistic analysis. We start by describing the extraction potential from plain text by using only linguistic knowledge. In the next step we concentrate on the ontology extraction potential of text annotated with Part-of-Speech (PoS) and morphology. The last step is concerned with knowledge extraction on the basis of phrase structure. The three different layers offer different opportunities for ontology extraction and interact with each other.

1.1 Motivation and Aim of the Thesis

Why does one chose a certain topic for a thesis? The question can be answered in many ways: the interest for a specific research area, the high interest for a specific topic, previous work in a specific research area or even coincidence. The research presented in this thesis was influenced by a specific research context[1] and the existing research in the area of linguistic-based ontology extraction. As mentioned above, linguistic-based ontology learning is performed by directly using phrase structure for extracting ontological knowledge, without investigating ”lower” linguistic levels. The fact that, in this area, researchers do not investigate the extraction potential from shallow linguistics led us to the decision to investigate the ontology extraction potential at different levels of linguistic annotation. Our goal is to show that not only is phrase structure relevant when performing ontology extraction but that different linguistic annotation layers also provide additional valid information useful for knowledge extraction.

1.2 Research and Development Context

The work presented in this thesis has been partly funded by and has been used in the Multi-Industry Semantic-Based Business Intelligence Solutions (MUSING) project. The MUSING[2] project was an FP-6 funded Integrated Project on semantic technology enabled knowledge management applied to Business Intelligence (BI) (MUSING-Annual Public Report, 2009) which started in April 2006 and lasted for four years. The overview on MUSING in this section is based on the MUSING deliverables, but especially on MUSING-Annual Public Report (2009).

The project addressed three domains: Financial Risk Management (with par-

[1] By research context we mean the MUSING project.

[2] http://musing.eu

ticular reference to Credit Risk), Internationalization Services (with particular reference to location and partnership selection) and IT Operational Risk Measurement and Mitigation. In the field of Financial Risk Management MUSING has been developing and validating next generation semantic-based BI solutions. These solutions will be useful to financial institutions evaluating the financial health of enterprises. Concerning Internationalization, MUSING is approaching this aspect by the development and validation of next generation semantic-based platforms. Related to the IT Operational Risk Management, the project addressed the development and validation of semantic-driven knowledge systems for risk measurement and mitigation.

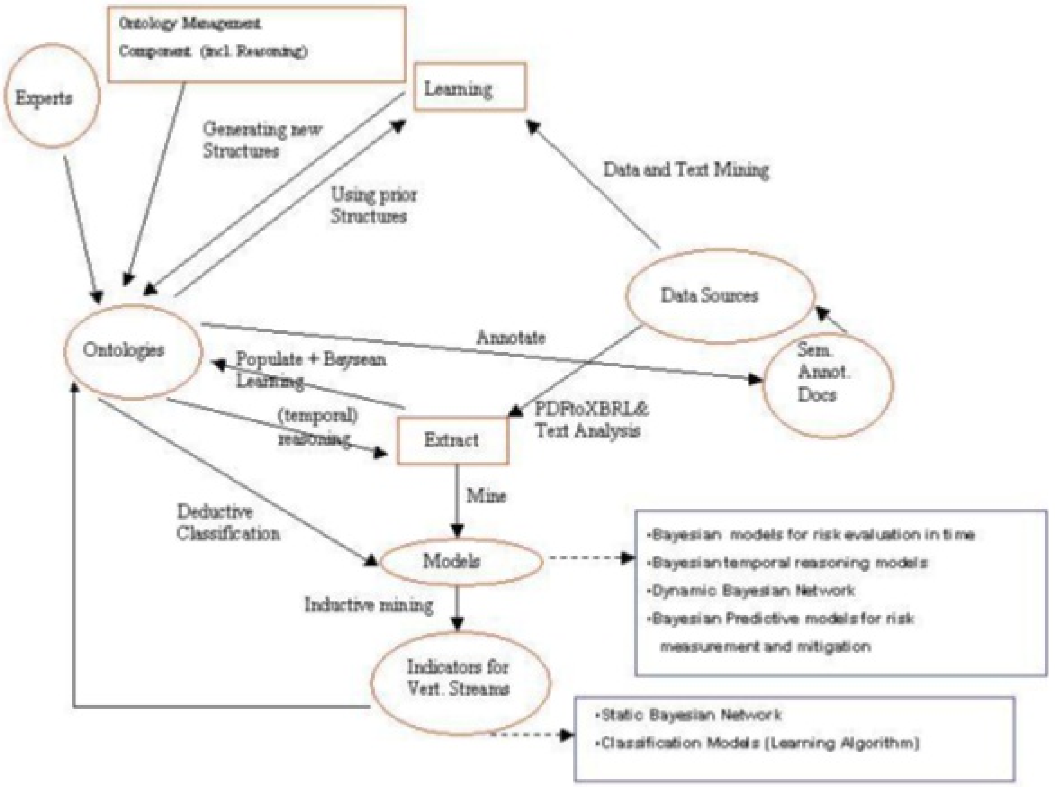

Ontology Management
Component (incl. Reasoning)
Experts
Learning
Generating new Structures
Using prior Structures
Data and Text Mining
Data Sources
Ontologies
Annotate
Populate + Baysean Learning
Sem. Annot. Docs
(temporal) reasoning
PDFtoXBRL& Text Analysis
Extract
Mine
Deductive Classification
Models
•Bayesian models for risk evaluation in time
•Bayesian temporal reasoning models
•Dynamic Bayesian Network
•Bayesian Predictive models for risk measurement and mitigation
Inductive mining
Indicators for Vert. Streams
•Static Bayesian Network
•Classification Models (Learning Algorithm)

Figure 1.1: Technological foundation components.

In MUSING, the technological development took place at various levels (MUSING-

Annual Public Report, 2009). Figure 1.1 depicts the technological foundation components. At a lower level, the project has been dealing with data collection and basic data analysis. At the knowledge representation level, the relevant information is extracted from the various data sources and is mapped into instances of ontology classes and concepts to ensure interoperability of the extracted information. At this level MUSING also provides a means to access knowledge, to update it and to check consistency. The last level, deals with the (re)usability of the built-in ontologies. Models exploiting this knowledge for supporting decision procedures, statistical and data mining models for exploiting the additional semantic features of ontologies are the aspects addressed at this level.

We conclude by saying that the MUSING project has developed and deployed a number of information extraction applications which target ontologies. The tools are designed to extract information from text which is then used to populate a knowledge repository and thus used by various business intelligence applications. During the project the conceptual model and the ontological commitments of the MUSING semantic BI have been constantly improved.

1.3 Thesis Structure

In Chapter 2, we provide definitions and descriptions of the different linguistic and semantic analysis steps. First we present the linguistic analysis tools which are commonly used for NLP. Then we give an overview on available semantic resources which we consider relevant for the work presented here. Although not all NLP tasks require semantic tagging, semantics is helpful for the more specific domain of ontology learning.

In Chapter 3, we present the state of the art with reference to the work presented here. First we concentrate on defining the concept of an ontology, then we give an overview on existing ontologies by describing them. The chapter continues with a detailed description of the existing approaches in the area of ontology learning and ontology population. We conclude this chapter by discussing the pros and cons of the existing scientific research in the ontology learning area by pointing out the advantages of the work presented here.

In Chapter 4, we present the methodology applied in our approach. We show that our work can be divided into two main parts: the construction of the rules and the application of these rules. Based on the assumption that shallow linguistic analysis is useful for the ontology learning process, we present here a bottom-up method for the extraction of ontology schema components. We do not argue that shallow linguistic analysis is enough for building an ontology. Rather than that, we show that much of the ontological knowledge can be extracted more easily and faster if we also use shallow linguistic analysis and show we how the ontology extraction rules generated from grammatical functions round out the designed set of rules.

In Chapter 5, we present a detailed description of the designed rules and their application for the extraction of ontological knowledge. We describe the three layers on which the ontology extraction is performed. We first specify in detail the potential for ontology extraction from plain text (first layer), respectively from text annotated with PoS and lexical semantics (second layer). The last layer deals with the extraction of ontological knowledge from predicate-argument structures. We are able to cover a wider range of linguistic phenomena, extending this way also the relation set. We also describe how, on the basis of the annotated Named Entity (NE), we are able to perform ontology population.

In Chapter 6, we describe the formalization of the ontological knowledge extracted by the method presented in this thesis. The formalization approach follows the W3C Recommendation for OWL, the Web Ontology Language, respectively OWL DL. After giving an overview on DL and OWL DL, we show how the ontological knowledge extracted with the method presented in this thesis is formalized in OWL DL.

Chapter 7 proofs evidence of how the approach can be extended to other areas. From that perspective we concentrate on showing how our approach applies to a completely different domain and a different language. As a domain we have chosen a corpus from the medical filed, respectively from radiology. We show how the rules actually developed for newspaper text can be applied with some restrictions also on the radiology corpus. Also part of this chapter is the application of the designed rules on a language from a different language family, such as French. The characteristics of the French language allow us to demonstrate the applicability of our rules for the extraction of ontological knowledge from compounds, paraphrases and modification phenomena.

In Chapter 8, we present the evaluation of our results from two perspectives. First we compare the results with the MUSING ontology and show to what extent we can extend the existing ontology. The second type of evaluation concerns the numeric evaluation by using a F-measure. For that purpose we manually built and annotated a test suite which was our reference when performing the F-measure calculus.

In Chapter 9, we give a summary of the work presented so far and we discuss different aspects concerning the integration, applicability and extension of the approach presented here in future research.

Chapter 2

Language Technologies and Semantic Resources Used in this Thesis

NLP is concerned with the interaction between Natural Language (NL) and computers. In order to make NL processable, the linguistic information encoded in free texts has to be made visible and understandable by computers. This means that both linguistic and semantic information encoded in free texts has to be brought into a machine understandable format. This is achieved by annotating the free texts with linguistic and semantic information. The result of the annotation process returns semi-structured texts from which computers can read relevant information. On the other hand, NLP implies the processing of large amount of data which makes it impossible to perform linguistic and semantic analysis manually. Therefore, the actual state of the art for the linguistic and semantic annotation is to use tools which perform the annotation (semi-)automatically.

In this chapter we present the different analysis steps used in this thesis. In order to perform the task of this thesis, we used part-of-speech tagging, morphological analysis, chunking, dependency structure analysis and semantic resources. The next sections provide definitions and descriptions of the different linguistic and semantic analysis steps based on Buitelaar and Declerck (2003). Section 2.1 describes possible linguistic analysis steps, whereas Section 2.2 describes some of the available semantic resources.

2.1 Linguistic Analysis

In this section we present the linguistic analysis tools used in this thesis and which are also commonly used for NLP. It is important to note that not all annotation steps are required in order to achieve good results. Depending on the approach and the proposed goals, the user can choose one, more or all linguistic annotation steps. In the following sections we concentrate on the shallow analysis (PoS tagging, morphological analysis and chunk parsing) as well as on grammatical function analysis.

2.1.1 Part-of-Speech Tagging

The Part-of-Speech (PoS) tagger assigns to each word, depending on its context, a syntactic class (e.g. noun, verb, adjective). The PoS tagging of a given text will return for each word in the text its corresponding PoS tag. The allocation of a syntactic category to a word in a given context implies more than just the assignment of PoS tags to this word. It also implies disambiguation, since words may have different meanings in different contexts (e.g. *light* as adjective or *light* as noun). Another aspect covered by a PoS tagger is the tokenization, which is not always trivial (e.g. *l'addition* in French). The currently available PoS tagger are either rule-based taggers (Brill, 1992) or statistical taggers (Schmid, 1994; Brants, 2000). For the approach presented here we used the PoS tagger integrated into SProUT[1] (Drozdzynski et al., 2004).

2.1.2 Morphological Analysis

The morphological analysis identifies, analyzes and describes the structure of words concerning their derivational, inflectional and compounding information. A morphological analysis of a given text returns for each word in the text information about the stem of the word, its inflectional properties (gender, number, case, tense, etc.) and, if possible, its compound analysis. The morphological analysis implies also disambiguation, which in most of the cases is interacting with PoS tagging and chunking described in Section 2.1.3. For example, the compound *Staubecken* can be interpreted as *Staub-Ecken* (*dusty corners*) or *Stau-Becken* (*reservoir*). Depending on the compound analysis, we are then able to detect which meaning is being used here. For German it is easier to perform morphological disambiguation, since German, in contrast to English, is a morphologically rich language. The available morphological analyzers usually use language dependent lexicons for their analysis. There are several morphological analyzer available for different languages. Here are some of them: PC-KIMMO (Koskenniemi, 1983), MMORPH (Petitpierre and Rusell, 1995), MORPHIX (Finkler and Neumann, 1988), IDX[2]. For our work we use the morphological analyzer integrated into SProUT (Drozdzynski et al., 2004).

[1] SProUT (Shallow Processing with Unification and Typed Feature Structures) is a platform for the development of multilingual shallow text processing and information extraction systems which incorporates in it a morphological analyzer and a PoS tagger.

[2] http://www.dfki.de/lt/project.php?id=Project_359&l=en

2.1.3 Chunking

Chunk parsing implies the identification of bigger linguistic units[3] in a sentence such as nominal phrases, prepositional phrases, adjectival phrases, adverbial phrases or verbal groups. A chunk parser returns just a list of identified chunks, which means that not all words in a text will be part of a chunk. For example punctuation signs do not belong to any of the phrases listed above. The available chunk parsers are either rule-based, such as SCHUG (Declerck, 2002), or built on statistical metrics, such as Chunkie (Skut and Brants, 1998). Statistical-based chunk parsers imply a training session on a training set before applying it on the corpus. Chunking is closely related to the dependency structure analysis, since in order to detect the grammatical functions, the chunks need to be identified. We used for our work SCHUG.

Also closely related to chunks is the NE (Named Entity) recognition process, since specific chunks or parts of chunks can be recognized as NEs (persons, organization, etc.). Usually NE recognition is performed by using lists with persons, organizations, etc. (also called gazetteers) in combination with regular expressions.

2.1.4 Dependency Structure Analysis

The dependency structure analysis identifies the dependencies between different chunks and words in a sentence. This means that chunks have already been identified and the dependency structure analysis assigns the dependency information to chunks. A complete dependency analysis covers, on the one hand, the identification of the dependencies between chunks in sentences and, on the other hand, the dependencies between the components of chunks. The dependen-

[3] By bigger we mean more word units.

cies between chunks are usually identified by the grammatical functions (subject, object), whereas the dependencies between the components of a chunk are determined by annotating the head, complement and modifier of a chunk. At the sentence level, the dominating node in the sentence tree is the predicate[4]. At the chunk level the dominating node of the tree is the identified head of the chunk. The complements within a chunk are the necessary qualifiers of the head and the modifiers within a chunk are the optional qualifiers of the head. As with chunk parsers, the dependency parsers can be either rule-based such as SCHUG, which we used, or statistical parsers, such as LoPar (Schmid, 2000) and Minipar (Lin, 1998).

2.2 Semantic Resources

Although not all NLP tasks require semantic tagging, such tagging has proved to be helpful for information extraction in general, and also for the more specific domain of ontology learning. Applications in these NLP fields require semantic analysis, which is performed on the basis of available semantic resources. Semantic resources are typically semantic lexicons, thesauri and semantic networks. In the next sections we describe some semantic lexicons, thesauri and semantic networks. Although we use in this thesis only the semantic lexicon GermaNet (Kunze and Lemnitzer, 2002), we also present here the closely related semantic thesauri and semantic networks. The presentation of the semantic thesauri and semantic networks in this context is motivated by the fact that this type of semantic resources could also be easily integrated into the work presented in this thesis. We decided to use only GermaNet as a semantic resource because we considered it the most appropriate for the method presented in this thesis.

[4] If chunk dependencies are determined by grammatical functions.

2.2.1 Semantic Lexicons

Semantic lexicons are semantic resources that group together words according to lexical semantic relations like synonymy, hyponymy, meronymy and antonymy (Buitelaar and Declerck, 2003). WordNet, EuroWordNet, GermaNet are semantic lexicons. Semantic lexicons are in fact lexicons enhanced with semantic information.

WordNet

WordNet[5] is a lexical reference system developed by the Cognitive Science Laboratory at Princeton, available online and whose design is inspired by psycholinguistic theories of human lexical memory. Although linguistically motivated, many groups have used it as a general ontology of concepts.

Within WordNet English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. It covers currently over 90000 semantic classes (*synsets*). Different relations link the synonym sets (e.g. antonyms, generalizations, etc). Synsets are collections of synonyms, grouping together lexical items according to meaning similarity. For example, the two synsets *[board, plank]* and *[board, committee]* are grouped together because a board and a plank are similar lexical items. At the same time, a board may also refer to a group of people. The synsets in WordNet range from very specific to very general, specific synsets covering a small number of items, general ones a large number of items.

[5] http://wordnet.princeton.edu/

GermaNet

GermaNet[6] is a lexical-semantic lexicon that relates German nouns, verbs, and adjectives semantically by grouping lexical units that express the same concept into synsets and by defining semantic relations between these synsets. GermaNet has much in common with the English WordNet and might be viewed as an on-line thesaurus or a light-weight ontology. GermaNet contains 57776 synsets, 81773 lexical units, 72057 literals, 12042 lexical relations and 68997 conceptual relations.

EuroWordNet

EuroWordNet[7] is a multilingual database for several European languages (Dutch, Italian, Spanish, German, French, Czech and Estonian). EuroWordNet is structured in the same way as the WordNet in terms of synsets (sets of synonymous words) with basic semantic relations between them. Each language is represented with a unique internal system of lexicalisations. In addition, the languages are linked to an Inter-Lingual-Index, which is based on WordNet1.5. Via this index, the languages are interconnected so that it is possible to go from the words in one language to similar words in any other language.

FrameNet

FrameNet (Fillmore, 1982) is an online lexical semantic resource for English, based on frame semantics and supported by corpus evidence. The aim is to document the range of semantic and syntactic combinatoric possibilities (valences) of each word in each of its senses, through computer-assisted annotation of example sentences and automatic tabulation and display of the annotation re-

[6] http://www.sfs.uni-tuebingen.de/GermaNet/

[7] http://www.illc.uva.nl/EuroWordNet/

sults. The major product of this work, the FrameNet[8] lexical database, currently contains more than 11600 English lexical units, more than 6800 of which are fully annotated, in more than 960 semantic frames, exemplified in more than 150000 annotated sentences. The creation of the German FrameNet was also part of the SALSA (The SAarbrücken Lexical Semantics Annotation and Analysis) project[9].

2.2.2 Thesauri

According to Bußmann (2008), a thesaurus is a dictionary in which the lexical items of a language are arranged systematically. A more specific definition is given by Buitelaar and Declerck (2003), which describe thesauri as semantic resources which group together similar words or terms according to a standard set of relations like broader term, narrower term, etc. A thesaurus may also include language equivalents and translation terms. In the following we present the Roget thesaurus and the Medical Subject Headings (MeSH) thesaurus.

Roget

Roget[10] is a thesaurus of English which groups words in synonym and antonym categories. First published in 1852, the Roget thesaurus has evolved to one of the widely used dictionaries.

MeSH

MeSH[11] is the United States National Library of Medicine's controlled vocabulary thesaurus. It consists of sets of terms naming descriptors in a hierarchical

[8] http://framenet.icsi.berkeley.edu/

[9] http://www.coli.uni-saarland.de/projects/salsa/page.php?id=index-salsa1

[10] http://machaut.uchicago.edu/rogets

[11] http://www.nlm.nih.gov/mesh/

structure that permits searching at various levels of specificity.

2.2.3 Semantic Networks

Bußmann (2008) defines semantic networks as graphs in which the nodes are connected to each other by relations. The same definition is also provided more explicitly by Buitelaar and Declerck (2003) which defined semantic networks as semantic resources that group together objects denoted by natural language expressions (terms) according to a set of relations that originate in the nature of the domain of application (The UMLS Semantic Network, CYC).

UMLS

The UMLS[12] is a compilation of more than 60 controlled vocabularies in the biomedical domain and is being created by the National Library of Medicine under an ongoing research initiative that supports applications in processing, retrieving, and managing biomedical text (Rindflesch and Aronson (2002)). Some of the medical terminologies integrated in UMLS are the Medical Subject Headings (MeSH), Systematized Nomenclature of Medicine (SNOMED), International Statistical Classification of Diseases and Related Health Problems (ICD), Physicians' Current Procedural Terminology (CPT), and Clinical Terms Version 3 (Read Codes).

The UMLS Knowledge Source is structured around three separate components: the Metathesaurus, the Semantic Network, and the SPECIALIST lexicon. The UMLS Metathesaurus is a multilingual thesaurus which contains semantic information about more than 8000000 biomedical concepts, each concept having variant terms with synonymous meaning.

[12] http://www.nlm.nih.gov/research/umls/

English terms from the Metathesaurus are also included in the SPECIALIST Lexicon, which contains more than 140000 entries of general and medical terms. The SPECIALIST Lexicon encodes morphosyntactic information about English nouns, verbs, adjectives and adverbs. Each concept in the Metathesaurus is also related to a semantic category from the Semantic Network, in which 134 semantic categories interact with 54 relationships.

CYC

The CYC[13] knowledge base (Knowledge Base (KB)) is a formalized representation of a vast quantity of fundamental human knowledge: facts, rules of thumb, and heuristics for reasoning about the objects and events of everyday life. The medium of representation is the formal language CycL. The KB consists of terms - which constitute the vocabulary of CycL - and assertions which relate those terms. These assertions include both simple ground assertions and rules.

At the present time, the CYC KB contains nearly two hundred thousand terms and several dozen hand-entered assertions about/involving each term. New assertions are continually added manually to the KB and CYC adds a vast number of assertions to the KB by itself as a product of the inferencing process. Additionally, term-denoting functions allow for the automatic creation of millions of non-atomic terms.

2.3 Conclusion

This chapter provided an overview on the different linguistic annotation tools and semantic resources used in this thesis. The aim of this chapter was to show what kind of information researchers can access in order to achieve their objectives. Of

[13] http://www.cyc.com

course, it is not obligatory to use all information. Depending on the approach, a researcher can use only the useful linguistic annotation steps.

Concerning the approach presented in this thesis, we use both semantics and shallow and dependency structure analysis. Since the aim of this thesis is to show that the detection of ontology schema components returns good results on the basis of shallow linguistic analysis (without necessarily using dependency structure analysis), we exploit all annotation levels presented above. Additional to the linguistic annotation, we take advantage from the available semantic resources and combine the linguistic analysis with semantic resources.

Chapter 3

State of the Art for Ontology Learning, Population, and Representation

In Chapter 1 we have argued for a multi-layer approach to ontology learning from unstructured text. In this chapter, we will give a detailed description of the existing approaches in this field addressing the following issues: the difference between ontology learning and ontology population, the current methods for ontology learning and the representation of ontologies.

In the vision of Berners-Lee et al. (2001), the Semantic Web is an extension of the current Web which is enriched with well-defined meaning, enabling this way machine interpretability and interoperability. For the Semantic Web to function, the semantically enriched information has to be represented in a way that allows the preservation of its meaning, but is, at the same time, abstract enough to provide machine-readability. In order to achieve this, Berners-Lee et al. (2001) proposes the use of ontologies. Nowadays, OWL has become the language for representing ontologies. The Web Ontology Language (OWL)[1] builds on a restricted subset of Resource Description Framework (RDF)[2], and EXtensible Markup Language (XML)[3] is a way of representing it.

In the following sections, we will give a detailed description of the state of the art in ontology learning and ontology population, concentrating on the approaches which we considered relevant for the research presented in this thesis. Before doing so, we give the definition of an ontology in Section 3.1 and describe in Section 3.2 the state of the art concerning formalizing an ontology. Section 3.4 presents current approaches in the area of ontology learning and Section 3.5 presents research in the area of ontology population.

3.1 Current Definition of Ontologies

Although the term ontology emerges from philosophy, in the fields of Computer Science and Computational Linguistics, an ontology is a formal explicit specification of a shared conceptualization (Gruber, 1993). Conceptualization refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. Explicit means that the type of concepts used, and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine-readable. Shared reflects the notion that an ontology captures consensual knowledge, that is, it is not private of some individual, but accepted group (Studer et al., 1998).

An ontology is build of two components: the ontology schema and the instances. The ontology schema contains the concepts and relations of a certain domain and is the structural and intensional component of conceptual relationships. The

[1] http://www.w3.org/TR/owl-guide/
[2] http://www.w3.org/RDF/
[3] http://www.w3.org/XML/

instances describe the attributes of individuals, the roles between individuals, and other assertions about individuals regarding their class membership. The instances correspond to the extensional component of an ontology.

3.2 Ontology Representation

Concerning the formalization language for ontologies, the Web Ontology Language OWL (Staab and Studer, 2004) has emerged as the state of the art formalism in knowledge representation. OWL is a recommendation of the World Wide Web Consortium[4] (W3C), builds on RDF[5] and RDF Schema[6], and has been designed to be read by computers. For the work presented here, we adopt OWL DL and follow the W3C specification for OWL[7].

3.2.1 A Short Introduction to Description Logic

In this section, we give a short overview on Description Logics, since OWL-DL (the ontology language for the present work) relies on it. This section gives an overview Description Logics based on Baader and Nutt (2003), Nardi and Brachman (2003) and Baader (2003).

Description Logic (DL) comprises a family of Knowledge Representation (KR) formalisms that describe the knowledge of an application domain and are based on fragments of first-order logic. In contrast to the predecessors, DLs can be given a formal logic-based semantics. Description Logics make two important assumption: the Open World Assumption (OWA) and the Non-Unique Name Assumption (NUNA). Open world assumption means that what cannot proven

[4] http://www.w3.org/
[5] http://www.w3.org/RDF/
[6] http://www.w3.org/TR/rdf-schema/
[7] http://www.w3.org/2004/OWL/

to be true is not believed to be false. The non-unique name assumption allows individuals with different names to be equal.

Knowledge Representation based on DL consists of two components: the T-Box (the terminological box), introducing the terminological knowledge of an application domain and the A-Box (the assertional box), which holds the knowledge about the individuals of the application domain. The T-Box is build through declarations that describe the application domain and contains the concept and role definitions. The A-Box contains assertions about concepts and roles from the T-Box.

Besides terminologies and assertions, DL also provides the possibility of reasoning about them. This means that, by using logical inference, implicit knowledge of an application domain can be made explicit.

The basic DL description language is $\mathcal{AL}$ (attributive language) and has been introduced by Schmidt-Schauß and Smolka (1991). More expressive description logic languages can be obtained by adding further constructors to $\mathcal{AL}$. In the following paragraphs we sketch the basics of the DL language $\mathcal{AL}$. Table C.4 in Appendix C.1 list possible extensions of DL and the corresponding naming of the new DL languages.

Elementary DL descriptions are atomic concepts (unary predicates, also called concept names) and atomic roles (binary predicates, also called role names). Atomic concepts are *Person*, *Female*, whereas atomic roles are *hasChild*, *hasParent*. From these atomic concepts and roles, we can build complex descriptions using concept and role constructors. Concept constructors are, for example, intersection, union, value restriction or cardinality restriction. The group of role constructors includes, for example, transitive, inverse or atomic roles[8]. The relation between concepts and roles is stated by terminological and assertional axioms.

[8] We have to mention here that $\mathcal{AL}$ does not provide role constructors.

Terminological axioms specify the relation between concepts and roles. Terminological axioms are the subsumption and the equivalence axiom. Assertional axioms are the concept assertion and the role assertion.

A more detailed description of DL concepts, roles, constructors and axioms and the corresponding syntax is given in Appendix C.1.

3.2.2 The Web Ontology Language

The Web Ontology Language (OWL) is based on the logical formalism called Description Logic (DL) (Baader et al., 2003) and has three variants: OWL Lite, OWL DL and OWL Full. The OWL variants are ordered hierarchically, such that every legal OWL Lite ontology is a legal OWL DL ontology and every legal OWL DL ontology is a legal OWL Full ontology. W3C describes the three OWL variants as follows. OWL Lite was designed for easy implementation and to provide users with a functional subset that will get them started in the use of OWL (Bechhofer et al., 2003). It was hoped that it would be simpler to provide tool support for OWL Lite than its more expressive relatives, allowing a quick migration path for systems utilizing thesauri and other taxonomies. OWL DL extends OWL Lite with disjunction, negation, cardinality constraints and nominals allowing more complex constructs[9]. The development of OWL Lite tools has proved almost as difficult as development of tools for OWL DL, and OWL Lite is not widely used.

OWL DL was designed to provide the maximum expressiveness possible while retaining computational completeness (either φ or $\neg\varphi$ hold), decidability (there is an effective procedure to determine whether φ is derivable or not) and the availability of practical reasoning algorithms (Bechhofer et al., 2003). OWL DL language constructs can only be used under certain restrictions. For example,

[9] Constructs are the RDF schema features (such as class, subClassOf), properties of classes, property characteristics and restrictions.

number restrictions may not be placed upon properties which are declared to be transitive. OWL DL is named so due to its correspondence with Description Logic, more precisely $\mathcal{SHOIN}$(D). A more detailed description of the meaning of $\mathcal{SHOIN}$(D) is given in Table 3.1.

Extension	**Symbol**
Negation	$\mathcal{C}$
Number restrictions	$\mathcal{N}$
Qualified number restrictions	$\mathcal{Q}$
Role hierarchy	$\mathcal{H}$
Role inverse	$\mathcal{I}$
Nominals	$\mathcal{O}$
Functional roles	$\mathcal{F}$
$\mathcal{ALC}$ + transitive roles	$\mathcal{S}$

Table 3.1: DL extensions.

OWL Full is based on a different semantics then OWL Lite or OWL DL, and was designed to preserve some compatibility with RDF Schema. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. This is not permitted in OWL DL. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary (Bechhofer et al., 2003).

3.2.3 OWL Elements

OWL provides the capability of creating classes and properties, of defining instances of classes and relationships between these instances. In the following we will present the basic OWL elements.

Classes

Ontology classes are the most basic concepts in a domain, since much of the power of ontologies comes from class-based reasoning. The most basic concepts in a domain should correspond to classes that are roots of various taxonomic trees. Every individual in the OWL world is a member of the class *owl:Thing*. Therefore, each user-defined class is implicitly a subclass of *owl:Thing*. Besides individuals, which are instances of the class, a class may also contain other subclasses.

An example for subclasses are *Person* and *Manager*. *Person* is a subclass of the class *owl:Thing*, but *Manager* is a subclass of the class *Person*. OWL also provides additional constructors to form classes. These constructors can be used to create class expressions. OWL supports the following operations on classes: union (`unionOf`), intersection (`intersectionOf`) and complement (`complementOf`). It also allows class enumeration (`oneOf`) and disjointness (`disjointWith`). For example, `oneOf` provides a method to specify a class via a direct enumeration of its members. This way, the class extension is defined without allowing other individuals for the respective class.

Individuals

In order to describe the members of a defined class, we need individuals (also called instances). A class is simply a name and a collection of properties that describe a set of individuals. Individuals are the members of this set. Classes correspond to naturally occurring sets of things in a domain of discourse, whereas individuals should correspond to actual entities that can be grouped into these classes. For example, *Joseph Ackermann* is an instance of the class *Manager*.

Properties

In order to connect the classes, respectively their instances, OWL allows for the definition of properties between the class instances. A property is a binary relation that specifies class characteristics. There are two types of simple properties: datatype properties and object properties. `Datatype properties` are relations between instances of classes and data types, such as integer or string. For example, `hasAge(String, Integer)` is a property of the class *Person*. Object properties are relations between instances of two classes. For example, `hasPosition` may be an object type property of the *Person* class and may have a range which is the class *Position*. A possibility to restrict a relation in OWL is by defining the domain and the range of the respective relation. The domain restricts the applicability of the relation of the left argument and the range restricts the applicability of the relation on the right argument. For example, the object property `hasPosition` can be specified by defining the class *Person* as domain and class *Position* as range. By this restriction the instances of the class *Company* are related to the instances of the class *Person*. Defining the domain and the range of a property is just a way for specifying a relation. By using property characteristics such as transitive, symmetric, functional and inverse we can further specify properties in OWL. Besides characterizing a property, we can also constrain its range. For this purpose, OWL provides property restrictions such as `allValuesFrom`, `someValuesFrom`, `cardinality` and `hasValue`. We can restrict the property `hasPosition` with domain *Company* and range *Person* by `allValuesFrom`, so that for every position in a company, the occupier of the position has to be a male.

3.3 Available OWL Ontologies and Tools

Already existing ontologies can be divided into core ontologies and domain ontologies. Core ontologies are the so-called general ontologies which provide the skeleton for domain ontologies. SUMO[10] and Proton[11] are core ontologies. Domain ontologies are ontologies specific to a certain domain. They can build on an already existing core ontology, an already existing domain ontology or can be built from scratch. The biggest resource for domain ontologies seems to be provided by the biomedical domain. The Open Biomedical Ontologies (OBO) Foundry[12] and the United States National Center for Biomedical Ontology (NCBO) BioPortal[13] are good references for this domain. On the other hand, on Protégé's[14] site a wast collection of ontologies can be found. For the finance domain, we notice here the MUSING[15] ontology, which includes also a finance ontology. Other ontologies can be found by searching for appropriate search terms with the filetype set to ".owl" or ".rdf" or by using the Swoogle semantic web search engine.

One of the advantages of OWL ontologies is the availability of tools that reason about them. At this point, there are several reasoners for OWL. We list here just some of them: RACER[16], FaCT++[17], Pellet[18], KAON2[19]. Tools for editing an ontology are also available such as SWeDE Eclipse Plugin[20], Protégé, TopBraid[21]. For the validation of ontologies, we list here the WonderWeb OWL ontology

[10] http://www.ontologyportal.org/

[11] http://proton.semanticweb.org/

[12] http://www.obofoundry.org/

[13] http://bioportal.bioontology.org/

[14] http://protege.stanford.edu/

[15] http://musing.eu/

[16] http://www.racer-systems.com/

[17] http://owl.man.ac.uk/factplusplus/

[18] http://clarkparsia.com/pellet

[19] http://kaon2.semanticweb.org/

[20] http://owl-eclipse.projects.semwebcentral.org/

[21] http://www.topquadrant.com

validator[22], the W2C validator [23] and the Swoop OWL validator[24].

3.4 Ontology Learning

Ontology learning (or knowledge acquisition) is concerned with the intensional part of a domain (T-Box), more specific with the detection of concepts and the development of relations between these concepts. When it comes to describing research done on ontology learning, Hearst (1992) is the reference study. Her proposal has inspired much of the research in this area. Hearst (1992) proposed a set of predefined lexico-syntactic patterns to automatically acquire hyponymy[25] lexical relations from corpus. The proposed linguistic patterns were then applied to a corpus in order to build up a general domain thesaurus, which was verified and augmented by using WordNet.

For ontology learning, we divide the state of the art research into two categories: rule-based approaches and machine learning approaches. Both ontology learning categories are using to some extent linguistic analysis. As described in the following sections, for rule-based approaches linguistics is the main component, whereas for machine learning approaches linguistics is a welcomed additional instrument. The decision to select the approaches below as state of the art is motivated by the fact, that we consider them relevant for the method presented in this thesis.

[22] http://www.mygrid.org.uk/OWL/Validator

[23] http://www.w3.org/RDF/Validator/

[24] http://code.google.com/p/swoop/

[25] In linguistics, a hyponym is a word or phrase whose semantic range is included within that of another word, its hypernym. For example, *manager*, *boss*, *chief* are all hyponyms of *leader* (their hypernym), which in turn, is a hyponym of *person*.

3.4.1 Rule-Based Approaches

Aguado de Cea et al. (2008) present work done on linguistic-based ontology learning for English[26] in the context of the NeOn[27] project. The aim of the NeOn project is to advance the state of the art in using ontologies for large-scale semantic applications in the distributed organizations. In this context, they developed patterns for solving design problems for the domain classes and properties of an ontology. This patterns are called Ontology Design Patterns (ODP), they are based on OWL ontology and can be divided into Structural, Correspondence, Content, Reasoning, Presentation, and Lexico-Syntactic ODPs (Presutti et al., 2008). An Ontology Design Pattern is in fact a modeling solution (in OWL or other logical languages) to solve a recurrent ontology design problem.

The goal of the research reported by Aguado de Cea et al. (2008) is to facilitate naive users the ontology building process by a predefined system, the S.O.S system, which contains ODPs. In the work presented here, the authors concentrate on patterns for modeling natural language into an ontology. This patterns are called Lexico-Syntactic Patterns (LSP). The ideal case would be to model a natural language sentence directly into the ontology by using Lexico-Syntactic Patterns, since facts defined in natural language have to be modeled with ontology concepts and relations.

Aguado de Cea et al. (2008) define the Lexico-Syntactic ODPs as formalized linguistic schemata or constructions derived from regular expressions in natural language. A LSP consists of certain linguistic and paralinguistic elements, following a specific syntactic order, which permit some conclusion about the meaning they express. With other words, the Lexico-Syntactic ODPs are linguistic-based patterns which are formalized in the project's ODP style.

[26] Spanish and German are also planned.

[27] http://www.neon-project.org/

Aguado de Cea et al. (2008) define six LSPs corresponding to the `subclass-of relation`, `object property`, `datatype property`, `disjoint classes`, `part-whole relation` and `participation` ODPs. The developed list of LSPs are in fact linguistic patterns which extract the specific OWL properties from text. Figure 3.1 shows how the LSPs for the `subclass-of relation` are defined. `CN` denotes the class name, `CD` a cardinal number, `NP` a nominal phrase, `PARA` a paralinguistic symbol like column, `CATV` a set of verbs of classification plus the preposition that follows them (e.g. *classify in/into*). The parentheses `()` group two or more elements, elements appearing in `[]` are optional and the `*` indicates repetition.

```
1. NP<subclass>be [CN] NP<superclass>
2. [(NP<subclass>,)* and] NP<subclass>be [CN] NP<superclass>
3. [(NP<subclass>,)* and] NP<subclass>(group into|as|in) |
   (fall into) | (belong to) CN NP<superclass>
4. NP<superclass>CATV [CD] [CN] [PARA] (NP<subclass>,)*
   and NP<subclass>
5. There are CD CN NP<superclass> PARA [(NP<subclass>,)* and]
   NP<subclass>
```

Figure 3.1: Deriving the `subClassOf` relation with LSPs.

The five rules for extracting the `subclass-of relation` match the following examples in Figure 3.2.

```
1. An orphan drug is a type of drug.
2. Odometry, speedometry and GPS are types of sensors.
3. Thyroid medicines belong to the general group of
   hormone medicines.
4. Membrane proteins are classified into two major categories,
   integral proteins and peripheral proteins.
5. There are two types of narcotic analgesics: the opiates
   and the opiods.
```

Figure 3.2: Sentences matching the `subClassOf` relation from Figure 3.1.

Some of the linguistic patterns described for ontology extraction by Aguado de

Cea et al. (2008) are polysemous, since verbs like *include* or *comprise* may introduce both the `subclass-of relation` and the `part-whole relation`. In these cases, the authors propose interaction with the user by using refining questions. This way, the system would help users to decide the correct modeling.

Aussenac-Gilles and Jacques (2008) present CAMÉLÉON, a method and a tool that supports a knowledge engineer in identifying relations and concepts for ontology engineering from French corpora. The presented tool provides the manual definition and evaluation of semantic relations extracted from corpora, and the modeling of the found relations into concepts and properties in an ontology. This task is divided into two modules. The first module supports pattern definition, pattern matching and pattern testing. The second one helps in integrating the extracted knowledge into the ontology.

In order to define new patterns from a given corpus, Aussenac-Gilles and Jacques (2008) propose the following alternatives: to adapt already existing patterns (also called generic patters) in CAMÉLÉON, to define new patterns for already identified domain relations and to define new relations and patterns after observing the contexts in which related terms are used. All three alternatives require that the adapted or defined patterns are searched in the corpus and validated. The validation of a pattern is done by checking some of the sentences in which the pattern appears.

Concerning the process of discovering new patterns, the CAMÉLÉON tool assumes that the corpus in which new patterns are searched is tagged with a PoS tagger and is represented by the KESKYA[28] concordance tool. The patterns do not extend beyond the sentence boundaries and are expressed by using lemmas combined with PoS, phrase type and operators, such as *or*, negation, and iteration. Figure 3.3 depicts two patterns which have been already added to the pattern repository in CAMÉLÉON. The first one is a definition pattern, whereas

[28] http://emdros.org/

the second is a hypernymy pattern.

```
1. <definir> 1 <comme>
2. NP1 <étre> 1 ART_DEF NP2 ART_DEF (plus|moins)
```

Figure 3.3: Definition pattern in CAMÉLÉON.

Figure 3.4 lists the sentences on which these patterns apply.

```
1. Un Project Logiciel peut se définir comme un Processus
   de Développement.
   A software project may be defined as a development process.
2. Le lave des coulées est la roche volcanique la plus résistante.
   The lava of lava flow is the most resistant volcanic rock.
```

Figure 3.4: Sentences matching the definition pattern in Figure 3.3.

Concerning the second module of the presented tool, extending the existing ontology, this is a task done manually by the user. In fact, after a pattern has been identified and applied to the corpus, the user has to analyze each pattern matching result, respectively the corresponding sentence, and to decide whether the pattern discovered new concepts and relations. This type of reasoning about introducing new concepts and relations into the ontology is complex, time-consuming and can be made only by an ontology engineer.

The evaluation of the extracted patterns is performed by applying the already existing patterns (also generic patterns) on eight corpora from eight different domains. The domains can be grouped in the following categories: technical writings, scientific papers, handbooks. These pattern repository comprises definition patterns and hypernymy patters, meronymy patterns as well as one pattern for reformulation and two 'varia' patterns. For the definition patterns evaluation is performed by measuring precision and recall of the available patterns. For the remaining patterns evaluation is performed by measuring just precision[29], since

[29] Precision is measured by dividing the correct phrases by the entire number of phrases matched by the pattern

no reference sentences were available for this kind of patterns. The results show that precision varies a lot in the corpus, as the frequency of specific phenomena varies between different corpora. This leads to the conclusion that there is no generic pattern, which can be easily applied on all corpora. The adaptation of patterns for each corpus improves efficiency, since the user does not need to analyze the corpus in order to observe and describe new patterns.

Ciaramita et al. (2008) present a method for unsupervised learning of semantic relations between ontological concepts in the biology domain. The method is based on the idea that relations can be represented as syntactic dependency parsers between ordered pairs of named entities. The presented system takes as input an English corpus and a set of concepts, applies deep syntactic analysis (Charniak, 2000) and generates based on patterns and constraints a set of candidate relations which are ranked, selected and possibly generalized. The extracted relations are ranked by using the chi-square measure and selected if the chi-square measure passes a given threshold. The generalization of the extracted rules is performed by an iterative algorithm proposed by Clark and Weir (2002). This algorithm takes as input a relation *r*, a class *c* and a syntactic slot *s* and returns a class *c'*, which is either *c* or one of its ancestors. The mapping of the extracting relations into the ontology occurs manually, since the relations are tested on their compatibility and consistency to the GENIA[30] ontology.

Ontology learning by using linguistic patterns is also a task of the ongoing **SCRIBO** project[31]. The aim of Semi-automatic and Collaborative Retrieval of Information Based on Ontologies (SCRIBO) is the development of algorithms and collaborative free software for the automatic extraction of knowledge from texts and images, and for the semi-automatic annotation of digital documents. In this context, the extracted ontological knowledge from French texts covers the ex-

[30] http://www-tsujii.is.s.u-tokyo.ac.jp/ genia/topics/Corpus/genia-ontology.html
[31] http://www.scribo.ws/xwiki/bin/view/Main/

traction of concepts, relations and the population with Named Entities (NE) (De la Clergerie, 2009). In order to achieve this, De la Clergerie (2009) proposes the following stages: linguistic analysis, extraction of the ontological knowledge, insertion into the ontology of the extracted ontological knowledge and the post-editing of the ontology by ontology experts. De la Clergerie (2009) proposes also the combination of the linguistic patterns with machine learning techniques by using Harris' distributional hypothesis[32]. Because this is work in progress, an evaluation or intermediate results are not provided yet.

3.4.2 Machine Learning Approaches

Cimiano et al. (2005) present a machine learning approach for the automatic acquisition of concept hierarchies from English text corpora. The presented research is based on the Formal Concept Analysis (FCA), a method based on order theory. The Formal Concept Analysis is mainly used for discovering inherent relationships between objects described through a set of attributes on the one hand, and the attributes themselves on the other. The data are structured into units which are formal abstractions of concepts, allowing meaningful comprehensible interpretation (Ganter and Wille, 1991). In order to apply the Formal Concept Analysis, Cimiano et al. (2005) analyze their corpus with shallow and deep linguistic analysis tools.

The approach can be describes as follows. First, the corpus is annotated with PoS by the TreeTagger (Schmid, 1994) and parsed using LoPar (Schmid, 2000). From the dependency tree they extract verb-argument dependencies like: verb-subject, verb-object and verb-prepositional phrase. In fact, the verb and the head of the subject, object and prepositional phrase are extracted and lemmatized. In the next step, the extracted lemmatized pairs are statistically weighted and only the

[32] Harris' distributional hypothesis is that words that occur in the same contexts tend to have similar meanings.

pairs over a given threshold are used for the Formal Concept Analysis. The result of the Formal Concept Analysis is a lattice which transformed into a compacted partial order returns the aimed concept hierarchy.

Cimiano et al. (2005) evaluate their approach from two perspectives. First, they evaluate the extracted concept hierarchy against two existing ontologies from the finance and tourism domain by using the evaluation method proposed by Maedche and Staab (2002). The evaluation method proposed by Maedche and Staab (2002) compares ontologies at three levels: semiotic, syntactic and pragmatic. Cimiano et al. (2005) use this evaluation method in order to measure the lexical and taxonomic overlap between the concept hierarchy extracted with their method and the ones existing in the reference ontologies. The calculated F-measures have shown that the proposed approach performs better for tourism (40.52%) than for the finance domain (33.11%).

The second evaluation Cimiano et al. (2005) perform, is in fact a comparison of their approach with the hierarchical agglomerative clustering[33] and Bi-Section-KMeans[34]. The results of this comparison have shown that the proposed approach by Cimiano et al. (2005) produces better results.

Gamallo et al. (2002) present a machine learning approach for the extraction of semantic relations from a Portuguese text corpus. The method relies on a unsupervised strategy for clustering semantically similar syntactic dependencies. More precisely, the approach allows for the clustering of those syntactic dependencies which introduce similar semantic relationships. The research by Gamallo et al. (2002) can be divided into three parts. In the first step, the syntactic dependencies are identified in the corpus and clustered in semantic groups. In the next step, the extracted clusters are mapped into semantic roles. In the last step,

[33] Hierarchical agglomerative clustering is a similarity-based bottom-up clustering technique in which at the beginning every term forms a cluster of its own.

[34] Bi-Section-KMeans is defined as an outer loop around KMeans

the clustered concepts and the relations between these concepts are used for a thesaurus design.

In order to apply the clustering algorithm on syntactic dependencies, Gamallo et al. (2002) tagged their corpus with a PoS tagger (Marques, 2000) and parsed it with a shallow parser (Rocio et al., 2001). Based on this linguistic analysis and a simple heuristics based on right association, Gamallo et al. (2002) select candidates for the dependencies. The candidate dependency between two words implies a relation, and the two words which are potentially related to each other. The two words are in fact the head nouns of the phrases detected by the parser, one being considered the head argument and the other one the complement argument of the relation. Fore example, from the expression *fase da evolução (phase of the evolution)* the candidate relation *de* between *fase* (in a head position) and *evolução* (in the complement position) is extracted.

Having the possible candidates for relation extraction Gamallo et al. (2002) apply the weighted Jaccard coefficient proposed by Gamallo et al. (2001) to measure the semantic similarity between word sets. Gamallo et al. (2002) assume that different arguments of a relation are semantically similar, if they require similar sets of words. This means that they measure the semantic similarity between all head and all complement positions in a relation by comparing their word distribution. The clustering algorithm returns for the relation introduced by *de* and having *fase* as a head the complement words *processo (process), evolução (execution), investigação (investigation), trabalho (work)* as extensional description of the semantic class required by the head *fase.* On the other hand, the algorithm clustered for the head position required by the complement *evolução* in the relation introduced by *de* the following words: *fase (pahse), momento (moment), período (period), resultado (result), fin (end).*

Based on the clustering results of the candidate relations, Gamallo et al. (2002) consider a candidate relation as a valid relation, if at least one of the two arguments (head or complement) are involved in the semantic clustering. For the example presented above, this means that the relation introduced by *de* between *fase* (in a head position) and *evolução* (in the complement position) is valid.

Having all possible relations from the corpus, in the next step Gamallo et al. (2002) define linguistic constraints on the relations in order to map them to thematic roles. The constraints imply morphology information, syntactic functions as well as information about the determiner of the argument filing the complement position in a relation. The result of this constraints is that for each generic relation, the arguments are mapped into thematic roles. So for example, for the relation introduced by *de* the head argument is mapped into the semantic role *possessed* and the complement argument is mapped into the role *possession.*

In the last step of their investigation Gamallo et al. (2002) build an thesaurus based on the clustering and semantic roles. A separate evaluation of the approach is not proposed, but the results of the clustering are integrated in two existing applications. The clustering results are introduced as semantic subcategorization patterns into the lexicon of the parser used above in order to correct false syntactic attachments proposed by the parser. The results achieved by Gamallo et al. (2002) are also integrated into an Information Retrieval (IR) system for the extension and improvement of documents recall.

3.5 Ontology Population

Ontology population (or knowledge markup) is concerned with the extensional part of a domain (A-Box), more specific with the instantiation of the concepts and relations already defined in the ontology. Similar to ontology learning, for ontology population we distinguish two approaches: rule-based approaches and machine learning approaches. We remark here that purely rule-based approaches always use linguistic analysis, whereas machine learning approaches do not always need linguistic analysis to accomplish their objective. We decided presenting here the following approaches because we consider them relevant when it comes to show that our method performs similar results with less effort.

3.5.1 Rule-Based Approaches

Suchanek et al. (2008) use rule-based patterns for ontology learning and population from Wikipedia's infoboxes (the English version) and category pages. The approach extracts candidates for classes and relations as well as instantiations of this classes and relations and is restricted to people, locations, institutions, companies and movies. From Wikipedia's structure of the infoboxes and category pages the extraction of intensional and extensional knowledge is achieved with regular expressions. A bigger challenge was to map the extracted knowledge into an ontology. For this purpose, the authors use here the taxonomic construction of WordNet in order to map the extracted classes and relations (and their instantiations) to an ontology. Although the method detects not only instantiations, but also new classes and relations, it remains mainly an ontology population approach. This is motivated by the fact, that the potential classes and relations are not derived on the basis of developed rules, but are just converted from Wikipedia' infoboxes and category pages by using string-based regular expressions. Still, the results are amazing: the 92 relations and 222391 classes are instantiated with 15 million relations and 1.7 million entities. The evaluation of the ontology is done manually by presenting to human judges a subset of the ontological knowledge extracted. The results of the evaluation confirms its quality, since precision returned 75%. The good precision is on the one hand motivated by the structured input data from Wikipedia and the determined set of classes and relations extracted, but on the other hand also on the restricted number of concepts and relations. Suchanek et al. (2008) decided to use their own representation model for the ontology, in order to be able to express n-ary relations while being decidable.

Navigli and Velardi (2008) present a linguistic-based approach for ontology population from Art and Architecture glossaries for the CRM CIDOC[35] cultural heritage core ontology. The population is restricted here on the instantiation of properties in the CIDOC ontology from English glossary definitions. This method can be described as follows. In the preprocessing step PoS tagging and NE recognition[36] is applied. The preprocessed text is then automatically annotated with relations from CIDOC ontology by using constraints on PoS, lexical chains, and constraints on domain and range of the ontology relations. The PoS constraints refer to a chain of words having specific PoS, such as: verb (V) preposition(P) noun(N). The lexical constraint requires that the lexical chain for the PoS chain above contains *composed of.* The semantic constraint on the domain and range of the existing ontology relation uses WordNet in order to extract, for example, from the existing ontology relation `is-composed-of` the domain *physical object* and the range *physical object.* The evaluation of the instantiation process was evaluated by comparing a partial set of the extracted instantiations with a manually built gold standard.

[35] http://cidoc.ics.forth.gr/

[36] The NE recognizer is used here for mapping a organization name to the general class *organization.*

3.5.2 Machine Learning Approaches

Maynard et al. (2008) present a bottom up approach for ontology population in the medical domain using both rule-based methods and machine learning. In the first step they present the TRUCKS system (Maynard and Ananiadou, 2000), a system for term recognition from English texts. The system uses contextual information of terms to measure semantic similarity between terms and candidate terms. For this purpose they use syntactic information (PoS tagging), terminological knowledge and the Unified Medical Language System (UMLS) semantic network[37]. By terminological knowledge the authors mean here a statistical metric to determine to what extent context information of a term is related to the respective term. The similarity between a term and its context is calculated by combining Example-Based Machine Translation (EBMT)-based techniques with techniques which measure semantic similarity. The method presented is classified as an ontology population method because, although not explicitly formulated, the extracted terms can be introduced as instantiations of ontology classes.

As an extension of their method Maynard et al. (2008) argue that their method can be easily applied for general Information Extraction (IE). In order to demonstrate that their approach is also technical realizable they are using GATE, the General Architecture for Text EngineeringCunningham (2002), respectively ANNIE (Maynard et al., 2002) and ANNIC (Aswani et al., 2005) .

In the last section the authors present work done in the area of Ontology-Based Information Extraction (OBIE) arguing indirectly that their method can be easily adapted to OBIE. This is motivated by the fact that the difference between IE and OBIE lies in the fact that traditional IE uses flat lexicons whereas OBIE uses formal ontologies. The authors argue that most of the presented information extraction systems (Kogut and Holmes, 2001; Ciravegna and Wilks, 2003)

[37] http://www.nlm.nih.gov/research/umls/

are ontology-oriented by using the ontology as their target output while ontology-based ones (Cimiano et al., 2004; McDowell and Cafarella, 2006; Kiryakov et al., 2004) use class and instance information during the information extraction process. The evaluation is performed by using the Balanced Distance Metric (BDM), a metric for the evaluation of the ontological classification, which uses similarity between the key (the gold standard) and response (the output of the system) instances in the ontology to determine the correctness of the extraction.

Tanev and Magnini (2008) present in their paper a method for ontology population of their own English named entity ontology with named entities from Wikipedia[38]. For their approach they use contextual similarity between a concept c in the ontology and a term t to be classified. For the experiments described in this paper they have chosen two named entity categories namely, geographical and person names. For each of this concepts ten subclasses were chosen for which based on lexical-syntactic features classification models are learned. For this purpose the corpus was parsed with the dependency parser MiniPar (Lin, 1998). Parallel to the syntactic models extracted from the corpus, training sets containing simple list of instances without context are built. The only condition for this list is that the instances in this training set have to appear at least twice in the dependency parsed corpus. For determining whether an instance can be introduced into the ontology the contextual similarity between the collected syntactic models and the training examples representing a certain class is determined. Tanev and Magnini (2008) evaluate their method by introducing the micro precision measure, micro recall measure and micro f-measure

Cimiano and Völker (2005) describes an unsupervised method for ontology population with named entities (NE) from English texts. The method relies on Harris' distributional hypothesis[39] as well as on the vector-feature similarity be-

[38] http://www.wikipedia.org/

[39] Harris' distributional hypothesis is that words that occur in the same contexts tend to have similar meanings.

tween each concept and its possible instantiation. With other words, a possible instantiation of a concept is assigned to the respective concept if its contextual similarity is as big as possible. The algorithm based on similarity vectors assigns to an instance, represented by a certain context vector `vi` the concept corresponding to the most similar vector `vc`. Figure 3.5 shows how the algorithm functions.

```
classify(set of instances I, corpus P, set of concepts C) {
  foreach c in C
    vc = getContextVector(c,t);
  foreach c in C
    doFeatureWeighting(vc);
  foreach i in I {
    vi = getContextVector(i,t);
    class(i)=maxarg sim(vc, vi);
    }
  return class;
}
```

Figure 3.5: The algorithm for computing similarity between a concept and its potential instantiation.

Cimiano and Völker (2005) show that using linguistically-based patterns for extracting concept vectors makes the method more efficient concerning both the evaluation and the efficiency of computing similarity. The evaluation is realized by calculating the learning accuracy (Maedche et al., 2002) and F-measure.

Pantel and Pennacchiotti (2008) present a minimally supervised bootstrapping algorithm for the extraction of semantic relations from English text and mapping them to an ontology. The presented approach is divided into the following three phases. In the first phase, also called pattern induction phase, based on the Hearst patterns (Hearst, 1992) seed instances of particular relations are used to extract generic patterns. For the induction of generic patterns Pantel and Pennacchiotti (2008) are using the slightly modified method of Ravichandran and Hovy (2002) replacing the instantiated concepts by labels. The new generated generalized patterns are then applied on the corpus. In the second phase, the general patterns are selected for further iterations by ranking them according to a new metric proposed here. The proposed metric uses the pointwise mutual information[40] (Cover and Thomas, 1991) weighted by the reliability of each instance. In the last phase, by using the selected patterns a set of instances is retrieved from the corpus. The retrieved instances are then mapped to WordNet and therefore two methods are proposed: clustering and similarity measure. The evaluation of this method is done by measuring precision, recall and f-measure after a gold standard was constructed manually. The average micro precision is calculated by dividing the correctly classified terms by the entire number of terms classified. The average micro recall is calculated by dividing the correctly classified terms by the entire number of terms and the micro F-measure is the result of the combination of the micro precision and the micro recall measure.

3.6 Conclusion

The state of the art in the field of ontology learning presented in section 3.4 can be pictured as follows. There are purely linguistic approaches (Aguado de Cea et al., 2008; Aussenac-Gilles and Jacques, 2008), linguistic approaches which use machine learning for generalization (Ciaramita et al., 2008) and machine learning approaches which use linguistic information (Cimiano et al., 2005; Gamallo et al., 2002). On the other hand, all approaches presented here are concentrating on discovering new relations, since this is indeed a challenging task. But some of the approaches are also concerned with discovering new concepts (Ciaramita et al., 2008; De la Clergerie, 2009; Cimiano et al., 2005). We notice here, that in the ontology learning field, there is no real synergy between linguistic-based approaches and machine learning approaches. The only connection between these

[40] This metric is used for measuring the strength of association between two instances.

two strategies is that machine learning approaches are using linguistically annotated text as a parameter, but the method per se remains a machine learning one.

The purely linguistic approaches (Aguado de Cea et al., 2008; Aussenac-Gilles and Jacques, 2008) presented above perform ontology learning on the basis of deep linguistic analysis, by activating a graphical interface controlled by the user for entering the extracted knowledge into the ontology. The method proposed in this thesis is also based on linguistic patterns, but different from Aguado de Cea et al. (2008) and Aussenac-Gilles and Jacques (2008) our method is a unsupervised method for ontology learning based on shallow and deep linguistic analysis. From that supervision perspective our method resembles most with the one presented by Ciaramita et al. (2008), which also propose an unsupervised method for ontology learning. Still, our method differs from that proposed by Ciaramita et al. (2008) by the fact that, we perform ontology learning from both shallow and deep linguistic analysis, covering this way a wider range of phenomena. Concerning the generalization of rules performed by the machine learning techniques applied by Ciaramita et al. (2008), we propose in this thesis also a set of linguistically derived generic rules. As mentioned, the generic rules are not the result of combining machine learning techniques with deep linguistic analysis as in Cimiano et al. (2005) and Gamallo et al. (2002). What we suggest instead is a set of generic patterns[41] manually derived from the corpus based on the encountered linguistic phenomena.

The investigations on ontology population presented in this chapter has shown that there is no clear line between rule-based and machine learning approaches (Pantel and Pennacchiotti, 2008; Cimiano et al., 2005). As described above, the ontology instantiation process covers in most of the cases both concept and relation instantiation (Suchanek et al., 2008), but it can also deal only with relation in-

[41] In machine learning language we would call them seeds.

stantiation (Navigli and Velardi, 2008; Pantel and Pennacchiotti, 2008) or even only concept instantiation (Tanev and Magnini, 2008; Cimiano et al., 2005; Maynard et al., 2008). Some of the described rule-based approaches use linguistic patterns (Navigli and Velardi, 2008), others regular expressions on linguistic annotation (Suchanek et al., 2008). Machine learning approaches are mostly concerned with instantiating concepts (Tanev and Magnini, 2008; Cimiano et al., 2005; Maynard et al., 2008), but we cannot say that relation instantiations is neglected (Pantel and Pennacchiotti, 2008).

On the other hand, as shown by Maynard et al. (2008) and Suchanek et al. (2008), there is no clear line between ontology learning and ontology population, ontology population being in the research presented by Maynard et al. (2008) closely related to term extraction and Suchanek et al. (2008) matches Wikipedia's infoboxes into an ontology.

Although we do not claim to present a better approach or a new approach for ontology population in this thesis, our method and results are comparable with those of the approaches presented in the section above. Since we perform ontology population as a welcomed side-effect of the shallow linguistic analysis, we noticed that out work fits with the approaches considered state of the art at this point.

Chapter 4

The Methodology for the Extraction of Patterns and Rules for Ontology Schema Components

In this chapter, we describe an incremental multi-layer rule-based methodology for the extraction of ontology schema components from German financial newspaper text. We concentrate on describing both the process of rule generation for the extraction of ontology schema components and the application of the developed rules. By *Extraction of Ontology Schema Components* we mean the detection of new concepts and relations between these concepts for ontology building. As described in Chapter 3, the process of detecting concepts and relations between these concepts corresponds to the intensional part of an ontology and corresponds to the ontology learning[1] process.

[1] Ontology learning is the process of semi-automatic support in ontology development (Buitelaar et al., 2005).

Most of the research on ontology learning (Hearst, 1992; Cimiano et al., 2005; Aguado de Cea et al., 2008) investigates the learning potential at sentential level, after the corpus has undergone a deep linguistic analysis[2]. In this thesis we present a bottom-up method for the extraction of ontology schema components, showing that the extraction of new classes and relations can be performed by using shallow and robust linguistic analysis, without directly applying deep linguistic analysis.

We start the investigation by extracting candidates for ontology classes and relations from plain text, by applying text- and string-based patterns. Then we go one step further and apply the accumulated knowledge from the previous step to semantically and Part-of-Speech (PoS) annotated text, validating candidates from the first step. In the last step, we complete the already constructed ontology with classes and relations extracted on the basis of deep linguistic processing, more precisely grammatical functions.

Section 4.1 describes the process of ontological rule generation from financial news corpus, whereas Section 4.2 shows how these rules can be applied to an enlarged corpus.

4.1 Text Analysis for the Generation of Extraction Rules

In this section, we present the process of rule generation for the extraction of ontology schema components. Since the aim of this thesis is to show that extraction of ontology schema components can also be carried out without directly applying deep linguistic processing, we start by investigating the potential for domain knowledge extraction from plain text. Although we expect that linguistic

[2] By deep linguistic analysis we mean grammatical function analysis.

knowledge alone (without the use of any linguistic tools) is not enough to extract full ontological knowledge, the plain text level is very important when it comes to define an anchor for the process of ontology learning. The analysis on the text level demonstrates how ontology learning can be performed in a quick and a ”dirty” way.

4.1.1 Detection and Analysis of Compound Words

For the work presented in this thesis we are using German texts for which we are taking certain specifics of this language into account. The first one is the fact that German text makes a heavy use of conflated compound words (this aspect is also shared with other languages, like Dutch, Finnish etc.). French and English have also compound words but very often in those languages the compounds are not merged into one string (*Prime Minister* vs. *Bundeskanzlerin*).

A first intuition guiding our investigation is the fact that German compound words are good indicators for the expression of relations between entities expressed by the elements of the compounded words. This intuition is also supported by German grammar studies. According to Erben (1993) the German determinative compounds[3] consist mostly of two elements. The second element, an adjective or a noun, is the main element which determines the grammatical (PoS and morphological features) and the conceptual class of the compound, whereas the first element, an adjective, a noun or a verb, is the determinative element which specifies the second element. Although German contains also copulative

[3] Determinative compounds are those compounds in which one element is subordinated to the other element of the compound, more precisely, one element determines/specifies the other element (Duden, 2006). In German the first element of the compound specifies the second one having as a result the hyponymy relation between compound and its second element: *Bundeskanzlerin* (*Prime Minister*) is specific type of a *Kanzler* (*minister*).

compounds[4], here we consider only determinative compounds.

For the detection of compounds we implemented a pattern-based approach and applied it to our German economic newspaper corpus. The pattern-based approach is quite straightforward: we first search for nouns in the corpus (for German, a string starting with a capital letter between blanks or between a blank and a punctuation sign). If, in a second search round, we can detect that such a noun item appears as substring in a larger noun, then we considered that we have found a compound. A further restriction was that the nominal items detected in the first round appear in the compound as a prefix or as a suffix. Nouns which have been identified as parts of a compound are *Chef* (*chief*), *Manager* (*manager*) and *Konzern* (*concern*) which appear, among others, in compounds like *Firmenchef* (*head of the firm*), *Finanzmanager* (*finance manager*) and *Konzernumsatz* (*the business volume of the concern*).

For the research presented in this thesis, we concentrate on binary `noun-noun` compounds. The identification of such compounds is made as follows: one of the compound elements has to be found in the list of all extracted nouns from the corpus, whereas the second element we look up in a German lexicon[5]. Concerning the German joint elements (Fugeelement), which may appear in compounds (such as *s* in *Wohnungsbau* (*house building*), they do not disturb the detection of compounds since the generated pattern looks for the two elements of the compound in the corpus, respectively in the lexicon.

The decision for taking only `noun-noun compounds` into account is motivated

[4] Copulative compounds are compounds were the elements are considered semantically coequal and which do not have a main element which specifies or determines the other element in the compound. This type of compounding is very seldom used in the German language (Lohde, 2006) and here they are not extracted yet. Copulative compounds are *Hosenrock* (*pantsskirt*), *Ofenkamin* (*stove chimney*), *Dichterkomponist* (*poet componist*)

[5] http://wortschatz.uni-leipzig.de/

on the one hand by the observations we make in our corpus[6] and on the other hand by the fact that binary `noun-noun` compounds are easier to be detected and interpreted with a shallow analysis.

The pattern returned compounds such as *Aktiengesellschaft* (*stock company*), *Bankensystem* (*banking system*), *Kursverfall* (*slump in prices*), *Notenbanken* (*central banks*), *Bankenvertreter* (*representative of the bank*), *Datenbanken* (*databases*).

Since the two elements of a compound, here represented by nouns acting as potential ontology classes, are connected to each other semantically (Fleischer and Barz, 1995; Lohde, 2006; Motsch, 2006), the task was to specify the relations between these two nouns.

4.1.2 Rules for the Extraction of Ontology Schema Components from Compounds

On the basis of the detection of compounds, we suggest the extraction rules in Figure 4.1 and Figure 4.2 for deriving potential T-Box elements. In this context we propose that the elements of a compound become ontology classes. Concerning the relations, we recognize two types: the structural type represented by the `subClassOf` relation (rendering the relation between the compound and its second element) or a relation denoting an `objectProperty`[7] (rendering the relation between the elements of the compound).

The two nouns, `noun1` and `noun2`, in Figure 4.1 and Figure 4.2 are potential

[6] We observed that the noun compounds covered by our pattern are mostly `noun-noun` compounds. The verbs appearing as the first element of the compounds are processed as nominalized verbs (*Vertriebschef* (*distribution chief*)) and the adjectives in first position in the compound are seldom. According to Lohde (2006) only 6% of the German compounds are `adjective-noun` constructions and for their detection we would need deeper linguistic analysis.

[7] Here we decided to use the OWL `objectProperty` notation for denoting a relation between two nouns.

ontology classes[8].

```
compound[noun1[suggestedClass] + noun2[suggestedClass]]
 ==> subClassOf(compound, noun2)
```

Figure 4.1: Deriving the `subClassOf` relation.

The rule in Figure 4.1 states that between a compound and the second noun of the compound there is a `subClassOf` relation. This relation is motivated by the definition of the determinative compounds which introduces hyponymy between the compound and its second noun. For example, from the compound *Bankenvertreter* we derive the relation: `subClassOf(Bankenvertreter, Vertreter)`, which translated into English means that *representative of a bank* is a `subClassOf` a *representative.*

```
compound[noun1[suggestedClass] + noun2[suggestedClass]]
 ==> objectProperty(noun1, noun2)
```

Figure 4.2: Deriving the `objectProperty` relation.

On the other hand, our intuition - sustained by the already existing analyses of the German compound (Fleischer and Barz, 1995; Lohde, 2006; Motsch, 2006) - was that there is also an additional relationship between the elements of a compound. Figure 4.2 describes this relation. The rules states that there is a relation between the two nouns of a compound. Applying this rule on the compound *Bankvertreter* we will have a relation between *Bank* (*bank*) and *Vetreter* (*representative*). From *Bankvertreter* we can extract the two relations depicted in Figure 4.3.

Obviously, the (naïve) processing strategy presented above is very general and the `objectProperty` relation is not really the most specific. In this context, we would like to specify more precisely the relation type between the elements of a

[8] A special case are the compounds constructed by appending two string with a hyphen, such as *Colonia-Konzern*. In order to analyze this type of construction we need both Part-of-Speech (PoS) information and lexical semantics. This is the reason why we deal with this constructions in Section 4.1.3.

```
Bankvertreter[Bank + Vertreter]
  ==> subClassOf(Bankvertreter, Vertreter)
  Bankvertreter[Bank + Vertreter]
  ==> objectProperty(Bank, Vertreter)
```

Figure 4.3: Example for the instantiation of the rules in Figure 4.1 and Figure 4.2.

compound and the simple existence of two nominal items is for this purpose not enough. So for example, in the case of *Aktiengesellschaft*, from which we can correctly derive `subClassOf(Aktiengesellschaft, Gesellschaft)`, we also want to derive `disposesOver(Gesellschaft, Aktie)`. In order to do so, we need contextual information and lexical semantics. By using contextual information we validate the relation between the elements of the compound and lexical semantics enables the specification of the relations between the two nouns.

In the next section, we present propose a method for specifying the already extracted generic `objectProperty` relation from compounds. We refine our heuristic rules for deriving T-Box components by searching for reformulations of the compounds in the corpus (see Section 4.1.3).

4.1.3 Detection and Analysis of Paraphrases for Compounds

After splitting the compound back into *noun1 + noun2*, we search for the paraphrases of all found compounds in our corpus. Our decision to look for the paraphrases of compounds is motivated by the fact that we assume that the elements of a compound are related semantically to each other and this fact becomes more evident when analyzing the paraphrases. Our assumption is also sustained by Lohde (2006) and Motsch (2006). Although they have two different methods for approaching this issue, the main idea is the same: the elements of a compound are semantically related to each other and this relation becomes visible in the paraphrase. The pattern we are looking for is *noun1* followed by at most three words and the *noun2* or *noun2*, followed by at most three words and *noun1*. We use this distance between the two nouns in a compound, because we assume that to determine the relation between the two nouns we do not need a larger ”window” than the one adopted by us. For each of the compounds found in the previous section, we looked for one of the patterns described in Figure 4.4.

```
firstCompoundElement + word{1,3} + secondCompoundElement
secondCompoundElement + word{1,3} + firstCompoundElement
```

Figure 4.4: Patterns for finding reformulated compounds.

Concerning the result of the extraction of the compounds and their paraphrases we observe that not all paraphrases are indeed semantically equivalent to the compound, but have been covered by the (purely string-based) patterns. So for example, for *Bankkunden* (*bank customer*), which was identified as a compound, the pattern returns the following reformulations: *Banken die Kredite*, *Banken eher bereit, Kredite*, *Banken die Kredite*, *Banken eher bereit, Kredite*. We do not consider any of the listed paraphrases as valid paraphrases, since this are not constructions from which we can deduce a relation between the nouns *Bank* and *Kunden*. Such paraphrases we filtered out and we kept only paraphrases like the ones listed in Table 4.1, from which we can deduce the relation between the two nouns in the paraphrase.

4.1.4 Rules for the Extraction of Ontology Schema Components from Paraphrases

By using information about PoS and lexical semantics, we can expand the extracted generic relation `objectProperty`. As a semantic resource we use Ger-

Compound	Paraphrase of Compound
Bankexperten	Experten der Bank
Bankmitarbeiter	Mitarbeiter einer deutschen Bank
Expertenschätzungen	Schätzungen von Experten
Bürofachmesse	Fachmesse für Büro
Westlöhne	Löhne im Westen
Designchef	Chef über deutsches Design
Umweltveträglichkeit	Verträglichkeit mit der natürlichen Umwelt

Table 4.1: Examples for valid paraphrases of compounds.

maNet (Kunze and Lemnitzer, 2002), more specifically the top semantic fields[9]. Until now, the nouns were considered only suggested classes for the ontology. By validating the relation between the nouns of a compound, we also validate the nouns as classes in the ontology. In this context, morphology becomes useful when it comes to solving the redundancy problem of the ontology classes. By using lemmas as classes, a noun appears just once in the ontology, and not as often as the number of its morphological variations. Additionally, by having PoS and lexical semantic information, we can also handle the compounds consisting of a named entity (NE)[10] and a common noun.

Concerning the rule extension of the `objectProperty` relation defined in Section 4.1.1, we looked at the extracted paraphrases and discovered two types of paraphrases. The first type is the paraphrase in genitive. The generic rule for finding all genitive is depicted in Figure 4.5.

```
noun1[ontologyClass] + string[PoS=article/pronoun, case=genitive]
+ modifier{0,2} + noun2[ontologyClass]
==> ontologicalRelation(noun2, noun1)
```

Figure 4.5: Rule for genitive paraphrase of compounds.

[9] GermaNet provides the following top semantic fields for nouns: artifact, attribute, shape, feeling, body, cognition, communication, motive, food, object, phenomenon, plant, substance, time, animal, state, act, process, person, group, possession, relation, attribute, event, quantity, location.

[10] Such as the names of persons, organizations, locations, quantities, monetary values.

This pattern covers genitive phrases such as *Aktien der multinationalen Gesellschaft* (*shares of multinational corporations*) or *Pflichten des Kunde* (*responsibilities of a customer*) and is the basis for developing the extraction rules. Based on GermaNet's semantic affiliation of the peripheric nouns in the paraphrase, we discovered the following six relations: `hasPosition`, `disposesOver`, `hasDimension`, `hasAttribute`, `hasEvent` and `hasLocation`. A detailed description of these rules is given in Chapter 5. For example, for the compound *Aktiengesellschaft* we found the reformulation *Aktien der Gesellschaft*, where *Aktien* was semantically classified as belonging to GermaNet's semantic class *Possession* and *Gesellschaft* has been classified as belonging to GermaNet's semantic class *Group*. The analysis of genitive constructions, where the nouns are identified as belonging to the more general semantic classes *Possession* and *Group*, generated the relation `disposesOver(Gesellschaft, Aktien)`.

By using GermaNet's semantic classification, we enable the structural integration of the newly discovered classes and relations into the ontology. For example the noun *Aktien* has been identified as belonging to the semantic class *Possession*. Having the nouns classified in semantic classes, these semantic classes are also introduced as superclasses into the ontology. In this way, we have a limited number of superclasses which have nouns in paraphrases as subclasses. Following this, both *Aktie* and *Possession* become classes in the ontology and *Aktie* will become a subclass of *Possession*.

The second type of paraphrase pattern concerns the paraphrases with prepositions. As for the genitive phrases, the generic `objectProperty` becomes a more specific relation depending on the lexical semantics of the two nouns in the paraphrase. We are aware of the fact that the prepositions themselves carry semantic information, which is not always determined. By using the semantics of nouns we can determine more exactly the type of ontological relation to be extracted. Figure 4.6 contains the generic rule for the extraction of ontological knowledge from this type of paraphrases.

```
noun1[ontologyClass] + string[PoS=preposition]
+ modifier{0,2} + noun2[ontologyClass]
==> ontologicalRelation(noun2, noun1)
```

Figure 4.6: Rule-pattern for deriving classes and relations from paraphrases of compounds using prepositions as links between the original nouns of the compounds.

The pattern above covers paraphrases such as *Autos aus inländischer Produktion* (*cars from domestic production*) or *Löhne im Westen* (*salaries in the west*). Analyzing this type of paraphrases, we discovered a set of seven rules for the extraction of ontological relations. From those six relations, five we also discovered during the analysis of genitive phrases: `disposesOver`, `hasDimension`, `hasAttribute`, `hasEvent`, `hasLocation`. Only one relation is new: the `hasAffiliation` relation. As for the genitive paraphrases, the rules for the extraction of ontological knowledge from prepositional paraphrases are based on GermaNet's semantic classification of the nouns occurring in the paraphrase. For example, the compound *Millionenverlust* (*million loss*) has been paraphrased as *Verlust von 100 Millionen* (*loss of 100 millions*). GermaNet classifies *Verlust* as *event* and *Millionen* as *quantity* and based on the analysis of the prepositional paraphrases we extract here the ontological relation `hasDimension(Verlust, Millionen)` with domain *Event* and range *Quantity*.

An important aspect here concerns GermaNet and the additional lexical semantic information it provides: the synonyms, antonyms, hyponyms and meronyms. If provided by GermaNet, for each noun occurring as argument of a relation we use, if available, GermaNet's synonyms, antonyms, hyponyms and meronyms. The synonyms will enter the ontology as class labels, the antonyms will introduce the `isOppositeTo` relation, the hyponyms will become subclasses and the meronyms introduce the `partOf` relation. For the noun *Verlust* GermaNet provides the antonym *Gewinn* and therefore `isOppositeTo(Verlust, Gewinn)` with domain and range *Event*. Besides this antonym, GermaNet also provides a set of hyponyms such as *Pleite* (*bankruptcy*) and *Ruin* (*ruin*), which will enter the ontology as subclasses of *Verlust*.

4.1.5 Detection and Analysis of Modification Phenomena

In the process of detecting paraphrases, we observed that many of the paraphrases contain modifiers. Based on this observation we concentrated on developing extraction rules for the extraction of ontological knowledge from modification phenomena. In order to determine the type of ontological relation that can be extracted from the structure modifier(s)-nominal head (such as *multinationale Gesellschaft* (*multinational corporation*)), some components of the structure had to be viewed from a lexical semantic point of view.

We consider here adjectives and adverbs, and apply to them various language specific classification schemes. For adjectives we used the classification by Lee (1994) and for adverbs the classification by Lobeck (2000)[11]. As for nouns, the semantic classes to which the adjectives and adverbs belong, are introduced into the ontology as superclasses. Based on this classification we introduce new relations between the modifier(s) and the noun they modify[12].

[11] We use for modifiers these semantic classification because they are more fine-grained than GermaNet's classification and we can easily add new adjectives and adverbs to it.

[12] This noun is in fact the nominal head of a nominal or prepositional phrase. But since we are not using any linguistic processing PoS is our indicator for finding a premodified noun.

4.1.6 Rules for the Extraction of Ontology Schema Components from Modification Phenomena

Having the paraphrase *Aktien der deutschen Gesellschaft* (*shares of the German corporation*), the extraction rule will return the following relation: `hasAffiliation(Gesellschaft, Deutsch)`. Many of nouns appearing in paraphrases are modified by just one modifier. But there are cases in the corpus (which are not covered by the paraphrase pattern) when a noun is preceded by more than one modifier. For multiple premodifiers which are not separated by any punctuation sign or conjunction to each other we speak of an aggregation of adjectives. For example for *großen deutschen Konzern* (*big German concern*), the first premodifier in the token chain modifies the remaining phrase (Zifonun et al., 1997). This way we extract `hasAffiliation(Konzern, Deutsch)` and `hasDimension(Deutscher Konzern, Groß)`.

A different principle applies for modifiers connected by punctuation signs or/and conjunctions: each premodifier introduces a relation between itself and the noun it modifies (Zifonun et al., 1997). From *Kleinen, Krisengeplagten Firmen* (*small, crisis affected firms*) we extract `hasDimension(Firma, Klein)` and `hasMode(Firma, Krisengeplagt)`.

A more specific case is represented by the modification of adjectives by adverbs such in *sehr großes Gehalt* (*very big salary*). In this case the adverb *sehr* modifies the adjective *großes* and not the phrase *großes Gehalt* (Duden, 2006): `hasAspect(Groß, Sehr)`, `hasDimension(Gehalt, Groß)`.

Since modification is a very powerful linguistic phenomenon with a high coverage in the corpus, the three extraction rules enounced by us above, cover 26 relations, depending on the semantic class of the modifier. This 26 relations are generated based on the semantic classification of adjectives and adverbs. Lee (1994) introduces 24 semantic classes for adjectives from which we use only 22[13] to introduce new relations into the ontology. Appendix A.2.3 lists the twenty four semantic classes for adjectives and the corresponding relations they introduce into the ontology.

The same principle applies also for adverbs. Lobeck (2000) classifies the adverbs into 13 classes from which we use only nine[14] in order to introduce new knowledge into the ontology. Appendix A.2.4 contains all thirteen semantic classes for adverbs and their correspondence to ontological relations derived from them. Because some of the semantic classes for adverbs overlap with those for adjectives, we introduce here only five new ontological relations. As for the compounds, we use here all semantic information provided by GermaNet, by introducing into the ontology all synonyms, antonyms, hyponyms and meronyms of nouns.

4.1.7 Detection and Analysis of Named Entities

Because the pattern for discovering compounds in Section 4.1.1 covers only conflated compounds, we aim to extract ontological information also from constructions of the type *US-Konzern* (*US concern*) or *Basf-Chef* (*Basf chief*) consisting of a NE connected with a hyphen to a common noun. Of course, we could enounce the subclass rule (*Colonia-Konzern* is a `subClassOf` *Konzern*), but we would rather extract more detailed information. For this type of compounds, we generated a pattern which detects hyphen compounds such as *Zeiss-Stiftung* (*Zeiss foundation*) and *AXA-Gruppe* (*AXA group*). Besides hyphen compounds, we also aim to analyze appositional constructions such as *Geschäftsführer Karl-Ulrich Kuhlo* (*manager Karl-Ulrich Kuhlo*) and *Professor Gerhard Schmitt-Rink*

[13] We do not take into consideration the classes *Objective plus Temporal Combination* (*verkehrsschwache Zeiten*) and *Objective plus Locative Combination* (*einsamer Ort*) because they go beyond the word level.

[14] We do not take into consideration the pronominal (*darüber*) and relative pronouns (*wann*) because they do not appear as modifier.

(*professor Gerhard Schmitt-Rink*). As for the hyphen compounds, the generated pattern extracts this kind of construction in order to integrate the newly acquired knowledge into the ontology.

4.1.8 Rules for the Extraction of Ontology Schema Components from Named Entities

Depending on the semantic class of the second element in the hyphen compound, we extract the following ontological knowledge. From the *AXA-Gruppe* example presented above we enunciate `instanceOf(AXA, Group)` and from *Basf-Chef* we extract `instanceOf(BASF, Group)` and `hasPosition(BASF, Chef)`. In fact, our extraction rules cover only these two cases when the second element in the hyphen compound belongs either to GermaNet's semantic class *Group*, or to *Person*.

The extraction rules for appositions cover only the case when a noun, semantically classified by GermaNet as belonging to *Group* is followed by a name. For example, from *Konzernchef Ackermann* (*concern director Ackermann*) our rule extracts *Ackermann* as an instantiation of *Konzernchef*. In this way, we show that we perform, as a side effect, ontology population.

As before, for each noun appearing as an argument in a relation, we use GermaNet's information about synonyms, antonyms, hyponyms and meronyms.

4.1.9 Detection and Analysis of Grammatical Functions

Although we generated many extraction rules based on the shallow linguistic analysis, we are aware that the sentential level is an additional resource for thr extraction of ontological information. Our assumption is that on the sentential level, by using predicate-argument structures, we can cover other phenomena than the ones already found before. In order to be able to analyze predicate-argument structures we parsed all sentences containing a paraphrase. We decided to select only those sentences because we considered they carry enough linguistic information for the analysis of predicate-argument structures.

The analysis of the extracted sentences has shown that there is potential for extracting ontology schema components, but no patterns could be identified. By this we mean, that no often occurring pattern could be discovered. As a consequence of this fact, we decided to enlarge our set of sentences with sentences selected arbitrarily from the corpus. Additionally, we also decided to use lexical semantics, more specifically to use the semantics of the verb. As a lexical semantic resource we use the verb classification by Schumacher (1986). The analysis of this set of sentences allowed us to identity patterns for extracting ontological information from predicate-argument structures.

4.1.10 Rules for the Extraction of Ontology Schema Components from Grammatical Functions

By using the grammatical functions we developed a set of rules for extracting the arguments of specific verbs in the corpus. In this way we introduce new relations such as `isa`, `cause`, `hasPossession`, `partOf`[15]. Example 1 depicts a sentence on which we performed extraction. Here the verb *sein* (*be*) connects the subject *Papierherstellung* (*paper production*) and the object *kapitalintensiven Branche* (*capital-intensive branch*) of the sentence. In fact the rule states that only the nominal heads of the phrases identified as subject and object enter the ontology[16] and therefore we extract `subClassOf(Papierherstellung, Branche)` with domain *Event* and range *Group*. Additionally, for each of the two nouns we use GermaNet's information about synonyms, antonyms, hyponyms and meronyms.

[15] A complete description of all the rules is given in Chapter 5.

[16] Kapitalintensiven will enter also the ontology, but we concentrated here on the ontological relations extracted from predicate-argument structures.

(1) Die Papierherstellung ist zu einer extrem kapitalintensiven Branche geworden.

Paper production evolved to be a very capital-intensive branch.

As described above, most of the discovered relations were developed on the basis of our observations when analyzing the corpus. We are aware of the fact that the extracted relations are not exhaustive[17], but this was not our ambition here. Our aim was to present a multi-layer, rule-based approach for the extraction of ontology schema components and to show that a lot of the ontological knowledge can be extracted without using exclusively grammatical functions.

4.2 Application of the Rules for the Extraction of Ontology Schema Components on the Entire Corpus

In Section 4.1, we described the rule finding process for the detection of ontology classes and relations. In this section, we describe the application process for the generated rules. Concerning the corpus on which the rules are applied, we use the same corpus as the one used for the generation of the rules for ontological extraction. By the fact, that for the rule generation process we used specific constraints, only a small part of the corpus was "used". In the rule application process, we do not have any constraints concerning the corpus, which means that the rules will be applied on the mostly unseen corpus. Another aspect which is different from the rule generation process concerns the annotation of the corpus. The generated rules are applied on a linguistically annotated corpus, more specifically on a corpus annotated with PoS and morphology. The use of a

[17] We have to remark here the fact that the developed and applied rules are not exhaustive and cover only the phenomena observed in this corpus.

annotated corpus for the application process is motivated on the one hand by the fact that nowadays to use a PoS and morphological analyzer is the state of the art in computational linguistics and does not “cost” much. On the other hand, by using a annotated corpus, we are able to refine the resulted ontology. For example, by using the lemma as a class, and not the noun we are able to reduce the number of classes in the ontology to the number of unique lemmas.

The following subsections describe not only rule application, but also describe the range of phenomena covered by the developed rules. How the different relations are modeled in OWL is described in detail in Chapter 6.

4.2.1 Application of Compound Rules on the Entire Corpus

The first linguistic unit on which we apply the developed rules are the compounds. By applying our rules extracted from paraphrases directly on compounds allows us to also extract relations from all compounds in the corpus, even from those compounds which do not have paraphrases in the corpus. Therefore, we first have to extract all compounds from the corpus. This is an easy task, since our morphological analyzer marks all compounds with a feature called `COMP`. Example (2) shows the analysis we have at hand for the noun *Bankmitarbeiter* (*bank employees*). Important for us here is the information about PoS, here noun, and the feature `COMP`. This feature contains the elements of the compound (here *Bank* (*bank*) and *Mitarbeiter* (*employee*)).

(2) <W COMP=”[bank mitarbeiter]” INFL=”[20 21 23 17 18 19]”
ORD=”3” POS=”1” STEM=”mitarbeiter” STTS_POS=”NN”
TC=”22”>Bankmitarbeiter</W>

As mentioned in the previous section - based on the compound definition (Erben, 1993) - the compound itself is a hyponym of the second noun of the compound. This type of relation is in fact a `subClassOf` relation. Applying this rule on our example we extract `subClassOf(Bankmitarbeiter, Mitarbeiter)`. Both *Bankmitarbeiter* and *Mitarbeiter* then become this way classes in the ontology, if not already there. Note that we do not introduce the string itself into the ontology. Instead, we use the information about the string's lemma, which in our annotation is marked by the feature `STEM`. In this way we exploit the fact that we are working with a morphologically annotated corpus and reduce the class redundancy. This strategy works fine for *Mitarbeiter*. Obviously for *Bankmitarbeiter* we cannot take the lemma as a class, since we want to have *Bankmitarbeiter* and not *Mitarbeiter* as a class. Therefore, for compounds we reconstruct the compound by concatenating the components of the `COMP` feature. This makes it possible to have for every morphological variation of *Bankmitarbeiter* the noun *Bankmitarbeiter* as a class.

In order to determine the relation between the two nouns of a compound, we annotate each noun in a compound with GermaNet. For the *Bankmitarbeiter* example, GermaNet identified *Bank* as belonging to the semantic class `Group`. On the other hand *Mitarbeiter* belongs to the semantic class `person`. By applying the corresponding extraction rule we introduce here `hasAffiliation(Mitarbeiter, Bank)` having *Person* as domain and *Group* as range, since the nouns *Bank* and *Mitarbeiter* were detected as belonging to the semantic class *Group* and *Person*. The nouns *Bank* and *Mitarbeiter* become subclasses of the GermaNet semantic classes *Group* and *science*: `subClassOf(Bank, Group)` and `subClassOf(Mitarbeiter, Person)`.

We also use GermaNet for determining, if marked in GermaNet, the synonyms, antonyms, hyponyms and meronyms of an identified ontology class. For the noun *Bank*, GermaNet returns as synonyms a set of nouns (such as *Geldinstitut*, *Kasse* and *Geldhaus*), which we introduce in the ontology also as labels of the class *Bank*. The hyponyms for the noun *Bank* are, among others, *Handelsbank* (*commercial bank*), *Hausbank* (*house bank*), *Landesbank* (*Land bank*). These we introduce in the ontology as subclasses of *Bank*.

4.2.2 Application of Paraphrase Rules on the Entire Corpus

As described in the previous section, the generated set of rules can be applied on genitive and prepositional phrases. Since all paraphrases used for rule generation are in fact phrases, we use from now on the term phrases for paraphrases. In order to apply the generated rules, we first determine (based on the developed patterns) all possible phrases on which our extraction rules can be applied. These phrases (which were already annotated with morphology and PoS) are then semantically annotated with GermaNet in order to apply the extraction rules.

On all extracted phrases, we apply the generated extraction rules[18]. As already mentioned, each of the eight relations has a domain and range. Additionally, for each noun in the relation, we introduce the (if available) synonyms, antonyms, hyponyms and meronyms into the ontology.

The seven extraction rules described in Section 4.1 and applied here use the PoS information and GermaNet's semantic classes and introduce the following relations: `hasPosition`, `disposesOver`, `hasDimension`, `hasAttribute`, `hasEvent`, `hasLocation` and `hasAffiliation`. A detailed description of this rules is given in Chapter 5. By applying the `hasLocation` relation, we match phrases like *Löhne im Westen* (*salaries in the west*) from which we extract `hasLocation(Lohn, West)` with domain *Possession* and range *Location*. Since the rule is applied on lemmatized text, we use the lemmas as classes and not the strings. Part of this

[18] The rules are of the form described in Figure 4.5 and Figure 4.6.

rule is also the fact that *Löhne* (*salaries*) and *West* (*west*) enter the ontology as subclasses of the more generic classes *Possession* and *Location*. Also part of each rule is the fact that for each of the two nouns we introduce into the ontology their synonyms, antonyms, hyponyms and meronyms.

The `hasPosition` rule matches phrases like *Chef des deutschen Chemiekonzerns* (*head of the German concern*) from which we extract `hasPosition(Konzern, Chef)` with domain *Group* and range *Person*. In this case the noun *Chef* enters the ontology as a subclass of *Person*, whereas *Konzern* enters the ontology as a subclass of *Group*[19].

4.2.3 Application of Premodification Rules on the Entire Corpus

Based on our observations from the phrases, we developed four extraction rules for the premodification phenomenon. These rules apply when a noun of a phrase is modified by one or more adjectives. The rules for premodification can be explained as follows. If we find in the corpus a noun which is modified by one or two adjectives or an adverb followed by an adjective, then we employ a linguistic construction from which we can extract a relation. The type of the relation depends on the semantic class of the modifier. For adjectives, we used the classification of Lee (1994), which classifies the adjectives into 24 semantic classes. For adverbs, the classification by Lobeck (2000) contains 13 semantic classes.

The rules generated for premodification cover phrases like *weltweit agierende Konzern* (*worldwide acting concern*), *deutscher Konzern* (*German concern*) but also *deutsche und amerikanische Autokonzerne* (*German and American car concerns*) and *sehr günstige Konditionen* (*very advantageous conditions*). Taking

[19] At this stage we ignore the adjective *deutsch*, since premodification phenomena are discussed in the following paragraphs.

for example the very simple phrase *deutscher Konzern* (*German concern*), the result of the extraction rule here is: `hasAffiliation(Konzern, Deutsch)` with domain *Group* and range *affiliation*. For *Konzern*, as for all classes introduced in the ontology, we are gathering all semantic information provided by GermaNet. On the other hand, *deutsch* will become a class in the ontology. Since *deutsch* was classified by Lee (1994) as an *adjective of affiliation*, the class *Deutsch* will become a subclass of the class *affiliation*. As for nouns, we use also for adjectives and adverbs all semantic information provided by GermaNet. For *deutsch* GermaNet provides a list of hyponyms which become subclasses of the class *Deutsch*. Because we want to also represent intensional modifiers[20], we will use reified relations to transform relations into classes. By reified relations[21] we mean that relations such as `hasAffiliation` are transformed into classes such as *affiliation-Relation*. This type of reification is different from that in RDF which implies the RDF construct `rdf:Statement`.

The necessity for using reified relations becomes obvious when applying the extraction rules on phrases like *ehemaliger Manager* (*former manager*). Here the intensional adjective *ehemalig* (*former*) makes it difficult to interpret *ehemaliger Manager* as a *Manager*, since now this person is not a *Manager* any more. By using reified relations we express the relation *temporalRelation* as the class *TemporalRelation*. Since *ehemalig* was classified as a *temporal* adjective it will enter the ontology as a subclass of the class *Temporal*. On the other hand, *Manager* semantically classified as a *person*, will enter the ontology as a subclass of the class *Person*. Because the formalization of the reified relations is described in detail in Chapter 6, we do not go into more details here.

The application of the generated rules for premodification cover the simple modification phenomena such as *große Firmen* (*big companies*), but also phrases in

[20] A modifier is intensional if, in its modification of a domain, it makes essential reference to the characteristics that comprise the denotation of the domain modified (Frawley, 1992)

[21] http://www.w3.org/TR/swbp-n-aryRelations/

which the noun is modified by more than one adjective connected to each other by conjunctions and/or comma: *deutsche und japanische Unternehmen* (*German and Japanese companies*). The extraction patterns also match constructions such as *weltweit agierende Konzern* (*worldwide operating concern*) where the first adjective *weltweit* modifies the phrase *agierende Konzern* and not *Konzern* (Zifonun et al., 1997). Another construction matched by our extraction rules is *sehr günstige Konditionen* (*very good terms*), where the adverb *sehr* modifies the adjective *günstig* and not the noun *Konditionen* or the phrase *günstige Konditionen* (Zifonun et al., 1997). No matter what kind of relation is extracted, for each new class introduced into the ontology we use, if available, its lemma as the class name. As for the other phenomena described above, we use here the full ontological power of GermaNet and introduce synonyms, antonyms, hyponyms and meronyms into the ontology.

4.2.4 Application of Named Entities on the Entire Corpus

As described in Section 4.1.7, during the compound analysis we also discovered compounds consisting of a NE which are connected with a hyphen to a common noun. Based on this type of compounds, we generated two extraction rules for the hyphen compounds. Also closely connected to NE's is the generated rule for appositions. We have to notice here that our annotation tool SProUT[22] augments the existing annotation with NE recognition. The application of the generated patterns on the annotated corpus makes it possible to cover more NE's, since NE recognition allows us to identify organizations and persons, in addition to locations, money, quantity and temporal units.

The first rule is concerned with the hyphen NE's. By applying the rule for hyphen NE's we match constructions such as *Metro-Konzern* from which we

[22] http://sprout.dfki.de/

extract `instanceOf(Metro, Konzern)`.

An example of how the rules apply can be shown on the apposition *Chemiekonzern BASF* (*the chemical corporation BASF*). Example (3) shows what information we have at hand for this phrase. The linguistic tool used for annotating the corpus recognized *BASF* as *Konzern*. Having all this information, our rule instantiates the class *Chemiekonzern* with *BASF*.

(3) <NE DESCRIPTOR=”konzern” ORGNAME=”BASF
ORGTYPE=”org-type” TC=”52” TYPE=”ne-organization”>
<W POS=”1” STTS_POS=”NN”>Chemiekonzern</W>
<W POS=”25” STTS$_P$OS=”NE”>BASF</W></NE>

The rule described above introduces, based on NE recognition, more specialized information about the noun *Konzern*. Until now, the noun *Konzern* was existing in the ontology as a subclass of the class *Group*. By using the information provided by the attribute `TYPE` (see example 3), the we introduce the class *Organization* as a subclass of *Group*. In this way we identify *Konzern* as an *Organization* and modify the existing class structure in the ontology. Actually, we are making it more specific.

4.2.5 Application of Grammatical Functions Rules on the Entire Corpus

In the section above we described our method for extracting ontological knowledge at word and phrase level. By applying these rules, we covered relations between the elements of compounds and between the elements of genitive and prepositional phrases. Evidently, a lot of information could be extracted this way. By using deep linguistic analysis, we expect to semantically connect bigger linguistic units, here phrases, with each other. In order to achieve this, we apply our developed extraction rules on the parsed corpus. The corresponding extraction rules for extracting ontology schema components are based on the predicate-argument structure of the sentence. In this way, depending on the verb and the grammatical function of its arguments, we extract different relations.

For example from the sentence in Example (4) we will extract the relation `obtainedRelation(Unternehmen, Darlehen)` filling the generic relation `obtain(SUBJ, OBJ)`. The nouns *Unternehmen* (semantically identified as belonging to the *Organization* class) and *Darlehen* (semantically identified as belonging to the *Possession* class) enter the ontology as subclasses of the more generic classes *Group* and *Possession*. For the nouns *Unternehmen* and *Darlehen* we introduce into the ontology all semantic relations provided by GermaNet.

(4) Nur [große Unternehmen][gf=subj] haben Phare-Darlehen[gf=dobj] erhalten.

Ony big companies have received Phare credit.

In example 5, we can see that there are also verbs which have more than two arguments and for which we need reified relations in order to represent the relations in OWL. The verb *verdienen* (*earning*) has as arguments the subject *Der größte deutsche Chemiekonzern* (*the biggest German concern*), the direct object *17 Millionen* (*17 Million*) and the indirect object *in den ersten neun Monaten* (*in the first nine months*). The relation we extract here is `earningRelation(Chemiekonzern, Million, Monat)` instantiating the generic relation `earn(SUBJ, DOBJ, IOBJ)`. We have to notice that the relation connects the head noun of the phrases and not the whole phrase. The relations between the modifiers and the head nouns are derived by the extraction rules presented in the previous sections.

(5) [Der größte deutsche Chemiekonzern BASF][gf=subj] verdiente [in den ersten neun Monaten][gf=pp_adjunct] [17 Millionen][gf=dobj].

> *The biggest German chemical concern earned in the first nine months 17 millions.*

Here too we introduce the lemma as class and add the additional semantic information provided by GermaNet.

In Figure 4.7 we sketch the relations extracted from the last sentence for the noun *Konzern*. In this way we cover relation extraction from compounds, phrases, NEs and sentences.

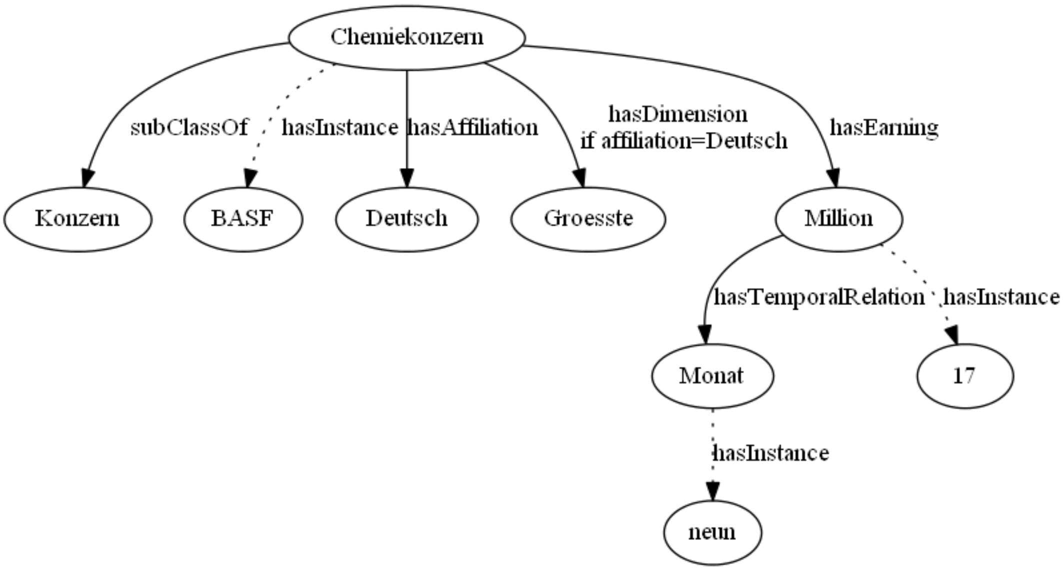


Figure 4.7: Graphical representation of the relations for the noun *Konzern* from sentence 5.

4.3 Conclusion

In this chapter we present the methodology applied for the process of extracting ontology schema components from German financial newspaper text. We have shown that the process of extracting ontology schema components consist of two main parts: the construction of the rules and the application of these rules. Based on the assumption, that shallow linguistic analysis provides enough information for extracting ontological knowledge, we present here a bottom-up method for the extraction of ontology schema components. We do not argue that for building an ontology shallow linguistic analysis is enough. What we want to show is that much of the ontological knowledge can be extracted more easily and faster than by using grammatical functions. Moreover, the ontology extraction rules generated from grammatical functions round out the set of rules. Section 4.1 describes every single step in the process of developing extraction rules from text by using lexical semantics.

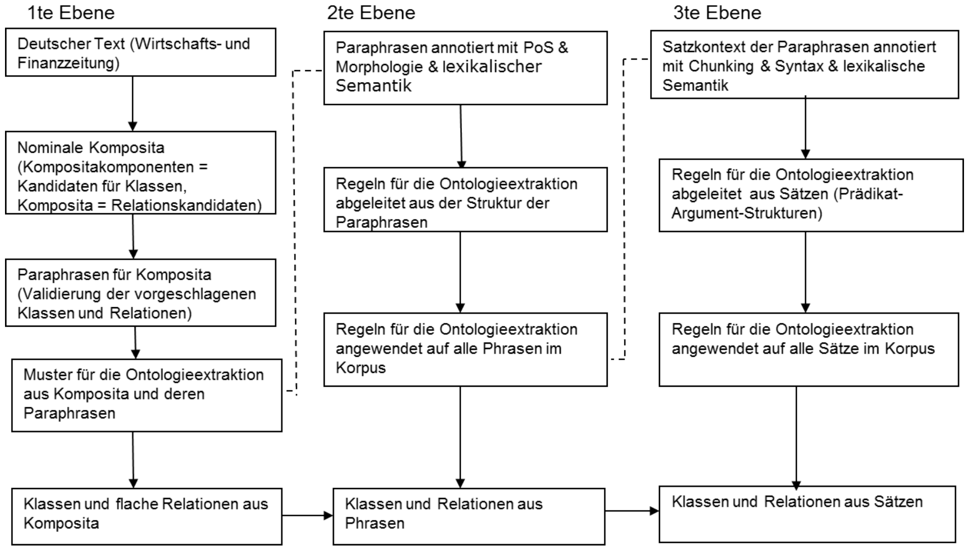

1te Ebene
2te Ebene
3te Ebene
Deutscher Text (Wirtschafts- und Finanzzeitung)
Nominale Komposita (Kompositakomponenten = Kandidaten für Klassen, Komposita = Relationskandidaten)
Paraphrasen für Komposita (Validierung der vorgeschlagenen Klassen und Relationen)
Muster für die Ontologieextraktion aus Komposita und deren Paraphrasen
Klassen und flache Relationen aus Komposita
Paraphrasen annotiert mit PoS & Morphologie & lexikalischer Semantik
Regeln für die Ontologieextraktion abgeleitet aus der Struktur der Paraphrasen
Regeln für die Ontologieextraktion angewendet auf alle Phrasen im Korpus
Klassen und Relationen aus Phrasen
Satzkontext der Paraphrasen annotiert mit Chunking & Syntax & lexikalische Semantik
Regeln für die Ontologieextraktion abgeleitet aus Sätzen (Prädikat-Argument-Strukturen)
Regeln für die Ontologieextraktion angewendet auf alle Sätze im Korpus
Klassen und Relationen aus Sätzen

Figure 4.8: Graphical representation of the processing pipeline.

We have shown that only based on PoS and semantic annotation we were able to develop extraction rules from compounds, from paraphrases of these compounds and from premodified nouns. In this way we are able to incrementally generate ontology extraction rules, which can then easily be applied on a different corpus (annotated or not). The set of text-based extraction rules is enlarged by the ontology extraction rules on the basis of grammatical functions. A welcome side effect of our approach is the instantiation of the ontology classes with persons, organizations, money, quantity, temporal units and locations. The process of ontology rule generation is an incremental theory-neutral process, which allows the application of the generated rules on arbitrary free text or linguistically annotated text. Figure 4.8 depicts the multi-layer process for the generation of ontology extraction rules.

The application of the generated rules is described in Section 4.2. Based on examples, we have shown how powerful the developed rules are, extracting all ontological knowledge available at word, phrase and sentence level. By applying the ontology extraction rules on linguistically annotated text, we succeed on the one hand to minimize the number of classes in the ontology and to increase on the other hand the number of instances in the ontology. In summary, by saying that in this chapter we have described a multi-layer, pattern-based approach for the extraction of ontological knowledge from text.

Chapter 5

The Rules and their Application for the Extraction of Ontology Schema Components

In this chapter, we describe in detail the development and application of the extraction of ontology schema components from Chapter 4. The main goal of the work described in this thesis is to show that ontology learning can be performed on the basis of shallow and robust linguistic analysis. In this chapter, we concentrate on showing how, from different linguistic knowledge encoded in text, we can extract ontology schema components. For our investigation we are using a corpus of German financial newspaper text, more precisely the 1992 edition of the German newspaper "Wirtschaftswoche"[1]. The decision to use financial newspaper text is motivated by two facts: the domain and the results of the extraction of ontological knowledge from this domain. Because the MUSING ontologies belong to the financial domain, our aim was to achieve comparable results with the MUSING ontologies. By using the 1992 edition of the German newspa-

[1] The corpus consists of 200107 tokens and 11583 sentences

per “Wirtschaftswoche” as corpus, we are able to extract ontological knowledge from the financial domain and to compare the results with those in the MUSING project. Concerning the annotation of the chosen corpus, we incrementally annotate the corpus with PoS, morphology and grammatical functions.

Since for big corpora the use of deep natural language processing strategies are time consuming and prone to different types of errors in the analysis (e.g. Grammatical Function (GF) detection), we suggest a multi-layered approach, which starts with the analysis of certain lexical properties of compound words and phrase expressions. In the next processing stage, we use PoS and morphological analysis, before using, in the last processing stage, information about grammatical functions. The motivation for this multi-layer approach lies in the fact that we are able to to detect ontology classes and relations in a quick and “dirty” way, which can then be consolidated, refined or rejected at further stages of analysis.

Section 5.1 describes the extraction of ontology schema components from plain text. Section 5.2 describes the extraction of ontology schema components from text annotated morphologically and with PoS and Section 5.3 describes the extraction of ontology schema components from grammatical functions.

5.1 Text-Based Layer

In this section, we describe the extraction of ontology schema components on the basis of linguistic knowledge, without using any linguistic tool. By this we mean that we have the linguistic knowledge to identify a German compound or noun, without using a linguistic tool for this task. Our aim is to state what kind of ontological knowledge can be extracted from financial financial newspapers without using any linguistic tools.

5.1.1 Concept Extraction

We start our investigation by first looking for a set of 10 relevant nouns. We decided on the following 10 nouns: *Konzern* (*corporation*), *Tochter* (*daughter*), *Unternehmen* (*company*), *Umsatz* (*business volume*), *Industrie* (*industry*), *Bank* (*bank*), *textil* (*textile*), *Branche* (*branch*), *Firma* (*firm*), *Versicherung* (*insurance*). We decided on these nouns because we considered them relevant for the economic domain. We look at whether the noun occurs alone, or in the context of a compound word, and in the latter case, whether it appears as a prefix or a suffix of the compound word. For example, the German noun *Konzern* (*corporation*) can appear in the following compounds:

(6) Der größte deutsche `Chemiekonzern`

the largest German Chemical corporation

(7) PKI erstellte erstmals einen `Konzernabschluss`

PKI generated for the first time a corporation report

(8) Der 75jährige `Konzernchef`

The 75 year old head of the corporation

(9) beim amerikanischen `Johnson-Konzern`

with the American Johnson corporation

From these examples, based on our observations, we can already extract a lot of information that can be used as the basis for an ontology:

- the compounded sequence `named_entity hyphen noun` leads to the definition of an `instanceOf` of an ontology class that could have *Konzern* (*corporation*) as its label (or an alias);

- the multi-word expression *Konzern* followed by a `noun` leads to a relation associated with the ontology class that could have *Konzern* (or an alias) as its label : `Konzern genericRelation Chef`;

- the multi-word expression `noun` followed by *Konzern* leads to a `subClassOf` relation between the expression itself and the class having *Konzern* (or an alias) as its label: *Chemiekonzern* (*chemical corporation*) is a `subClassOf` of the class *Konzern* (*corporation*);

As attractive as this very simple approach might appear, the first and most obvious drawback of this approach is that it allows us to extract possible ontology classes and relations only for nouns defined by the user. In this way we achieve high precision but a very low recall. On the other hand, the extraction is applicable only on words alone, not taking into account any possible textual context. In the following paragraphs we present a generalized approach for ontology extraction from plain text. Based on extraction rules, we will show how the extraction of ontological knowledge from plain text by just using linguistic knowledge is performed.

In order to develop a more generalized method (non-user defined) for the extraction of ontological knowledge we decided to start by extracting all noun compounds from the corpus. The decision for extracting all noun compounds is based on the assumption that from noun compounds we can extract ontological knowledge. This assumption is also supported by grammaticians who investigate the specificities of the German language (Fleischer and Barz, 1995; Lohde, 2006; Motsch, 2006). In their view, in most of the cases a noun compound[2] is built from two or more words which can also stand alone in the text and which are semantically connected[3] to each other (Duden, 2006). Based on this, the elements

[2] We deal here with the specific case of determinative noun-noun compounds. More on this aspect in Section 5.1.2

[3] Semantically connected means that, the components of the compound are connected to each other by a semantic relations.

of a compound are, for our task, potential ontology classes and the relations between the elements of the compound are potential ontological properties. On the other hand, a determinative compound is a hyponym of the second element of the compound (Erben, 1993; Donalies, 2007). As described in Chapter 4, we assume that all extracted compounds are determinative compounds.

To attain our aim, we implemented a pattern-based algorithm which exploits specific characteristics of the German language. Since noun compounding in German implies the existence of a noun and nouns start in German with a capital letter, we first decided to select from the corpus words starting with a capital letter. We just assumed that all words starting with a capital letter are nouns.

Key	Frequency	Key	Frequency
Mark	797	Ende	140
Prozent	653	Deutschen	128
Unternehmen	340	Zeit	119
Jahr	305	Branche	99
Millionen	295	Bank	96
Milliarden	264	Markt	95
Jahren	223	Dollar	94
Deutschland	171	Umsatz	88
Jahre	141	USA	86

Table 5.1: The top 20 nouns and their frequencies.

The pattern-based extraction of all possible candidates for compounding (but also ontology class candidates) has shown that, from a total number of 200107 words in the corpus, 19292 words are possible candidates for appearing as part of a compound, and therefore being an ontology class candidate. Table 5.1 lists the top 20 nouns and their frequencies in the corpus.

Here we have to notice that words like articles, prepositions, pronouns and particles such as *der*, *für*, *es*, *doch* have been already filtered out and do not appear in the list. Another aspect which has to be pointed out, is the fact that at this processing stage the counted candidate nouns are nothing else but the number of tokens potentially used later in the process of ontology extraction. It would be to ideal to count just types, because from different forms one can extract a single relation type. Not counting for morphological variations does not introduce real errors, but redundancy takes place. This redundancy we intend to reduce in a further step when we will use morphological information for defining the classes or labels of classes.

5.1.2 Compound Analysis

After extracting all noun compounding candidates the next step is to detect the compounds in the text. Therefore, we transform all extracted nouns into lower case and apply a matching algorithm on the corpus: we searched in the corpus for all words which start or end with one of the possible nouns from Section 5.1.1 and marked with a P (for prefix) or an S (for suffix) the position of the noun in the compound. In this way we filtered out a set of 3875 distinct nouns and 22142 compounds. The algorithm is designed in such a way that it covers only binary noun-noun compounds.

Table 5.2 shows an excerpt from the list of compounds for the noun *Konzern* (*corporation*). *Konzern* appears as part of a compound 75 times in the corpus, 24 times in suffix position and 51 times in a prefix position. The table can be read as follows: when *Konzern* is occurring in a prefix position it is marked by a P, and when it occurs in a suffix position is marked by an S.

From the German compounds depicted in Table 5.2 we can deduce that there is potential for extracting ontology knowledge on the basis of linguistic knowledge without using linguistic tools. From the analysis of the extracted compounds we formulated the extraction rules depicted in Figure 5.1:

The first rule introduces a generic `objectProperty` relation between the two elements of the compound (Fleischer and Barz, 1995; Lohde, 2006; Motsch, 2006).

Position encoding	Compound
S	Konzernumsatz
S	Konzernstratege
S	Konzernstruktur
S	Konzernvorstand
S	Konzernboss
P	Tabakkonzernen
P	Weltkonzerne
P	Papierkonzerne
P	Elektrokonzerns
P	Ruestungskonzern

Table 5.2: The compounds for *Konzern*.

```
compound[noun1[suggestedClass]
+ noun2[suggestedClass]]
==> objectProperty(noun1, noun2)
compound[noun1[suggestedClass]
+ noun2[suggestedClass]]
==> subClassOf(compound, noun2)
```

Figure 5.1: Ontology extraction rules from compounds.

The second one introduces a `subclassOf` relation between the compound and its second element (Erben, 1993; Donalies, 2007).

Based on the `subClassOf` rule defined in Figure 5.1 and the compounds listed in Table 5.2 we can conclude that from the compound construction `prefix + Konzern` as for *Medienkonzern* (*media corporation*), *Mutterkonzen* (*mother corporation*), *Weltkonzern* (*worldwide corporation*) we can extract `Medienkonzern subClassOf Konzern`, `Mutterkonzen subClassOf Konzern` and `Weltkonzern subClassOf Konzern`. Konzernumsatz, Konzernvorstand, Konzernboss From the construction Konzern + suffix such as *Konzernumsatz* (*corporation business volume*), *Konzernvorstand* (*corporation executive board*) and *Konzernboss* (*corporation boss*) we extract `Konzernumsatz subClassOf Umsatz`, `Konzernvorstand subClassOf Vorstand` and `Konzernboss subClassOf Boss`.

On the other hand, as depicted in the `objectProperty` rule in Figure 5.1 the components of a compound are related to each other. Based on this rule from *Konzernumsatz* (*corporation business volume*), *Konzernvorstand* (*corporation executive board*), *Konzernboss* (*corporation boss*) we can extract `Konzern objectProperty Umsatz`, `Konzern objectProperty Vorstand`, `Konzern objectProperty Boss`.

Although the examples shown here demonstrate that even at the string level to some extent ontology extraction is possible, the major drawbacks remain: the lack of domain and range, the constraint relative to the number of extracted relations and classes, multiple appearance of morphological variations, no textual context. The shortcomings such as the lack of domain and range are comprehensible since from single words no conclusion about domain or range can be drawn. The same principle applies also relative to the constraint on the number of extracted relations: the use of only linguistic knowledge combined with the extraction restriction on compounds offers no big variation and possibilities so that no more than two types of relations can be defined. All these shortcomings as well as the morphological variations such as *Konzernchef* (*chief of corporation*) and *Konzernchefs* (*chiefs of corporation*) we expect to solve by using morphological information. The larger textual context (word sequences of at least three words) is taken into consideration in the next section, where we are looking for the reformulations of the compounds extracted in this section.

5.1.3 Identification of Patterns for the Reformulation of Compounds

In order to specify and expand the set of rules extracted from compounds, we decided to search in the corpus for paraphrases of the extracted compounds. This decision is motivated by the assumption that two elements of a compound are semantically related to each other. This fact becomes more evident when analyzing the paraphrase (Lohde, 2006; Motsch, 2006). For this purpose, all extracted compounds from Section 5.1.2 are split into their components as `noun1 + noun2`, corresponding to the two noun elements. After splitting the compound back into its components, we applied a pattern-based matching algorithm for finding the paraphrases for the already extracted compounds. The pattern we are looking for is either *noun1* followed by at most three words and *noun2* or *noun2*, followed by at most three words and *noun1*. We decided for a span with a maximum of three words between the two elements of the compound because the manual analysis of the paraphrases in the corpus had shown that this distance is appropriate for covering the semantic relations between the two compound elements.

For each of the 22142 compounds found we looked in the corpus for one of the patterns described above (see Figure 5.2). The result of this pattern search returned 845 nouns which appear in 479 compounds and which have 1211 reformulations. From the 1211 reformulations we expect either to specify the relations extracted in Section 5.1.2 or to detect other phenomena which haven't been covered until now.

```
firstCompoundElement + word{1,3} + secondCompoundElement
secondCompoundElement + word{1,3} + firstCompoundElement
```

Figure 5.2: Patterns for finding paraphrases of compounds.

Taking into consideration the frequency information in Table 5.3, we can conclude that a selection of potential noun was indeed achieved. If in the beginning we had 19292 potential ontology classes to be used, in the compound selection process we had only 3875 relevant nouns, which make 20% from the initial number of nouns. The last processing step reduces the set of nouns to 4% from the initial number of nouns and to 21% from the number of nouns being part of a compound.

Concerning the relations between the two peripheric nouns of the paraphrase,

Processing stage	Concept	Compounds	Reformulations
Concept selection	19292	-	-
Compound selection	3875	22142	-
Compound filtering	845	479	1211

Table 5.3: Frequency evolution for nouns and compounds.

from the analysis of the extracted paraphrases we observe several phenomena which need to be described here. The first observation concerns the matching algorithm. Because the pattern for the identification of paraphrases is rather general than restrictive, in the corpus it also finds composition of strings such as *Länder und des Bundes* (*Länder and of federation*), *Banken die Kredite* (*banks with credits*), *Branchen, deren Kredite* (*banks whose credits*), *Unternehmen Taiwans Industrie* (*companies Taiwan's industry*). These kinds of reformulations do not add any useful information to our ontology learning approach and are therefore not taken into consideration. Moreover, this type of erroneous paraphrases will not be covered by the extraction rules developed in further processing steps.

Compound	Reformulation
Partnerairline	Airlines siehe Tabelle sind Partner
Bundesländer	Länder und des Bundes
Bankkredit	Banken eher bereit Kredite
Branchenrankings	Branchen, deren Rankings
Industrieunternehmen	Unternehmen Taiwans Industrie
Fondsverwalter	Verwalter immer dann einen Fonds

Table 5.4: Erroneous reformulations for compounds.

Besides this matching error, the reformulations as stated before also validate and enrich the ontology learning process. The validation and extension of the already extracted ontological knowledge is realized by developing new extraction rules from the extracted paraphrases. The analysis of the extracted paraphrases has shown that the valid reformulations can be grouped into two categories: the genitive paraphrase and the prepositional paraphrase. The genitive paraphrase is in fact a Genitive Phrase (GEN_Phrase), whereas the prepositional paraphrases is in fact a phrase constructed with a Preposition (Prep). Figure 5.3 depicts the two generic rules for the paraphrases. Tn this context *noun1* and *noun2* correspond to the two initial compound elements for which the Paraphrase was extracted.

```
noun2 art[genitive] modifier? modifier? noun1
noun1 prep art? modifier? noun2
```

Figure 5.3: Generic patterns for the paraphrases for compounds.

Table 5.5 lists some of the compounds and their reformulations corresponding to the two types of paraphrases.

Compound	Reformulation of compound
Aktienoptionen	Optionen auf Aktien
Bundespräsident	Präsident des Bundes
Gebührenfinanzierung	Finanzierung über Gebühren
Vorstandsmitglieder	Mitglieder des Vorstands
Aktiengesellschaften	Aktien der multinationalen Gesellschaften
Aktienbank	Aktien der deutschen Bank
Bankmitarbeiter	Mitarbeiter einer deutschen Bank

Table 5.5: Compounds and the corresponding genitive and prepositional paraphrases.

From the analysis of the extracted paraphrases we observed that, in fact, both the genitive and prepositional paraphrases encode two types of relations: one between the two nouns and the other between the second noun and its modifier. For example, the prepositional paraphrase *Mitarbeiter einer deutschen Bank* (*employees of a German bank*) validates the fact that there is a relation between *Mitarbeiter* (*employees*) and *Bank* (*bank*), namely `objectProperty(Mitarbeiter, Bank)`, but it also introduces a modification of *Bank* (*bank*) by the Adjective (Adj) *deutschen* (*German*). The same principle applies also to the genitive compound paraphrases such as *Aktien der deutschen Bank* (*shares of the German bank*). As for the the prepositional paraphrase, we extract a relation between *Aktien* and

Bank on the one hand, and *deutschen* and *Bank* on the other hand. In this way we are able to extract not only relations between the noun components of the paraphrases, but we are also able to deal with premodification phenomena. Figure 5.4 depicts once more the possible relation to be detected from a prepositional paraphrase.

```
noun1 + prep/art + modifier + noun2
==> objectProperty1(modifier, noun2)
==> objectProperty2(noun1, noun2)
```

Figure 5.4: Generic rule for ontology extraction from constructions containing nominal modifiers.

Concerning the relation type, the generic notation `objectProperty` denotes the fact, that at this stage we cannot commit to a specific relation, since the relation itself depends on the semantic classification of the modifier[4].

The generic representation of the extraction rules in Figure 5.4 show that there is indeed potential for ontology extraction from paraphrases, but the generic `objectProperty` relation needs to be further specified. In order to constrain the generic `objectProperty` we argue here for the use of linguistic annotation and lexical semantics.

5.1.4 Summary from the Text-Based Layer

From the text-based processing we can conclude that by using the specificities of the German language, more specifically of the German determinative noun-noun compound, we detected two types of relations : the `subClassOf` relation and the more generic `objectProperty` relation. The `subClassOf` relation introduces a relation between the compound and its second component, whereas the

[4] The notation `objectProperty1` and `objectProperty2` was chosen to show that between the different components of the reformulation two distinct relations can be extracted.

`objectProperty` relates semantically the two components of the compound. The generic `objectProperty` can be extended to more specific relations by analyzing the paraphrases for the extracted compounds. The analysis of the extracted paraphrases has shown that in order to extract more concrete relations, we need linguistic analysis, more precisely PoS and semantic resources.

5.2 Shallow Linguistic Analysis Layer

In this section we will show how based on PoS tagging, morphology and semantic annotation, the generic relations defined in the previous section can be further specified. Here we make a distinction between the two types of semantic connectivities described in the previous section: the semantic relation between the two nouns in the paraphrase and the semantic relation between the second noun in the paraphrase and its modifier.

5.2.1 Phrase Analysis

In this section we will describe in detail the analysis of the paraphrases for compounds based on PoS annotation, morphological annotation and lexical semantics. The aim is to show here how the already defined generic relations can be extended. As a semantic resource we use GermaNet's top semantic fields[5]. Our decision to use GermaNet is motivated by the fact that each noun in GermaNet belongs to a semantic field. For example, if for *Gesellschaft* (*corporation*) the most general hypernym is *entity*, the one considered for our investigation is the one appearing two levels above the most general hypernym, here *group*.

In order to define more specific ontology extraction rules for the extracted para-

[5] Each word in GermaNet is assigned to a semantic field. Table A.1 in the Appendix list all semantic fields for nouns.

phrases, we annotate the corresponding set of phrases with SProUT (Drozdzynski et al., 2004), more precisely with the PoS and morphological analyzer incorporated into Shallow Processing with Unification and Typed Feature Structures (SPROUT)[6]. For the lexical semantic annotation we use GermaNet (Kunze and Lemnitzer, 2002). In addition to PoS and lexical semantics, we also used the morphological information about the lemma of the nouns occurring in the paraphrases. The lemmas are important for avoiding redundancy in ontology classes. By using lemmas as classes, a noun appears just once in the ontology, without all its morphological variations.

The paraphrases for the compounds can be split into 2 categories: the genitive phrases and the prepositional phrases.

Genitive Phrases

We first describe the set of developed rules for the extraction of ontology schema components from genitive phrases. Before applying the rules we developed the pattern in Figure 5.5 which detects all genitive phrases[7].

```
noun1[PoS=noun]GN=semanticClass] +
art/pron[case=genitve] + modifier{0,1} +
noun2[PoS=noun]GN=semanticClass]
```

Figure 5.5: Pattern for the extraction of ontology schema components from genitive phrases.

The pattern described in Figure 5.5 can be explained as follows: if between the two nouns encountered in the paraphrase we find an article denoting the genitive and an optional modifier, then we have identified a genitive phrase from which

[6] SProUT (Shallow Processing with Unification and Typed Feature Structures) is a platform for development of multilingual shallow text processing and information extraction systems which incorporates in it a morphological analyzer and a PoS tagger.

[7] We use the notation GermaNet Semantic Class (GN) for marking the semantic class of the noun and Semantic Class (SC) for marking the semantic class for modifiers.

we can extract ontological knowledge. Depending on the semantic annotation of the nouns in the genitive phrase we developed a set of six extraction rules. For each of these rules the following principles apply: the lemmas of the nouns in the phrase become classes in the ontology, whereas the relation between these nouns is specified by properties in the ontology. The type of the ontological property depends on the semantic classification of the nouns and the developed rules. The domain and the range of the specified property corresponds to the semantic class of the two nouns in the phrase. In contrast, each noun becomes a subclass of the semantic class it belongs to. In this way we enable the structural integration of new classes and relations into the ontology and can connect them to more general nouns such as person, group, possession, relation, attribute, event, object, state, and location. Another issue here is the exploitation of GermaNet, which for each noun provides us, ideally, with synonyms, antonyms, hyponyms and meronyms. In fact, if any of this semantic information is available, we introduce it into the ontology either as a class or a class label.

As mentioned above, depending on the semantic classification of both nouns we assigned 6 rules to this pattern. The first rule is depicted in Figure 5.6. The rule itself is an instantiation of the pattern described above and describes the case when the first noun in the paraphrase is semantically identified as a subclass of *Group* or a *Person* and the second noun as a subclass of *Group*.

```
if one of the nouns has GN=person/group and the other GN=group
==> noun[GN=person/group] hasPosition noun[GN=group]
```

Figure 5.6: `hasPosition` extraction rule from genitive phrases.

Table 5.6 lists 2 examples in which one noun has been semantically identified as a `person` or a `group`, such as *Experten* (*experts*) and *Führung* (*leadership*), and the second noun is denoting a group, such as *Bank* (*bank*) and *Konzern* (*corporation*). The developed rule for this pattern extracts the following relation `hasPosition(Bank, Experten)` and `hasPosition(Konzern, Führung)`. The relation denotes the fact that in a bank (*Bank*) or in a corporation (*Konzern*) there is a position which is occupied by a person, here an expert (*Experten*) or a group of people such as leaders (*Führung*).

Example 1	Compound	Bankexperten
	Paraphrase	Experten[GN=person] der Bank[GN=group]
	Relation	hasPosition(Bank, Experten)
Example 2	Compound	Konzernführung
	Paraphrase	Führung[GN=group] des Gerling Konzerns[GN=group]
	Relation	hasPosition(Konzern, Führung)

Table 5.6: Examples for the `hasPosition` extraction rule from genitive phrases.

When writing the extracted knowledge into the ontology, the lemmas of the nouns involved in the relation, *Expert*, *Bank*, *Führung* and *Konzern* become classes in the ontology. At the same time, they enter the ontology as subclasses of *Person* and *Group*. The binary relations `hasPosition(Bank, Experten)` and `hasPosition(Konzern, Führung)` will then have domain *Group* and range *Person*, respectively *Group*. As mentioned already, for each of these four nouns, we collect all semantic information provided by GermaNet: synonyms, antonyms, hyponyms and meronyms. For the noun *Expert* GermaNet provides the two synonyms *Fachmann* and *Fachfrau* and a larger set of hyponyms. The synonyms become labels of the class *Expert* and the hyponyms will become subclasses of the class *Expert* in the ontology. The same principle, applies for each new noun introduced into the ontology.

```
if one of the nouns has GN=group and the other GN=possession
==> noun[GN=group] disposesOver noun[GN=possession]
```

Figure 5.7: `disposesOver` extraction rule from genitive phrases.

The next rule for genitive phrases concerns the case when one noun has been classified by GermaNet as belonging to the semantic class *Possession* and the other one to the semantic class *group*. The rule itself is depicted in Figure 5.7. An example for the application of this rule is listed in Table 5.7. The compound *Aktiengesellschaft* has been paraphrased with *Aktien der Gesellschaft* and is covered by the pattern defined for genitive phrases. The two nouns of the phrase *Aktien* and *Gesellschaft*, have been semantically classified as belonging to the semantic classes *Possession* and *group*, which means that from this phrase we can extract `disposesOver(Aktie, Gesellschaft)` with domain *Possession* and range *group*. As for the previous rule, the two classes *Aktie* and *Gesellschaft* will become subclasses of *Possession* and *group*. Concerning the additional semantic information provided by GermaNet, for *Aktie* we get a set of hypoynms such as *Stammaktie* and *Bankaktie* which become subclasses of *Aktie*. GermaNet also provides the holonym[8] for *Aktie*, *Aktienkapital*. In this case we introduce the following relation: `partOf(Aktie, Aktienkapital)` with domain *Possession* and range *Possession*. For *Gesellschaft* GermaNet returns a set of hyponyms. As for *Aktie*, the hyponyms become subclasses of *Gesellschaft*.

Example 1	Compound	Aktiengesellschaft
	Paraphrase	Aktien[GN=possession] einer Gesellschaft[GN=group]
	Relation	disposesOver(Gesellschaft, Aktien)
Example 1	Compound	Bankaktie
	Paraphrase	Aktie[GN=possession] der Bank[GN=group]
	Relation	disposesOver(Bank, Aktie)

Table 5.7: Examples for `disposesOver` extraction rule from genitive phrases.

The third rule deals with the *hasDimension* defined in Figure 5.8. In order for this rule to fire, one noun needs to be semantically classified as a measure, whereas the semantic classification of the second one needs to be different from measure.

For example, from the phrase *Zahl der Beschäftigten* (*number of employes*) in Table 5.8 we extract `hasDimension(Beschäftigten, Zahl)` with domain *Person* and range *Quantity*. For both nouns *Beschäftigten* (*employees*) and *Zahl*

[8] Holonymy defines the relationship between a term denoting the whole and a term denoting a part of, or a member of, the whole.

```
if one of the nouns has GN=quantity and
the second GN=person/possession
==> noun[GN=!quantity] hasDimension noun [GN=quantity]
```

Figure 5.8: `hasDimension` extraction rule from genitive phrases.

(*number*), we collect all semantic information provided by GermaNet (hyponyms and holonyms) and introduce it into the ontology as described above.

Example 2	Compound	Arbeitslosenzahl
	Paraphrase	Zahl[GN=quantity] der Arbeitslosen[GN=person]
	Relation	hasDimension(Arbeitslose, Zahl)
Example 2	Compound	Beschäftigtenzahl
	Paraphrase	Zahl[GN=quantity] der Beschäftigten[GN=person]
	Relation	hasDimension(Beschäftigte, Zahl)

Table 5.8: Examples for the `hasDimension` extraction rule from genitive phrases.

The next rule, depicted in Figure 5.9, concerns the case when one noun in the phrase was semantically identified as an event and the other noun as something different from event. The rule itself, as for all the other rules for genitive phrases presented in this section, is an instantiation of the pattern in Figure 5.5.

```
if one of the nouns has GN=event and the second GN=!event
==> noun[GN=!event] hasEvent noun[GN=event]
```

Figure 5.9: `hasEvent` extraction rule from genitive phrases.

Table 5.9 contains two paraphrases on which this rule applies. For example, the compound *Konjunkturankurbelung* which has been paraphrased in the corpus as *Ankurbelung der Konjunktur* introduces the relation `hasEvent(Konjunktur, Ankurbelung)` with domain *Situation* and range *Event*. The same principle applies for the phrase *Förderung der Investition* from which we extract `hasEvent(Investition, Förderung)` with domain *Possession* and range *Event*. As before, for each of these four nouns we collect the available synonyms, hyponyms and meronyms and introduce into the ontology class labels, subclasses or the `partOf` relation.

Example 1	Compound	Konjunkturankurbelung
	Paraphrase	Ankurbelung[GN=event] der Konjunktur[GN!=event]
	Relation	hasEvent(Konjunktur, Ankurbelung)
Example 2	Compound	Investitionsförderung
	Paraphrase	Förderung[GN=event] des Investition[GN=!event]
	Relation	hasEvent(Investition, Förderung)

Table 5.9: Examples for the `hasEvent` extraction rule from genitive phrases.

The fifth rule depicted in Figure 5.10 covers the case when one noun has been semantically identified as attribute and the other one as something different to an attribute. In this case we introduce the `hasAttribute` relation with domain and range determined by the two nouns of the paraphrase.

```
if one of the nouns has GN=attribute and the second GN=!attribute
==> noun[GN=!attribute] hasAttribute noun[GN=attribute]
```

Figure 5.10: `hasAttribute` extraction rule from genitive phrases.

Table 5.10 lists two examples for this rule. From the phrase *Motivation der Mitarbeiter* we extract `hasAttribute(Mitarbeiter, Motivation)` with domain *Person* and range *Attribute*. From *Pflichten der Kunden* we extract `hasAttribute(Kunde, Pflicht)` with domain *Person* and range *Attribute*. For all four nouns we collect from GermaNet all available semantic information, here the synonyms, antonyms and hyponyms, and introduce them into the ontology as class labels or subclasses. For example, GermanNet provides the noun *Kunde* as a the synonym for *Kundin*, the antonym *Verkäufer* and a set of hyponyms such as *Käufer* and *Auftraggeber*. The synonym *Kundin* will become a class label of the class *Kunde*, *Verkäufer* becomes an argument of the relation `isAntonymTo(Kunde, Verkäufer)` with domain *Person* and range *Person*. The hyponyms *Käufer* and *Auftraggeber* become arguments in the relation `partOf(Kunde, Käufer)` and `partOf(Kunde, Auftraggeber)` both with domain *Person* and range *Person*.

Example 1	Compound	Mitarbeitermotivation
	Paraphrase	Motivation[GN=attribute] der Mitarbeiter[GN!=attribute]
	Relation	hasAttribute(Mitarbeiter, Motivation)
Example 2	Compound	Kundenpflichten
	Paraphrase	Pflichten[GN=attribute] der Kunden[GN=person]
	Relation	hasAttribute(Kunde, Pflicht)

Table 5.10: Examples for the `hasAttribute` extraction rule from genitive phrases.

The last rule covers the case when one noun has been semantically identified as a location and the other noun as something different to a location. The rule, depicted in Figure 5.11 introduces the `hasLocation` relation between the two nouns in the phrase.

```
if one of the nouns has GN=location and the  other has GN=!location
==> noun[GN=!location] hasLocation noun[GN=location]
```

Figure 5.11: `hasLocation` extraction rule from genitive phrases.

Table 5.11 lists two examples for this rule. As before, the domain and the range of the relations is determined by the semantic classification of the left and right argument of the relation. The relation `hasLocation(Industrie, Land)` has domain *Group* and range *Location*, since GermaNet identified *Industrie* as belonging to the semantic class *Group* and *Land* as belonging to the semantic class *Location*. The relation `hasLocation(Unternehmer, Land)` has domain *Person* and range *Location*. For each of the nouns in Table 5.11 we collect the additional semantic information provided by GermaNet, here synonyms and hyponyms, and transform them into class labels and subclasses of the corresponding nouns.

Prepositional Phrases

The second type of phrases identified in our set of paraphrased compounds is the Prepositional Phrase (PP). The pattern for the propositional phrases is depicted in Figure 5.12. The pattern is similar to the first pattern described in Figure 5.5

Example 1	Compound	Industrieländer
	Paraphrase	Industrie[GN=group] der alten Länder[GN=location]
	Relation	hasLocation(Industrie, Land)
Example 2	Compound	Landesunternehmer
	Paraphrase	Unternehmer[GN=person] des Landes[GN=location]
	Relation	hasLocation(Unternehmer, Land)

Table 5.11: Examples for the `hasLocation` extraction rule from genitive phrases.

and can be explained as follows: if between the both nouns encountered in the phrase we find a preposition followed by an optional article or/and an optional modifier, then we have identified a prepositional phrase from which we consider we can extract ontological knowledge.

```
noun1[PoS=noun][GN=semanticClass] +
preposition + article/modifier{0,1}? + modifier{0,1}? +
noun2[PoS=noun][GN=semanticClass]
```

Figure 5.12: Pattern for the extraction of ontology schema components from prepositional phrases.

In order to extract ontological knowledge from this type of phrase we developed a set of seven extraction rules which build on PoS and the GermaNet's semantic classification of the nouns in the phrases. Depending on the semantic classification of the two nouns, new classes and ontology properties are extracted. From the prepositional phrase we extract five of the six relations already enumerated above and a additional one, the `hasAffiliation` relation. Since the first five rules (`disposesOver`, `hasDimension`, `hasEvent`, `hasAttribute`, `hasLocation`) we already described in detail in the previous section, we concentrate here on just listing examples for these already defined relations. We explain in more detail the new defined relation `hasAffiliation`.

One of the rules already defined for genitive phrase and applicable also for prepositional phrases is the `disposesOver` rule. Figure 5.12 lists two examples from which we extract the `disposesOver` relation.

Example 1	Compound	Aktienbörse
	Paraphrase	Aktien[GN=possession] an der Börse[GN=group]
	Relation	disposesOver(Börse, Aktien)
Example 1	Compound	Aktienmärkte
	Paraphrase	Aktien[GN=possession] in freien Märkten[GN=group]
	Relation	disposesOver(Märkten, Aktien)

Table 5.12: Examples for the `disposesOver` extraction rule from prepositional phrases.

Table 5.13 lists two examples for the already defined rule for the `hasDimension` relation. The rule is the same as defined for the genitive phrase (see Figure 5.8), only its application is extended to prepositional phrases.

Example 1	Compound	Milliardenhöhe
	Paraphrase	Höhe[GN=attribute] von 12 Milliarden[GN=quantity]
	Relation	hasDimension(Höhe, Milliarden)
Example 2	Compound	Millionengewinn
	Paraphrase	Gewinn[GN=possession] von Millionen[GN=quantity]
	Relation	hasDimension(Gewinn, Millionen)

Table 5.13: Examples for the `hasDimension` extraction rule from prepositional phrases.

Table 5.14 and Table 5.15 list two examples for the extraction of the corresponding `hasEvent` and `hasAttribute` rules.

Example 1	Compound	Produktentwicklung
	Paraphrase	Entwicklung[GN=event] von Produkten[GN=object]
	Relation	hasEvent(Produkten, Entwicklung)
Example 2	Compound	Problemlösung
	Paraphrase	Lösung[GN=event] für das Problem[GN=cognition]
	Relation	hasEvent(Problem, Lösung)

Table 5.14: Examples for the `hasEvent` extraction rule from prepositional phrases.

As for the other extraction rules, the examples listed in Table 5.16 comply with the corresponding extraction rule depicted in Figure 5.11.

	Compound	Mitarbeitermotivation
Example 1	Paraphrase	Motivation[GN=attribute] für Mitarbeiter[GN=person]
	Relation	hasAttribute(Mitarbeiter, Motivation)
	Compound	Zementpreise
Example 2	Paraphrase	Preise[GN=attribute] für Zement[GN=substance]
	Relation	hasAttribute(Zement, Preise)

Table 5.15: Examples for the `hasAttribute` extraction rule from prepositional phrases.

The really new relation extracted from the prepositional phrases is the `hasAffiliation` relation.

	Compound	Westlöhne
Example 1	Paraphrase	Löhne[GN=possession] im Westen[GN=location]
	Relation	hasLocation(Löhne, Westen)
	Compound	Stadtwohnung
Example 2	Paraphrase	Wohnung[GN=object] in der Stadt[GN=location]
	Relation	hasLocation(Wohnung, Stadt)

Table 5.16: Examples for the `hasLocation` extraction rule from prepositional phrases.

Figure 5.13 depicts the newly introduced `hasAffiliation` relation. As for the rules defined before, the result of the rule depends on GermNet's semantic classification of the two nouns appearing in the phrase. The rule describes the case when the first noun in the phrase has been classified by GermaNet as a *group* or a *person* and the second one has been classified as *Group*.

```
if one of the nouns has GN=person/group and the other GN=group
==> noun[GN=person/group] hasAffiliation noun[GN=group]
```

Figure 5.13: `hasAffiliation` extraction rule from prepositional phrases.

The application of the rule in Figure 5.13 on prepositional phrases can be described as follows. If the first noun has been identified as *group* or a *person* such as *Ministerium* and *Angestellten* and the second noun has been identified as a *group* such as *Wirtschaft* and *Banken*, then we extract the following `hasAffiliation` relations: `hasAffiliation(Wirtschaft, Ministerium)` with domain *Group* and range *Group* and `hasAffiliation(Angestellten, Banken)` with domain *Person* and range *Group*. Table 5.17 lists the corresponding examples for the `hasAffiliation` rule. The nouns in the phrase are introduced into the ontology as subclasses of the classes *Group* and *Person*. As for the other relations introduced until now, we will use all additional information delivered by GermaNet for the corresponding nouns. For example, for the noun *Angestellte* (*employee*) GermaNet provides the synonym *angestellter_Mensch* and a list of hyponyms such as *Bankangestellter* (*bank employee*), *Gerichtsangestellter* (*justice emplyee*) and *Bibliothekar* (*librarian*). The synonyms will enter the ontology as class labels and the hyponyms as subclasses of the corresponding noun.

Example 1	Compound	Wirtschaftsministerium
	Paraphrase	Ministerium[GN=group] für Wirtschaft[GN=group]
	Relation	hasAffiliation(Wirtschaft, Ministerium)
Example 2	Compound	Bankangestellten
	Paraphrase	Angestellten[GN=person] in Banken[GN=group]
	Relation	hasAffiliation(Angestellten, Banken)

Table 5.17: Examples for the `hasAffiliation` extraction rule from prepositional phrases.

As a final remark, we need to say here that the rules presented above must be applied in a given order in order to provide correct relations. The constraint on the application order of the presented rules implies that the ontology extraction rules for the `hasAffiliation` and the `hasDimension` relations should be applied before applying the ontology extraction rules for `hasAttribute` and `hasEvent`.

5.2.2 Analyzing Premodification Phenomena

During the analysis process of the extracted paraphrases, we observed that the nouns in the phrases are modified by adjectives and adverbs. A closer analysis of this modification phenomena has shown that modification constructions exhibit potential for the extraction of ontology knowledge. From modification phenomena we are able to extract a different type of relation to the ones described in the previous section. In order to cover all modification phenomena, we extended our analysis to all genitive and prepositional phrases in the corpus and semantically classified the top 100 most frequent adjectives and adverbs.

PoS	Overall Frequency	Top 100 Cumulated Frequency
Adjectives	4991	5713
Adverbs	364	7742

Table 5.18: Frequencies of adverbs and adjectives.

For adjectives we use the semantic classification proposed by Lee (1994) and for adverbs we used the classification proposed by Lobeck (2000). A detailed classification of the adjectives and adverbs can be found in Section A.2.3 and A.2.4 in the Appendix. For adjectives we decided to use this classification instead of GermaNet, because GermaNet does not allow the extension of the lexicon. By using these classifications for adjectives, we are able to extend the semantic lexicon with new entries. Since for adverbs GermaNet does not provide any classification, we decided, based on our previous experience, to use here the one proposed by Lobeck (2000).

Table 5.18 lists the total number of adjectives and adverbs, as they have been identified as such by the PoS tagger. We have 4991 unique adjectives and 364 unique adverbs in the corpus. In parallel the table also lists how often the top 100 adverbs and adjectives appear in the corpus. The count has shown that the top 100 adjectives appear 5713 times in the corpus, whereas the top 100 adverbs 7742 times.

The analysis of the paraphrases in which modification phenomena appear has shown that we can distinguish between two construction types. The first one, depicted in Figure 5.14 is concerned with the case when a noun is modified by one or more modifiers which are not connected to each other by any Conjunction (Conj) or comma. An example for this type of modification constructions is *großen deutschen Konzern* (*big German concern*).

```
modifier1[PoS=adv/adj][SC=semanticClass]
+ modifier2[PoS=adj][SC=semanticClass]{0,1}
+ noun[PoS=noun][GN=semanticClass]
```

Figure 5.14: Pattern1: Pattern for the extraction of modified nouns.

The second pattern, depicted in Figure 5.15, covers the cases when the modifiers are connected to each other by a comma or a conjunction, such as *kleinen, krisengeplagten Firmen* (*small, crisis affected firms*).

```
(modifier[PoS=adj][SC=semanticClass]
+ separator[PoS=comma]){0,n}
+ modifier[PoS=adj][SC=semanticClass]
+ separator[PoS=conj/comma]
+ modifier[PoS=adj][SC=semanticClass]
+ noun[PoS=noun][GN=semanticClass]
```

Figure 5.15: Pattern2: Extended pattern for the extraction of modified nouns.

For the first pattern we developed three extraction rules, depending on the number of modifiers and the type of the modifier, more precisely Part-of-Speech (PoS) of the modifier. The first rule concerns the simple case when a noun is modified by just one modifier. In this case, depending on the semantic classification of the modifier, a new relation is introduced.

```
modifier1[PoS=adj][SC=semanticClass]
+ noun[PoS=noun][GN=semanticClass]
==> relationDerived(noun, modifier1)
```

Figure 5.16: First extraction rule from phrases matched by pattern 1.

For the sentence depicted in example 10 the construction *deutschen Tochterfirmen* introduces the relation `hasAffiliation(Tochterfirmen, deutsch)` according to the classification of *deutsch* as an adjective of affiliation. So each adjective classified as an adjective of affiliation introduces the `hasAffiliation` relation. The extracted relation will be written into the ontology by using reified relations[9] as follows: the noun *Tochterfirmen* will become a subclass of the generic class *Group*; as for all nouns until now, we gather all semantic information provided by GermaNet (synonyms, antonyms, hyponyms and meronyms) for *Tochterfirmen* and introduce it into the ontology. The relation `hasAffiliation` holds between the class *Tochterfirmen* and the class *AffiliationRelation* having *Group* as domain and *affiliationRelation* as range. Additionally, the class *AffiliationRelation* has as value the stem of the adjective *deutsch*: `hasAffiliationValue(AffiliationRelation, Deutsch)`. By using reified relations we also allow not only the representation of the relations between adverbs and adjectives such in *wahrscheinlich große Gewinne*, but also the representation of intersective adjectives. Reified relations also allow us to represent n-ary relations.

(10) Entsprechend verfahren wird bei deutschen Tochterfirmen.

We proceed corresponding with German subsidiary companies.

The next rule (see Figure 5.17) is applied when a noun is modified by two adjectives which are not connected to each other by any punctuation sign or conjunction. In this case we speak of an aggregation of adjectives. The rule in Figure 5.17 for aggregative modifiers can be explained as follows: each modifier in a Nominal Phrase (NP), depending on its semantic class, introduces a specific relation between itself and the head. Furthermore, each modifier that is not a direct neighbor of the head noun modifies the subsequent modification sequence of modifiers and the head noun. The types of the introduced relations depend on the semantic classification of the modifiers.

[9] Reified relations in OWL are defined as relations which are transformed into classes.

```
modifier1[PoS=adj][SC=semanticClass]
+ modifier2[PoS=adj][SC=semanticClass]
+ noun[PoS=noun][GN=semanticClass]
==> relationIntroducedByModfier2(noun, modifier2)
==> relationIntroducedByModfier1(modifier1, modifier2 noun)
```

Figure 5.17: Second extraction rule from phrases matched by pattern 1.

Applying the rule on the NP *selbständig bilanzierende Tochterfirmen* in sentence 11 we extract the following relations: `hasReference(Tochterfirmen, Bilanzierend)`, but also `hasMode(Bilanzierende Tochterfirmen, Selbstständig)`. The adjectives *bilanzierend* and *selbstständig* are semantically classified as belonging to the semantic class of reference adjectives, respectively modal adjectives. The representation of those relations in the ontology is performed by using reified relations: `hasReference(Tochterfirmen, ReferenceRelation)` with domain *Group* and range *ReferenceRelation*, `hasReferenceValue(ReferenceRelation, Bilanzierend)` and `hasMode(Bilanzierende Tochterfirmen, ModeRelation)` with domain *Group* and range *ModeRelation*, `hasModeValue(ModeRelation, Selbstständig)`.

(11) Selbständig bilanzierende Tochterfirmen werden in den Branchenrankings nur aufgeführt, wenn das Grössenkriterium erfüllt ist und die Muttergesellschaft einer anderen Branche zugeordnet ist.

Autonomously balanced subsidiary companies are listed in the branch ranking if the dimension criteria is fulfilled and the holding company belongs to another branch.

The next rule describes the case when an adverb and an adjective modify a noun. Such constructions are handled as described by Duden (2006): the adjective modifies the noun and the adverb modifies the adjective. The introduced relations depend on the semantic class of the adjective and adverb.

```
modifier1[PoS=adv][SC=semanticClass]
+ modifier2[PoS=adj][SC=semanticClass]
+ noun[PoS=noun][GN=semanticClass]
==> relationIntroducedByModfier2(noun, modifier2)
==> relationIntroducedByModfier1(modifier2, modifier1)
```

Figure 5.18: Third extraction rule from phrases matched by pattern 1.

From the phrase *sehr viel Geld* in sentence 12 according to the classification of the dimensional adjective *viel* and of the adverb *sehr* we extract the following relations: `hasDimension(Geld, Viel)` and `hasAspect(Viel, Sehr)`. The noun *Geld* will enter the ontology as subclass of the generic class *Possession*, whereas both modifier *sehr* and *viel* enter the ontology as subclasses of the generic class *Aspect* and *Dimension*. The relation itself is represented as described above by using reification.

(12) Tradition und Imagepflege bringen nichts und kosten sehr viel Geld.

Tradition and image maintenance do not pay and cost a lot of money.

Through the analysis of the extracted genitive and prepositional phrases we detected two types of modification phenomena: constructions where multiple modifiers are not connected to each other by any punctuation sign or conjunction and the constructions where the modifiers are connected to each other by comma or/and conjunctions. The first three rules presented above, belong to the first category.

```
modifier1[PoS=adj][SC=semanticClass]
+ separator[PoS=comma]
+ modifier2[PoS=adj][SC=semanticClass]
+ noun[PoS=noun][GN=semanticClass]
==> relationIntroducedByModfier1(noun, modifier1)
==> relationIntroducedByModfier2(noun, modifier2)
```

Figure 5.19: First extraction rule from phrases matched by pattern 2.

The next three rules presented below (see Figure 5.19, Figure 5.20 and Figure 5.21) cover the modification constructions when one or more modifiers are connected to each other by a comma and/or a conjunction.

```
modifier1[PoS=adj][SC=semanticClass]
+ separator[PoS=conj]
+ modifier2[PoS=adj][SC=semanticClass]
+ noun[PoS=noun][GN=semanticClass]
==> relationIntroducedByModfier1(noun, modifier1)
==> relationIntroducedByModfier2(noun, modifier2)
```

Figure 5.20: Second extraction rule from phrases matched by pattern 2.

For each of the cases implying multiple premodifiers, each modifier introduces a new relation between itself and the head noun. Since these rules deal in fact with an enumeration of adjectives, we expect here that every adjective in the enumeration belongs to the same semantic class.

```
modifier1[PoS=adj][SC=semanticClass]
+ separator[PoS=punct]
+ modifier2[PoS=adj][SC=semanticClass]
+ separator[PoS=conj]
+ modifier3[PoS=adj][SC=semanticClass]
+ noun[PoS=noun][SC=semanticClass]
==> relationIntroducedByModfier1(noun, modifier1)
==> relationIntroducedByModfier2(noun, modifier2)
==> relationIntroducedByModfier3(noun, modifier3)
```

Figure 5.21: Pattern for the extraction of modified nouns.

The phrase *kulturelle, wirtschaftliche und gesellschaftliche Umfeld* in Example 13 fulfills our expectation in that sense, since all three adjectives have been semantically classified as reference adjectives. According to rule depicted in Figure 5.21 and based on the semantic class of the adjectives, we extract then `hasReference(Umfeld, Kulturelle)`, `hasReference(Umfeld, Wirtschaftliche)` and `hasReference(Umfeld, Gesellschaftliche)`. For writing these relations into the ontology we use, as before, reification. The modifier and the noun enter the ontology as subclasses of the semantic classes they have been semantically assigned to.

(13) Dabei werden wir das kulturelle, wirtschaftliche und gesellschaftliche Umfeld auf lokaler Ebene respektieren.

The local cultural, economical and social environment is respected.

5.2.3 Ontology Population with Named Entities

In the previous sections we have shown and explained the extraction rules for ontology schema components. In this section we deal with ontology population, more specifically with the instantiation of organizations, persons, locations and the detection of money, quantity and temporal units. We also decided to perform ontology population at this stage because the morphological analyzer contains a NE recognition component. In this way we can perform ontology population without any additional effort.

Concerning the NEs, we distinguish between two types of constructions here: the compounded construction between a NE and a noun such as *Colonia-Konzern* and *Manager Herbert Henzler* and the stand alone NEs such as *Daimler-Benz*. Based on the NE recognition, we developed a set of eleven rules which handle both NE types and instantiate the generic classes *Organization*, *Person*, *Location*, *Money*, *TemporalUnit* and *Quantity*.

The first rule covers the organizations with a descriptor, such as *Chemiekonzern Rhone-Poulac* or *Technologiekonzern Alcatel-Alsthom*. The descriptors are here *Chemiekonzern* and *Technologiekonzern*, which in fact explain what *Rhone-Poulac* and *Alcatel-Alsthom* are. The application of the rule depicted in Figure 5.22 on the NE *Chemiekonzern Rhone-Poulac* can be explained as follows: *Chemie-*

konzern has been identified as the descriptor of the organization *Rhone-Poulac* which means that *Rhone-Poulac* becomes an instance of the class *Chemiekonzern*, which already exists in the ontology as a subclass of the class *Konzern*. In addition to the instantiation, the NE recognition also provides the information that *Konzern* is an organization, respectively a subclass of the more generic class *Group*. By this, we modify the existing class structure in the ontology by making it more specific. As for all relations described in this chapter, we use here the full semantic power of GermaNet by adding, if not already introduced, the synonyms, antonyms, meronyms and hyponyms of the nouns into the ontology.

```
NE[ne-organization[descriptor, orgname]]
==> instanceOf(orgname, descriptor)
==> subClassOf(descriptor, organization)
```

Figure 5.22: First extraction rule from NE's detected as organizations.

The next rule (see Figure 5.23) covers the case when an organization such as *Coca-Cola* or *Lufthansa-Condor* appear alone in the text, without being accompanied by any descriptor. In such cases we just perform the instantiation `instanceOf(Coca-Cola, Organization)` and `instanceOf(Lufthansa-Condor, Organization)`.

```
NE[ne-organization]
==> instanceOf(ne-orgname, organization)
```

Figure 5.23: Second extraction rule from NE's detected as organizations.

The third rule for NEs covers the case when the NE holds a designator such as in *Woolworth Inc.* and *Torras-Holding* where *Inc.* and *Holding* were identified as the designators. By the designator we detect in fact the organizational form of the company. By applying the rule depicted in Figure 5.24 we are able to instantiate the generic class *Organization* with *Woolworth* and *Torras*. In addition to the instantiation, we also introduce into the ontology the relation `isOrganizedAs(Woolworth, Inc)` and `isOrganizedAs(Torras, Holding)` which gives us information about the the various organization types.

```
NE[ne-organization[ne-designator, orgname]]
==> instanceOf(ne-orgname, organization)
==> isOrganizedAs(organization, ne-designator)
```

Figure 5.24: Third extraction rule from NE's detected as organizations.

The next rules handle the persons identified as such in the corpus. Here we distinguish between persons which have been recognized as occupying a specific position and persons without a position. Figure 5.25 covers the case when, in text, we have *Geschäftsführer Karl-Ulrich Kuhlo* and *Professor Gerhard Schmitt-Rink*. In this case *Geschäftsführer* and *Professor* were identified by the NE recognizer as positions occupied by the persons *Karl-Ulrich Kuhlo* and *Gerhard Schmitt-Rink*. Based on the results delivered by the NE recognizer we introduce into the ontology the following relations `subClassOf(Geschäftsführer, Position)`, `subClassOf(Professor, Position)` and the generic relation `hasPosition(Person, Position)` with domain *Person* and range *Position*. All arguments of the relations above enter the ontology as classes. Additionally, we instantiate the generic class *Person* as follows: `instanceOf(Karl-Ulrich Kuhlo, Person)` and `instanceOf(Gerhard Schmitt-Rink, Person)`.

```
NE[ne-person[position]]
==> instanceOf(ne-person, person)
==> subClassOf(ne-position, position)
==> occupiesPosition(ne-person, position)
```

Figure 5.25: First extraction rule from NE's detected as persons.

The next rule covers the person NEs which were recognized as such without any information about the position they occupy. In such cases we just instantiate the generic class *Person*. For example *Hans-Jürgen Krupp* and *McGraw-Hill* are becoming instantiations of the class *Person*.

```
NE[ne-person]
==> instanceOf(ne-person, person)
```

Figure 5.26: Second extraction rule from NE's detected as persons.

Figure 5.27 shows the instantiation rule for locations. By applying this rule we instantiate the generic class *Location* with *Sachsen-Anhalt*.

```
NE[ne-location]
==> instanceOf(ne-location, location)
```

Figure 5.27: Rules for ontology extraction from NE's detected as location.

The next rule covers all money NEs such as *1,2 Milliarden US-Dollar* and *2 Millionen Mark*. The rule in Figure 5.28 handles this type of construction and introduces the datatype property `hasMoneyValue(Million, String)` with domain number and range string, `instanceOf(String, 2)` and `instanceOf(Currency, Mark)`.

```
NE[ne-money[currency]]
==> hasMoneyValue(ne-money, string)
==> instanceOf(string, stringValue)
==> instanceOf(ne-currency, currency)
```

Figure 5.28: Rules for ontology extraction from NE's detected as money.

The next three rules presented below perform the instantiation of the generic classes *Date* and *Quantity*. For example *1991* was recognized as a date NE having as temporal unit *Jahr* (*year*). By applying the rule in Figure 5.29 we introduce the datatype property `hasTimeUnitValue(Jahr, String)` with domain *TimeUnit* and range *String*. The class *String* is instantiated then with *1991*.

The same principles apply also to the instantiation of the generic classes *Quantity*. The corresponding instantiation rules are depicted in Figure 5.30. As a result of these rules from *800-Megawatt* and *Zehn Gramm*, we introduce the datatype prop-

```
  NE[ne-duration[date]]
  ==> hasTimeUnitValue(ne-duration, String)
  ==> instanceOf(String, StringValue)
```

Figure 5.29: Rules for ontology extraction from NE's detected as span.

erty `hasQuantityValue(Gramm, String)` and `hasQuantityValue(Megawatt, String)` with the instantiations `instanceOf(String, zehn)` and `instanceOf(String, 800)`.

```
  NE[quantity]
  ==> hasQuantityValue(ne-quantity, String)
  ==> instanceOf(String, StringValue)
```

Figure 5.30: Rules for ontology extraction from NE's detected as quantity.

A special case is the construction of the type *WDR-Chef Friedrich Nowottny* where *WDR* and *Chef Friedrich Nowottny* have been identified as two different NEs. In such a case, from the NE *Chef Friedrich Nowottny*, we instantiate the generic class *Person* with *Friedrich Nowottny* and introduce the relation `occupiesPosition(Friedrich Nowottny, Chef)` with domain *Person* and range *Position.* The NE *WDR* instantiates the generic class organization and from the construction *WDR-Chef* we extract `hasPosition(WDR, Chef)` with domain *Group* and range *Position.*

```
NE[ne-organization]-NE[ne-person[position]]
==> instanceOf(ne-person, person)
==> instanceOf(ne-organization, organization)
==> occupiesPosition(ne-person, position)
==> hasPosition(ne-organization, position)
```

Figure 5.31: Rules for ontology extraction from NE's detected as organizations.

5.2.4 Summary from the Semantic- and PoS-Based Processing

In this section we have shown how based on PoS and lexical semantics we were able to extract a set of relevant classes and relations. For this purpose, we analyzed and extracted ontological knowledge from two types of linguistic phenomena: genitive and prepositional phrases and premodification of nouns. Table 5.19 lists some numbers on the phrases covered by our patterns and rules. For example, our patterns extract 1637 genitive phrases from the corpus, but only 1137 are processed by our rules. This is due to the fact that not all nouns have been semantically classified by GermaNet. This also applies to the prepositional phrases. Somewhat different are the NE's, since here we do not have any patterns and the rules are directly applied on the corpus. For premodification, the difference in numbers is given by the fact, that we worked with the top 100 most frequent adjectives and adverbs. A second reason is also the fact that not all adverbs have a semantic connotation.

Phrase type	Coverage by Pattern	Coverage by Rule
Genitive phrase	1637	1137
Prepositional phrase	2546	1683
Named entity	-	7812
Premodification phenomena	10529	2965

Table 5.19: Some numbers on the recall of the patterns for ontology extraction.

As a side effect from our morphological analyzer, which also provides NE recognition, we additionally performed ontology population. In fact, we populated based on NE recognition and instantiated the generic classes person, location, organization, quantity, date and money.

This section covered, in fact, the extraction of ontological knowledge between single lexical units. In the next section we will show the potential for the extraction of ontological knowledge from predicate-argument structures. What we intend here is to extract ontological knowledge from bigger linguistic units by using the arguments of the Verb (V).

5.3 Grammatical Functions Layer

In this section we describe the extension of the already existing ontology extraction rules with new rules from predicate-argument structures. By using predicate-argument structures, we discover new ontological relations between the arguments of the predicate.

We first describe the generic rule for extracting ontological knowledge from predicate-argument structures. This generic rule we apply to all verbs in the corpus. In this way we ensure that we cover all relevant information from predicate-argument structures. In the next step we elaborate the generic extraction rules for the top ten most frequent finite verbs in our corpus. As a starting point for these specific rules we use the semantic classification of verbs provided by Schumacher (1986). We decided to use this semantic classification of verbs, instead of the one provided by GermaNet, because we considerer it more complete in the sense that it offers for each verb the arguments which have to be filled. For example the verb *geben* (*give*) can have one argument, a direct object, or three arguments, a subject, a direct object and an adjunct. If it appears with only a direct object Schumacher (1986) classifies *geben* as a verb expressing stative existence. When it appears with thee arguments, the verb expresses transfer of possession. And exactly these aspects we intend to cover by these rules.

5.3.1 Phrase Structure and Syntactic Information

The linguistic analysis on which the extraction rules from grammatical functions rely is provided by SCHUG (Declerck, 2002). Shallow and Chunk Based Unification Grammar (SCHUG) is a rule-based chunk parser which provides for each phrase syntactic information. Based on it we developed the generic extraction rule in Figure 5.32. The generic rule extracts for each finite verb[10] in the corpus its arguments[11].

```
ARG[GF=SUBJ] +
VG[FORM=finite] +
ARG[GF=DOBJ] +
(ARG[GF=IOBJ] |
ARG[GF=PP_ADJUNCT])? +
==>VG(SUBJ, DOBJ, (IOBJ/PP_ADJUNCT)?)
```

Figure 5.32: Pattern for the extraction of ontology schema components from grammatical functions.

For a more detailed semantic analysis of the verbs, we combine the result of the syntactic analysis provided by SCHUG with the semantic classification of verbs by Schumacher (1986) and GermaNet. In this way we are able to discover new relations and increase our ontological knowledge base.

We start this more detailed rule development by looking for the most frequent verbs in the corpus. In order to extract the rules from predicate-argument structures we concentrate on the top 10 verbs in our corpus. Table 5.20 lists the top 10 most frequent verbs[12] in our corpus[13]. In this section we concentrate on the verb *geben* (*to give*), since it occurs most frequently in our corpus[14].

[10] Verbs are marked in SCHUG as Verb Group (VG).

[11] We are interested in finite verbs which have at least two arguments. Unary relations are not interesting when building an ontology.

[12] Appendix A.6 lists all verbs which appear more than thirty times in or corpus.

[13] For the remaining verbs, we decided to introduce the verbs as relations into the ontology.

[14] Appendix B.5 lists the extraction rules for the top ten most frequent verbs.

Compound	Relation
geben	179
liegen	177
gehen	175
kommen	174
stehen	158
machen	143
gelten	143
sehen	116
bleiben	114
setzen	74

Table 5.20: Top 10 most frequent verbs in the corpus.

Both GermaNet and Schumacher (1986) assign *geben* into more than one semantic class: stative existence, change, communication, change of possession. From the semantic perspective we also need the semantic information about nouns in order to know where to introduce the new relation into the ontology. As before, for the classification of nouns we use GermaNet. From Schumacher (1986) we use the information about the semantic class of a given verb. Depending on its semantic class the verb introduces different relations between its arguments.

Figure 5.33 shows the extraction rule for the verb *geben* when it appears with only one argument and denotes stative existence. The verb has here just one argument, the direct object. The introduced relation is an unary existence relation which is not bad, but from the perspective of ontology building it does not bring much.

(14) Es gibt viele Jobs.

There are many jobs.

From example 14 we are able to extract `exist(Job)`, but without further information, this relation is useless. Of course, we could introduce the class *Job* into the ontology, but only for this finding we do not need to look for the arguments of the verbs in our corpus.

```
es[PoS=pron]
+ VG[STEM=geben]
+ phrase[GF=DOBJ]
==>exists(DOBJ)
```

Figure 5.33: *Geben* as a verb denoting stative existence.

The `exist` relation makes sense only if *geben* allows two arguments, as shown in Figure 5.34. Here the verb *geben* has two arguments, a direct object and a Prepositional Adjunct (PP_ADJUNCT)[15]. With this rule we can cover constructions like that in Example 15.

(15) Es gibt viele Jobs in Brüssel.

There are many jobs in Bruxelles.

No matter what the semantic class of the noun is, the semantics of the verbs introduces an existence relation between the arguments of *geben*: `exist(Job, Brüssel)`.

```
es[PoS=pron]
+ VG[STEM=geben]
+ phrase[GF=DOBJ]
+ phrase[GF=PP_ADJUNCT]
==> exists(DOBJ, PP_ADJUNCT)
```

Figure 5.34: *Geben* denoting existence.

According to Schumacher (1986) the verb *geben* also denotes change, if it is followed by a change Noun (N) such as *Veränderung* (*transformation*) or *Änderung* (*change*) (see Example 16). Of course, these nouns denote a change, but in fact the relation introduced by *geben* is also an existence relation. The rule depicted in Figure 5.35 specifies the one in Figure 5.34.

[15] The prepositional adjunct may also appear before the verb, but the introduced relation won't change.

(16) Es gibt eine Veränderung am Markt.

There are changes on the market.

Schumacher (1986) also classifies the verb *geben* as belonging to the class of verbs denoting communication, such as *informieren* (*to inform*) or *melden* (*to announce*). According to Schumacher (1986), *geben* denotes information only if it is followed by an direct object whose head noun has been identified as denoting communication. Example 16 contains such a construction.

```
es[PoS=pron]
+ VG[STEM=geben]
+ phrase[GF=DOBJ][SC=change]
+ phrase[GF=PP_ADJUNCT]
==> exists(DOBJ, PP_ADJUNCT)
```

Figure 5.35: *Geben* denoting change.

The entire verb argument construction denotes that some information is known about a person, which means that some information indeed exists about this person.

(17) Es gibt Informationen über den neuen Chef.

There is information about the new boss.

As for the example above, the rule in Figure 5.36 specifies the one in Figure 5.34. The difference between the more general and the more specific rule is that, for the more specific rule, the semantic class of the direct object is constrained.

```
es VG[STEM=geben]
+ phrase[GF=DOBJ][SC=information]
+ phrase[GF=PP_ADJUNCT]
==> exists(DOBJ, PP_ADJUNCT)
```

Figure 5.36: *Geben* denoting communication.

If the *es gibt* constructions above can be reduced to just one extraction rule, the constructions presented in 18 and 19 need to be handled separately. Both sentences deal with a change of possession. In 18 the Subject (SUBJ), *Chef* (*boss*), gives the Indirect Object (IOBJ), *Bonus* (*bonus*), to the Direct Object (DOBJ), here *Mitarbeiter* (*employees*). Here the *Bonus* is the object which belongs to the *Chef*. After the action of giving is completed the *Bonus* belongs to the employees. We have to notice here that only things which can be touched and are visible and real can be the object of possession change. This is also the reason why the rule for extracting this kind of ontological knowledge (see Figure 5.37) specifies a semantic constraint on the semantic class of the direct object.

(18) Der Chef gab seiner Mitarbeiter den Bonus.

The boss gave his employees the bonus.

Based on the extraction rule we are able to extract the relation between the verb *geben* and its arguments, actually the head nouns of its arguments: `possessionChange(Chef, Mitarbeiter, Bonus)`. We have to notice here, that the nouns also enter the ontology, as described in detail in the previous sections, as subclasses of the semantic classes they belong to. Concerning the OWL formalization of this relation, we use reified relations here, since RDF, respectively OWL, can handle only binary relations.

In Example 19 we have the same construction as in 18, but we still cannot say that this is a change of possession.

(19) Der Chef gab seiner Mitarbeiter ein Rat.

The boss gave his employees an advice.

In this example the boss gives an advice, something which is not palpable and therefore cannot be really possessed. In such a case we introduce a new relation,

```
phrase[GF=SUBJ][SC=person]
+ VG[STEM=geben]
+ phrase[GF=IOBJ][SC=person]
+ phrase[GF=DOBJ][SC=object]
==>changePossessionRelation(SUBJ, DOBJ, IOBJ)


phrase[GF=SUBJ][SC=person]
+ VG[STEM=geben]
+ phrase[GF=DOBJ][SC=object]
+ phrase[GF=IOBJ][SC=person]
==>changePossessionRelation(SUBJ, DOBJ, IOBJ)
```

Figure 5.37: *Geben* denoting change of possession.

the `giveRelation`. The rule for handling such cases is depicted in Figure 5.38. It is the same rule as in Figure 5.37, with a small difference: the direct object denotes something abstract, which is not palpable. From Example 19 we extract `giveRelation(Chef, Mitarbeiter, Rat)`.

```
phrase[GF=SUBJ][SC=person]
+ VG[STEM=geben]
+ phrase[GF=IOBJ][SC=person]
+ phrase[GF=DOBJ][SC=abstract]
==>giveRelation(SUBJ, DOBJ, IOBJ)


phrase[GF=SUBJ][SC=person]
+ VG[STEM=geben]
+ phrase[GF=DOBJ][SC=abstract]
+ phrase[GF=IOBJ][SC=person]
==>giveRelation(SUBJ, DOBJ, IOBJ)
```

Figure 5.38: *Geben* denoting giving.

5.3.2 Summary from the Grammatical Functions-Based Processing

In this section we have shown how ontology extraction is performed on the basis of predicate-argument structures. We have to notice here that the presented rules are not exhaustive and cover phenomena which appear in our corpus. For this purpose we first syntactically annotated the corpus and filtered from the 908 finite verbs in the corpus, the top ten most frequent verbs. Then we semantically annotated these verbs and developed, based on Schumacher (1986) and on our observations from the predicate-argument structures, the rules for ontology extraction. The rules themselves apply to phrases, but the resulting relations connect the head nouns of the phrases.

A remark has to be made on the semantic classification of verbs proposed by Schumacher (1986). We did not just rewrite the verb classification of Schumacher (1986) and transform it into rules. The role of Schumacher (1986) was to guide us when choosing the semantic relation introduced by the verb. We adapted and enlarged it in order to cover as many phenomena as possible.

5.4 Conclusion

In this chapter we present a detailed description of the developed rules and their application for the extraction of ontological knowledge. The chapter is divided into three sections, which correspond to the three linguistic analysis levels form which we extract ontological knowledge.

Section 5.1 and Section 5.2 described the potential for ontology extraction from plain text, respectively from text annotated with PoS and lexical semantics. Although we were aware of the fact that predicate-argument structures can contribute to the process of extracting ontological knowledge, Section 5.1 and Section 5.2 demonstrated that ontology classes and relation can also be extracted without necessary using deep linguistic analysis. Section 5.2 covered the extraction potential from PoS and semantically annotated text. As a side-effect and without any additional effort Section 5.2 described also the ontology population process based on the NE recognition component incorporated in the morphological analyzer. Section 5.3 described the extraction of ontological knowledge from predicate-argument structures enlarging not only the covered linguistic phenomena, but also the range of extracted relations.

We close this chapter by saying that it described in detail the rules and their application for the extraction of ontological knowledge from three layers: plain text, PoS and semantically annotated text and from predicate-argument structures.

Chapter 6

Representing the Extracted Schema Components with OWL

In this chapter we describe the representation of the extracted ontological knowledge. The representation follows the W3C Recommendation for OWL, the Web Ontology Language, respectively OWL DL. We decided to use OWL DL because it is the OWL variant which fits best to our needs concerning the representation of the extracted ontological knowledge. OWL is widely used and accepted for such formalizations. Consequently a lot of the available upper and domain ontologies are represented in OWL DL.

This chapter is divided into two sections: Section 6.1 describes the DL and the corresponding OWL DL constructs and Section 6.2 is based on examples of how the extracted ontological knowledge is formalized with OWL DL.

6.1 DL and OWL DL

OWL is based on RDF, RDF Schema and XML Schema (XSD) datatypes. The basic RDF data model contains resources, properties and statements[1]. Resources are the things described by RDF expressions (concepts in DL). A property in RDF (roles in DL) is a specific aspect, a characteristic, an attribute, or relation used to describe a resource[2]. A specific resource together with a named property plus the value of that property for that specific resource represent a RDF statement[3]. The three elements of a RDF statement are also called subject, predicate and object. From that we can conclude that RDF can express binary relations, also called subject-predicate-object triples. We notice here that OWL makes use of a slightly different vocabulary as RDF and DL. DL roles and RDF predicates are called properties, the DL concepts and RDF subject and object are classes (written as `owl:Class`) which are instantiated with individuals. Instantiations can be either written by using `rdf:type` or `rdf:about`.

OWL Axiom	**DL Syntax**
`rdfs:subClassOf`	$C \sqsubseteq D$
`owl:equivalentClass`	$C \equiv D$
`owl:disjointWith`	$C \sqsubseteq \neg D$
`owl:sameAs`	$\{a\} \equiv \{b\}$
`owl:differentFrom`	$\{a\} \sqsubseteq \neg\{b\}$
`rdfs:subPropertyOf`	$R \sqsubseteq S$
`rdfs:equivalentProperty`	$R \equiv S$
`owl:inverseOf`	$R \equiv S^-$
`owl:transitiveProperty`	$R^+ \sqsubseteq R$
`owl:symmetricProperty`	$R \equiv R^-$

Table 6.1: OWL DL axioms.

[1] Resource Description Framework (RDF) Model and Syntax Specification, http://www.w3.org/TR/PR-rdf-syntax/

[2] http://www.w3.org/TR/PR-rdf-syntax/

[3] http://www.w3.org/TR/PR-rdf-syntax/

OWL Constructor	**DL Syntax**	**Name**
`owl:Thing`, `owl:Nothing`	$\top$, $\bot$	universal and bottom concept
`owl:complementOf`	$\neg A$	negation
`owl:intersectionOf`	$C \sqcap D$	intersection
`owl:unionOf`	$C \sqcup D$	union
`owl:oneOf`	$\{a_1, \ldots, a_n\}$	enumeration
`owl:allValuesFrom`	$\forall R.C$	value restriction
`owl:someValuesFrom`	$\exists R.C$	existential quantification
`owl:hasValue`	$\exists R.b$	exist. quant. with
restricted value		
`owl:maxCardinality`	$\geq nR$	cardinality restriction
`owl:minCardinality`	$\leq nR$	
`owl:cardinality`	$= nR$	
`owl:maxCardinality` + `owl:valuesFrom`	$\geq nR.C$	qualified cardinality restriction
`owl:minCardinality` + `owl:valuesFrom`	$\leq nR.C$	
`owl:cardinality` + `owl:valuesFrom`	$= nR.C$	

Table 6.2: OWL DL constructors.

Another difference to DL and RDF consists of the fact that OWL makes the distinction between object properties and datatype properties. Object properties (written as `owl:objectProperty`) relate two individuals, whereas datatype properties relate individuals to data types. Datatype properties (written as `owl:datatypeProperty`) may range over RDF literals or simple types defined in accordance with the XML Schema datatypes. XSD data types are predefined data types such as string, integer or boolean[4].

OWL allows for the representation of restrictions. The most common one is the restriction of the property by defining a domain (`rdfs:domain`) and range (`rdfs:range`). A different kind of restriction is that applied directly on properties: such as `owl:allValuesFrom`, `owl:someValuesFrom`, `owl:cardinality` and `owl:hasValue`. The restriction is written as `owl:Restriction` and `owl:onProperty`

[4] Complex data types can be build by using constructors such as enumeration.

indicates the restricted property.

The origin of the different constructs within OWL can be determined from the namespace prefixes. An RDF construct is prefixed with `rdf:`, an Resource Description Framework Schema (RDFS) construct with `rdfs:` and a XSD datatype declaration with `xsd:`.

Concerning the DL axioms, Table 6.1 shows the OWL equivalents. One of the most common axioms is the subsumption axiom between classes written as `rdfs:subClassOf`. Table 6.2 lists the OWL DL constructs and the corresponding DL syntax.

6.2 OWL DL Representation of the Ontology Schema Components

In this section we present the OWL DL formalization of the extracted ontological knowledge. In order to demonstrate the OWL DL formalization, we choose the sentence in Example 20.

(20) [Der größte deutsche Chemiekonzern BASF][gf=subj] verdiente [in den ersten neun Monaten][gf=pp_adjunct] [17 Millionen][gf=dobj].

The biggest German chemical concern earned in the first nine months 17 million.

We start by representing the relation introduced by the verb *verdienen* (*earn*)[5] In order to represent *verdienen* in OWL we need the ontological reified relations[6],

[5] We use the notation `relationName(Class1, Class2)`, introducing domain and range restrictions. The application domain is always the superclass of *Class1*. The range is the superclass of *Class2*.

[6] The ontological reified relations presented here are in fact ontology patterns which are different from the reified relations defined in RDF. The latter applies when a RDF Statement=(subject, predicate, object) is included into another Statement.

since *verdienen* has as arguments the subject *der größte deutsche Chemiekonzern* (*the biggest German concern*), the direct object *17 Millionen* (*17 Million*) and the indirect object *in den ersten neun Monaten* (*in the first nine months*). The relation we extract here is `hasEarning(Chemiekonzern, Million, Monat)` instantiating the generic relation `earn(SUBJ, DOBJ, IOBJ)`. We have to notice here, that by this relation the head nouns of the phrases are connected with each other. By using reified relations, we are able to represent predicates such as *verdienen* which might have more than two arguments: `hasEarning(Chemiekonzern, EarningRelation)` with domain *Group* and range *EarningRelation*, `hasEarningValue(EarningRelation, Million)` with domain *EarningRelation* and range *Number* and `hasEarningTime(EarningRelation, Monat)` with domain *EarningRelation* and range *TemporalUnit*.

In order to represent these relations in OWL DL, we need to introduce the generic classes *Group*, *Relations*, *Number* and *TemporaUnit* into the ontology. These classes are superclasses of the classes *Chemiekonzern*, *EarningRelation*, *Million* and *Monat*. Figure 6.1 and Figure 6.2 depict the corresponding OWL representation. We use here the `allValuesFrom` restriction in order to restrict the fact that each instance of *Chemiekonzern* needs an earning specification. We also restricted the reified relation `hasEarningTime` and `hasEarningValue` which means that at a given point in time, the earning must be unique (i.e., be a functional property, see Figure 6.2).

By applying the extraction pattern on the phrase *größte deutsche Chemiekonzern* we obtain `hasAffiliation(Chemiekonzern, Deutsch)` with domain *Group* and range *Affiliation* and `hasDimension(Deutsche Chemiekonzern, Größte)` with domain *Affiliation* ∪ *Group* and range *Dimension*.

In a similar way to the verb *verdienen* we decided to represent the adjectives by reified relations. As a consequence, the arguments of the relation `hasAffiliation` change to `hasAffiliation(Chemiekonzern, AffiliationRelation)` with do-

```
 <owl:Class rdf:about="#Chemiekonzern">
    <rdfs:subClassOf rdf:resource="#Konzern"/>
    <rdfs:subClassOf><owl:Restriction>
            <owl:onProperty rdf:resource="#hasEarning"/>
            <owl:allValuesFrom rdf:resource="#EarningRelation"/>
    </owl:Restriction></rdfs:subClassOf>
    <rdfs:subClassOf><owl:Restriction>
            <owl:onProperty rdf:resource="#hasAffiliation"/>
            <owl:allValuesFrom rdf:resource="#AffiliationRelation"/>
    </owl:Restriction></rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Million">
 <rdfs:subClassOf rdf:resource="#Number"/>
</owl:Class>
<owl:Class rdf:about="#Monat">
 <rdfs:subClassOf rdf:resource="#TimeUnit"/>
</owl:Class>
<owl:Class rdf:about="#EarningRelation">
    <rdfs:subClassOf rdf:resource="#Relation"/>
    <rdfs:subClassOf><owl:Restriction>
            <owl:onProperty rdf:resource="#hasEarningValue"/>
            <owl:someValuesFrom rdf:resource="#Number"/>
    </owl:Restriction></rdfs:subClassOf>
    <rdfs:subClassOf><owl:Restriction>
            <owl:onProperty rdf:resource="#hasEarningTime"/>
            <owl:someValuesFrom rdf:resource="#TimeUnit"/>
    </owl:Restriction></rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Group">
 <rdfs:subClassOf rdf:resource="&owl;Thing"/>
</owl:Class>
<owl:Class rdf:about="#Number">
 <rdfs:subClassOf rdf:resource="&owl;Thing"/>
</owl:Class>
<owl:Class rdf:about="#Relation">
 <rdfs:subClassOf rdf:resource="&owl;Thing"/>
</owl:Class>
<owl:Class rdf:about="#TimeUnit">
 <rdfs:subClassOf rdf:resource="&owl;Thing"/>
</owl:Class>
<owl:Class rdf:about="&owl;Thing"/>
```

Figure 6.1: The OWL representation of the classes from sentence 20.

main *Group* and range *AffiliationRelation* and `hasAffiliationValue(AffiliationRelation, Deutsch)` with domain *AffiliationRelation* and range *Affiliation*. The same principle applies also to `hasDimension(deutsche Chemiekonzern, Größte)` resulting

```
<owl:ObjectProperty rdf:about="#hasEarning">
    <rdfs:range rdf:resource="#EarningRelation"/>
    <rdfs:domain rdf:resource="#Group"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#hasEarningTime">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="#EarningRelation"/>
    <rdfs:range rdf:resource="#TimeUnit"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#hasEarningValue">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="#EarningRelation"/>
    <rdfs:range rdf:resource="#Number"/>
</owl:ObjectProperty>
```

Figure 6.2: The OWL representation of the properties from sentence 20.

in the relations `hasDimension(Deutsche Chemiekonzern, DimensionRelation)` with domain *Group* and range *DimensionRelation* and `dimensionRelationValue(DimensionRelation, Größte)` with domain *DimensionRelation* and range *Dimension*.

For the classes *Chemiekonzern*, *Größte* and *Deutsch* everything remains unchanged, they enter the ontology as subclasses of the generic classes *Group*, *Dimension* and *Affiliation*. We decided to introduce *Deutsch* and *Größte* as classes, and not as instances, into the ontology, because we have decided to view adjectives as classes. The OWL representation of the ontological knowledge from the phrase *größte deutsche Chemiekonzern* is depicted in Figure 6.3 and Figure 6.5. Figure 6.3 contains the OWL representation of the classes described above, whereas Figure 6.5 contains the relations between these classes. Additionally, from the compound *Chemiekonzern*, we are able to extract the `subClassOf` relation between *Chemiekonzern* and *Konzern*, where the class *Konzern* is a subclass of the generic class *Group*.

```
<owl:Class rdf:about="#Affiliation"><rdfs:subClassOf rdf:resource="&owl;Thing"/></owl:Class>
<owl:Class rdf:about="#AffiliationRelation">
    <rdfs:subClassOf rdf:resource="#relation"/>
    <rdfs:subClassOf><owl:Restriction>
            <owl:onProperty rdf:resource="#hasAffiliationValue"/>
            <owl:someValuesFrom rdf:resource="#Deutsch"/>
    </owl:Restriction></rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Deutsch"><rdfs:subClassOf rdf:resource="#Affiliation"/></owl:Class>
<owl:Class rdf:about="#Chemiekonzern">
    <rdfs:subClassOf rdf:resource="#Konzern"/>
    <rdfs:subClassOf><owl:Restriction>
            <owl:onProperty rdf:resource="#hasEarning"/>
            <owl:allValuesFrom rdf:resource="#EarningRelation"/>
     </owl:Restriction></rdfs:subClassOf>
     <rdfs:subClassOf><owl:Restriction>
            <owl:onProperty rdf:resource="#hasAffiliation"/>
            <owl:allValuesFrom rdf:resource="#AffiliationRelation"/>
     </owl:Restriction></rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#DeutscheChemiekonzern">
    <rdfs:subClassOf rdf:resource="#Chemiekonzern"/>
    <rdfs:subClassOf><owl:Restriction>
            <owl:onProperty rdf:resource="#hasDimension"/>
            <owl:allValuesFrom rdf:resource="#DimensionRelation"/>
        </owl:Restriction></rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Dimension"><rdfs:subClassOf rdf:resource="&owl;Thing"/></owl:Class>
<owl:Class rdf:about="#DimensionRelation">
    <rdfs:subClassOf rdf:resource="#Relation"/>
    <rdfs:subClassOf><owl:Restriction>
            <owl:onProperty rdf:resource="#hasDimensionValue"/>
            <owl:someValuesFrom rdf:resource="#gr&#246;&#223;te"/>
    </owl:Restriction></rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Größte"><rdfs:subClassOf rdf:resource="#Dimension"/></owl:Class>
```

Figure 6.3: The OWL formalization of the ontology classes extracted from the phrase *größte deutsche Chemiekonzern.*

```
<owl:DatatypeProperty rdf:about="#hasNumberValue">
    <rdfs:domain rdf:resource="#Number"/>
    <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasTimeUnitValue">
    <rdfs:domain rdf:resource="#TimeUnit"/>
    <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
<Million rdf:about="#17"/>
<Monat rdf:about="#neun"/>
```

Figure 6.4: The OWL formalization of the two datatype properties `hasTimeUnitValue` and `hasNumberValue` and two instances for *17 Millionen* and *neun Monate.*

From *Chemiekonzern BASF* we are able to instantiate the class *Chemiekonzern* with *BASF*.

```
<owl:ObjectProperty rdf:about="#hasAffiliation">
    <rdfs:range rdf:resource="#affiliationRelation"/>
    <rdfs:domain rdf:resource="#group"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#hasAffiliationValue">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:range rdf:resource="#Affiliation"/>
    <rdfs:domain rdf:resource="#AffiliationRelation"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#hasDimension">
    <rdfs:domain rdf:resource="#Affiliation"/>
    <rdfs:range rdf:resource="#DimensionRelation"/>
    <rdfs:domain rdf:resource="#Group"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#hasDimensionValue">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:range rdf:resource="#Dimension"/>
    <rdfs:domain rdf:resource="#DimensionRelation"/>
</owl:ObjectProperty>
```

Figure 6.5: The OWL domain and range restrictions of ontological relations extracted from the phrase *größte deutsche Chemiekonzern*.

The ontological knowledge extracted from the phrases *neun Monate* and *17 Millionen* is depicted in Figure 6.4. For representing them, we use the OWL datatype properties `hasTimeUnitValue(Monate, neun)` with domain *TimeUnit* and range *string* and `hasNumberValue(Million, 17)` with domain *Number* and range *integer*.

Figure 6.6 depicts the graph resulted from the extraction of ontological knowledge from sentence 20. Because the graphical representation was generated by Protégé, the `subClassOf` relation is depicted by the `is-a` relation. Appendix C.2 lists the representation of the extracted ontological knowledge from Example 20 in OWL's functional syntax. Appendix C.3 lists the extracted ontological knowledge from Example 20 as logical formulas.

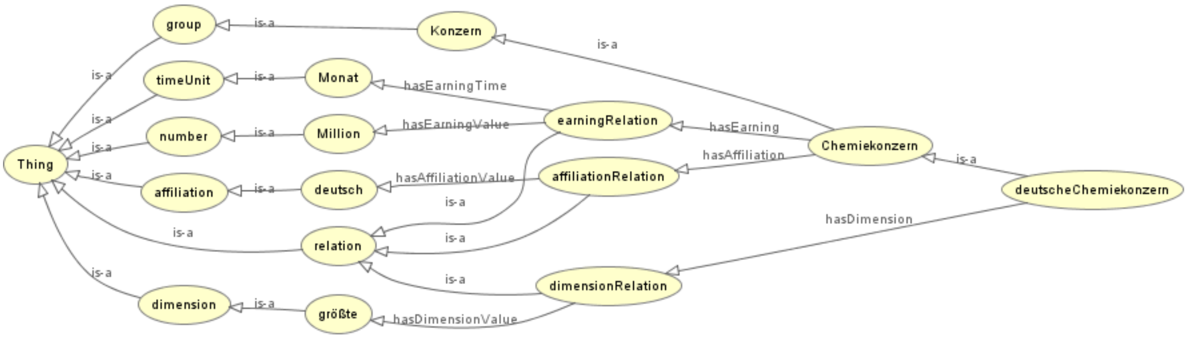


Figure 6.6: Graphical representation of the ontological relations extracted from Example 20.

6.3 Conclusion

In this chapter, we described the representation of the ontological knowledge with OWL DL. We decided to use OWL DL not only because it has evolved into a standard for ontology formalization, but it also allows for expressing the `unionOf` restriction on properties. If we did not need these restriction, OWL Lite would fit our needs too. In order for our ontology to be used by others, an ontology should be decidable, should terminate[7]. Ontologies represented both by OWL Lite or OWL DL are decidable. In Section 6.1 we presented the reltionship between DL and OWL DL by discussing the same constructs in DL and OWL. Section 6.2 deals with the OWL formalization of the extracted ontological knowledge.

We have to remark here that the use of OWL for knowledge representation makes it easier, syntactically, to integrate our work into existing or future ontologies. The OWL representation is not necessary for merging two ontologies, since extracted knowledge can be easily converted into the desired format. In order to integrate our work into existing ontologies, we need a good ontology alignment. By ontology alignment, we mean T-Box alignment. To realize a good ontology alignment, we need common entry points between the two ontologies. In Sec-

[7] The algorithm to determine whether a statement is true.

tion 8.2 we show the anchor points for integrating the extracted knowledge into an existing ontology. In this case, we decided that the anchor points are general classes which appear in both ontologies.

Chapter 7

Extending the Applicability of our Approach

In this chapter, we describe how our method can be applied to French text and to a different domain, the radiology domain. This chapter we consider as a good guide for further work since the application to other languages and domains is of general use in the ontology learning area.

Section 7.1 describes the application of the rules for the detection of ontology schema components from French text. Section 7.2 describes the application of the rules for the detection of ontology schema components from a radiology corpus.

7.1 Application of the Method on French Text

This section sketches how the ontology extraction rules described in this thesis could be applied to French text. Although compounding in French is different from compounding in German, we outline to what extent our approach might be applied to French compounds[1].

7.1.1 Extraction of Ontology Schema Components from French Compounds

In contrast to the German compounds, the French compounds are not always conflated to a single word. The cumulated form of compounds such as *sociolinguistique* are in French the exception. The majority of compounds in French consist either of two components connected by a hyphen such as *timbre-poste* (*stamp*) or are just two or more words which appear in a lexical chain such as *dessin animé* (animated cartoon) or *séance marathon* (*marathon session*). The most productive of the latter compounds are the compounds constructed with prepositions such as *mesure de sécurité* (*safety measure*)[2]. Concerning the PoS of constituents of the French compounds, Stein (2005) differentiates between compounds built of two nouns such as *taxi-camionnette* (*taxi pick-up truck*), of an adjective and a noun *table ronde* (*round table*), of a preposition and a noun or two *café en poudre* (*instant coffee*) and the compounds built of a nominalized verb and a noun *cure-dent* (*toothpick*).

Thiele (1993) classifies compounds from a different perspective. He differentiates between copulative and determinative compounds. As for German, copulative compounds are compounds were the elements are considered semantically coequal and which do not have a main element which specifies or determines the other element in the compound. According to Thiele (1993), for French, the relation between the elements of a copulative compound rely on an additive relation. For example, *taxi-camionnette* (*taxi and pick-up truck*) is at the same time a *taxi* and

[1] The extraction of ontological knowledge from French compounds assume a morphological analyzer for the compound recognition and the semantic classification of adjectives.

[2] Noun-noun compounds are in French less frequent than in German or English (Geckeler and Dietrich, 2007)

a *camionnette* (*pick-up truck*). Copulative compounds in French consist of two nouns which are either connected by a hyphen or are conflated. As for German, the determinative compounds contain an element which specifies and an element which is specified. But unlike in German, in French the second element of the compound specifies the first one (Thiele, 1993).

In the following sections, we sketch how the developed rules for the extraction of ontological knowledge can be applied to French text.

Compounds Consisting only of Nouns

Compounds consisting of two nouns in French are either copulative or determinative compounds. Concerning the form of this compounds, the two nouns[3] are either connected by a hyphen or not. Because for French, as for German, a morphological analysis tool is not able to make the distinction between copulative and determinative compounds, we are not able to distinguish which relation applies. This is the reason why we cannot simply handle compounds consisting of two nouns.

If we were able to distinguish between copulative and determinative compounds, the following ontological knowledge could be extracted from hyphen compounds. So for example from the compound *général-président* (*president general*), which is considered a copulative compound, we could extract the classes *Général* and *Président* and the relation `isCoordinatedWith(Général, Président)`. Compounds such as *chou-fleur* (*cauliflower*), *ingénieur-électronicien* (*electronical engineer*) and *wagon-restaurant* (*dining car*), which consist of two nouns, are determinative compounds in which the first noun is the main element which is made more specific by the second noun (Thiele, 1993). From this type of construction, we extract the classes *Wagon-Restaurant* and *Wagon* and the relation

[3] Compounds consisting of three nouns are considered occasional constructions.

`subClassOf(Wagon-restaurant, Wagon)`.

Similar to the hyphen compounds, we can handle the compounds consisting of two nouns separated by a space such as *femme ingénieur* (*woman engineer*) and *voiture sport* (*sports car*). From them we can extract the `subClassOf` relation: `subClassOf(Femme ingénieur, Femme)` and `subClassOf(Voiture sport, Voiture)`.

Compounds Consisting of a Noun and an Adjective

Compounds consisting of a noun and an adjective appear very often in French. In such compounds, the adjective limits the noun. This type of compound corresponds to the German construction `premodifier+noun`. Such compounds are *blouson noir* (*black jacket*), *table ronde* (*round table*) and *espace cosmique* (*outer space*) (Thiele (1993)). This compounds are seldom connected by a hyphen, and if so, they are in most of the cases lexicalized: *procès-verbal* (*protocol*), *fer-blanc* (*tin*) (Thiele, 1993). Our rules do not specify this type of modification This means that in order to extract the ontological knowledge from this type of compounds, the adjective has to be classified semantically. So for example, from *blouson noir* (*black jacket*) we could extract the class *Blouson* and the relation `hasProperty(Blouson, Noir)`. The adjective *noir* could enter the ontology as a subclass of the generic class *Colour*.

The compounds constructed from a noun followed by an adjective can also be extended by an additional adjective[4] such as *grande propriété* (*big estate*) and *grande propriétaire foncière* (*rich landowner*). In this case the adjective *grande* (*big*) limits the compound *propriété foncière* (*estate*) and *foncière* (*ground*) specifies *propriété* (*property*). From this compound we are able to extract the following ontological knowledge: `hasDimension(Propriété foncière, Grande)` and

[4] Thiele (1993)

`hasReference(Propriété, Foncière)` since the adjective *grande* would be classified as a dimensional adjective and *foncière* as a referential adjective. The two adjectives enter the ontology as subclasses of the classes *Dimension* and *Reference.*

In French compounds, the adjective can also appear in front of the noun, such as in *bon sens* (*common sense*), *franc-maçon* (*freemason*) and *plein air* (*outdoor*). According to Thiele (1993) this type of compound corresponds to old French language and are, in most of the cases, lexicalized.

Prepositional Compounds

Compounds consisting of two nouns connected by a preposition are very frequent in French. Such compounds are *chemin de fer* (*rail*), *avion à réaction* (*jet fighter*) and *café en poudre* (*instant coffee*) (Thiele (1993)). The prepositions mainly used for this type of compounds are *de*, *à* and *en* and sometimes the preposition is followed by an article such as in *maison de la culture* (*forcing house*). This type of compound corresponds in fact to the German paraphrases of compounds.

From the semantic point of view, the second noun in the compound limits the first one (Thiele, 1993), which means that main element of the compound is the first noun. Concerning the extraction of ontological knowledge from this type of compound, we extract the `subClassOf` relation. In order to determine the relation between the two noun components of the compound we could apply here the rules developed for the extraction of ontological knowledge from paraphrases in Section 5.2. These ontology extraction rules match here because the relations between the two nouns in the compound depend on the semantic classification of those nouns and prepositions. Since a preposition can express more than one relation (Thiele, 1993), we need the semantic classification of nouns.

For example, from *directeur de la banque* (*bank director*) we could extract `subClassOf(Directeur, Directeur de la banque)`. By semantically classifying the two nouns *directeur* and *banque* as *Person* and *Group* we would extract `hasPosition(Banque, Directeur)` with domain *Person* and range *Group*. *Directeur* will enter the ontology as a subclass of the generic class *Person* and *Banque* will enter the ontology as a subclass of the generic class *Group*.

We conclude this section by observing that the ontology extraction rules presented in the previous chapters can be applied to French text. Besides applying the rules for the ontology extraction from compounds, postmodification phenomena and paraphrases, we also sketched a possibility of handling compounds consisting of more than two elements. Another aspect which we deal with in this section are compounds consisting of an adjective and a noun for which we also sketch a possibility of extracting ontological knowledge. We did not deal with the extraction of ontological knowledge from the sentential level. On the sentential level, the process of ontology learning relies, as for German, on the semantics of the verb and its relation to its arguments. From that perspective, we do not expect that French predicate-argument structures are very different from German predicate-argument structures.

7.2 Application to the Radiology Domain

In this section we will show how the ontology extraction rules presented here apply to a corpus from the medical domain, more specifically from the radiology domain. As a corpus, we use the anonymized RadLex copus from the THESEUS MEDICO[5] project. We opted for this corpus because it is totally different from the financial newspaper corpus. The corpus contains the findings and the corresponding evaluation of a radiological examination without using classical sentential constructions. Its particular style makes it difficult to process the tele-

[5] http://www.theseus-programm.de/anwendungsszenarien/medico/default.aspx

graphic phrases by a classical parser. In the following section we describe the ontology extraction potential from compounds and phrases. We also will show why the rules for the detection of ontology schema components from sentences can not be applied to this corpus. In order to apply the method presented in this thesis we use SProUT[6] to annotate our corpus morphologically and with PoS. As a semantic resource we use GermaNet (Kunze and Lemnitzer, 2002).

7.2.1 Extraction of Ontology Schema Components from Compounds

The compounds in the radiology corpus are mostly constructed by a word denoting an organ and a word denoting a phenomenon connected to the specific organ such as *Leberhämatom* (*liver hematoma*), *Leberkarzinom* (*liver carcinoma*), *Leberläsion* (*liver lesion*), *Leberherde* (*liver metastases-like structure*) or *Lebermetastase* (*liver metastases*). Having the morphological analysis for the compounds (provided by SProUT), we are able to automatically detect the elements of the compound: *Leber + Hämatom*, *Leber + Karzinom*, *Leber + Läsion*, *Leber + Metastase.* We assume that all detected compounds are determinative compounds, which means that according to the definition for determinative compounds (Duden, 2006) we are able to extract the `subClassOf` relations `subClassOf(Leberläsion, Läsion)` and `subClassOf(Leberkarzinom, Karzinom)`.

As already described in Chapter 4, analyses of the German compound (Fleischer and Barz, 1995; Lohde, 2006; Motsch, 2006) assume that there is also another type of relation between the two elements of a compound. In order to detect this relation, we look for the paraphrases for those compounds and apply the designed ontology extraction rules. The compounds in the radiology corpus are

[6] http://sprout.dfki.de/

reformulated as *Karzinoms in der Leber* (*carcinome in the liver*), *Läsion der Leber* (*lesion in the liver*), *Zysten in der Leber* (*cyst in the liver*). If we want to apply the already designed rules we need to know the semantic classification of the components of the phrase. The noun *Leber* (*liver*) is classified by GermaNet as belonging to the semantic class *body*. Our set of ontology extraction rules does not contain any rule which implies a noun semantically classified as a *body*. Unfortunately, GermaNet does not have an entry for nouns like *Karzinom* (*carcinoma*), *Läsion* (*lesion*) or *Zyste* (*cyst*) which makes it impossible to apply our rules on these paraphrases. This means that besides GermaNet's shortcoming, we need to define an additional rule for nouns denoting organs or medical conditions in order to be able to extract all ontological knowledge from this type of compound.

An alternative to GermaNet's semantic classification would be to use the RadLex[7] terminology. RadLex provides for each constituent of the compound its specific semantic superclass. The difference between GermaNet and RadLex is that RadLex is indeed a specific terminology for radiology, whereas GermaNet is trying to cover more general aspects of life. In addition, GermaNet provides more semantic relations than the RadLex.

7.2.2 Extraction of Ontology Schema Components from Phrases

In order to extract the ontological knowledge from premodification phenomena, we classified the adjectives and adverbs in this corpus by Lee (1994), respectively Lobeck (2000). As for the financial newspaper corpus, we notice here several types of premodification. The first one, is the construction `adjective+noun` such as *größere Läsionen* (*bigger lesions*) and *kleinere Läsionen* (*smaller lessions*).

[7] http://www.radlex.org/

Since both adjectives, *größ* (*big*) and *klein* (*small*), belong to the semantic class *Dimension* we are able to extract here `hasDimension(Läsion, Größere)` and `hasDimension(Läsion, Kleinere)` with the adjectives *größere* and *kleinere* as subclasses of the generic class *Dimension*. The ontology class *Läsion* (*lesion*) is not covered by GermaNet, but its synonym *Verletzung* is classified by GermaNet as a *state*. From here we can conclude that *Läsion* will enter the ontology as a subclass of the generic class *State*.

The second premodification type concerns the premodification of a noun by two premodfiers: either two adjectives or an adverb followed by an adjective. For the constructions with two adjectives which precede the noun such as *vergrößerte mediastinale Lymphknoten* (*enlarged mediastinaö lymph nodes*) we apply our extraction rule and extract first `hasLocation(Lymphknoten, Mediastinal)` and `hasDimension(Vergrößerte, Mediastinale Lymphknoten)`. If the first premodifier is an adverb such as in *größtenteils progrediente Lymphknoten* (*mostly progredient lymph nodes*), our rule extracts `hasMode(Lymphknoten, Progredient)` and `hasManner(Größtenteils, Progredient)`.

A different type of premodification is the construction where two or more adjectives are connected by comma or/and a conjunction. In this case each premodifier introduces a relation between itself and the noun it modifies. For example from *mediastinale und hiläre Lymphknoten* (*mediastinal and hilar lymph nodes*) our rule extracts `hasLocation(Lymphknoten, Mediastinal)` and `hasLocation(Lymphknoten, Hilär)`.

We can conclude here, that our ontology extraction rules for premodification phenomena can be applied to the radiology domain, but only if the domain specific modifiers are properly semantically classified. We observed also that the radiology reports are not written in sentences, as we are used to. The statements in the radiology corpus are in fact expressions in which the verb is missing but implicitly understood. Such sentences are *Pleuraerguss rechts mehr als links* (*pleura* *contusion more right than left*) or in *Harnblase bei liegendem Blasenkatheter leer* (*bladder with catheter empty*). Another aspect which we notice here is the unusual postmodification of nouns like in *Pleuraerguss rechts* (*pleura contusion right*) and *Flüssigkeit perihepatisch* (*liquid perihepatic*). This kind of phenomena are not covered by our rules, but we could easily adapt the premodification rules (as shown in Section 7.1) to cover this kind of phenomena.

7.3 Conclusion

In this chapter, we have shown how the method presented in the previous chapters can be applied to French text and to a completely different corpus. For French we describe how our designed rules apply to different types of compounds. The application of the rules for the extraction of ontology schema components to the radiology domain has shown that our rules for compounds, paraphrases and modification phenomena apply with some restrictions also to this domain.

Chapter 8

Evaluation of the Extraction Method

In this chapter we present two different ways of evaluating the approach presented in this thesis. Section 8.1 gives an overview on the evaluation methods for ontology extraction. Section 8.2 describes the comparative evaluation (more precisely the possibilities for extending the MUSING company ontology), whereas Section 8.3 shows the results of a quantitative evaluation (by using the F-measure metric).

The MUSING company ontology relies on the Enterprise Ontology[1] which represents a collection of terms and definitions relevant to business enterprises (Bachlechner et al., 2008). The MUSING company ontology imports classes and properties from the NACE and the BACH[2] ontology. NACE[3] is an European industry standard classification system, whereas the BACH ontology relies on the Bank for the Accounts of Companies Harmonised database scheme and is an attempt to allow for interoperability of accounting data at an European level.

[1] http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html

[2] http://ec.europa.eu/economy_finance/indicators/bachdatabase_en.htm

[3] http://ec.europa.eu/competition/mergers/cases/index/nace_all.html

8.1 Methods for Evaluating Ontology Extraction

There are different methods of performing evaluation of ontologies. The first evaluation method is the one presented by Maedche and Staab (2002), which compares two ontologies on the semiotic, syntactic and pragmatic level. The main condition for applying this method is that the two ontologies have been built from similar text or at least model similar ontological knowledge. Only if this condition is fulfilled, can the lexical and taxonomic overlap be measured. Since our method extracts ontological knowledge from financial newspapers and the MUSING company ontology describes a fragment of the economy from the perspective of a single company, we no not have a common basis for performing the evaluation metric proposed by Maedche and Staab (2002).

The second evaluation method is the one adopted by Suchanek et al. (2008). He evaluated his approach manually, by letting human judges decide whether the extracted ontological knowledge, actually a subset of it, is correct or not. This implies that in order to decide if the extracted ontological knowledge is correct, the human judges have to be domain and ontology experts. We could have chosen that method for evaluating our approach, but we could not find the appropriate experts.

The third method, used by Navigli and Velardi (2008), is to manually build a gold standard, which is then compared with the results of the ontology learning approach. This method was adopted in the approach and described in Section 8.3.

8.2 Comparison with the MUSING Ontology

For the extension of the MUSING company ontology, we use version 1.4 published on the 29th of February 2010[4] of the MUSING ontology. The analysis of the the company ontology shows that several classes and relations in the MUSING ontology can be extended with ontological knowledge extracted by the method presented in this thesis[5].

The integration of ontological knowledge into larger ontologies implies common anchor points between the two ontologies. The analysis of the two ontologies has shown, that in our case this anchor points are the general superclasses. From our twenty-three generic classes (see Table A.1 in Appendix A.2) we have found that the following seven *Group*, *Person*, *Location*, *Event*, *Attribute*, *Object* and *Abstract* are also appearing in the MUSING ontologies. The classes *Person* and *Group* are the ones which offer the biggest potential for extension. In MUSING the *Person* class has the following structure (see Figure 8.1): it has three subclasses *Customer*, *Partner* and *Vendor*. The class *Customer* has as subclass the class *Reseller* whereas the class *Vendor* has the two subclasses *Competitor* and *Supplier*.

With our method, we are able to extend this class with new subclasses such as *Chef* (*chief*), *Direktor* (*manager*), *Mitarbeiter* (*employee*), *Leiter* (*leader*), *Angestellte* (*employee*), *Sprecher* (*spokesman*), *Minister* (*minister*), *Präsident* (*president*). This classes we can introduce directly under *Person*, but by using GermaNet's information about hyperonyms we can make the class *Person* more fine-grained concerning its structure. For example, between the class *Direktor* and the class *Person* GermaNet build the following path: *Direktor-Leiter-*

[4] http://www.musing.eu/

[5] Although the MUSING ontology uses English notations for the classes, it offers for each class German labels.

```
Person                (CLASS)
 |
 |
Customer (Kunde)   (SUBCLASS)
     |
     |
  Reseller (Wiederverkäufer) (SUBCLASS)
Vendor  (Verkäufer)
     |
     |
  Competitor (Konkurrent) (SUBCLASS)
  Supplier   (Lieferant)   (SUBCLASS)
Partner (Partner)   (SUBCLASS)
```

Figure 8.1: The class *Person* in MUSING.

Vorgesetzter-hierarchisch_ausgerichteter_Mensch-Person. But the path is a different one for *Mitarbeiter*: *Mitarbeiter-Berufstätiger-Person*.

The MUSING class *Group* contains a single subclass, *Organization* which has seven subclasses, each of them having further subclasses. Figure 8.2 depicts the two levels under the superclass *Group*.

```
Group                 (CLASS)
 |
 |
Organization     (SUBCLASS)
       |
       |
  Commercial Organization (SUBCLASS)
  Educational Organization (SUBCLASS)
  Government Organization (SUBCLASS)
  International Organization (SUBCLASS)
  Religious Organization (SUBCLASS)
  Research Organization   (SUBCLASS)
  Sport Organization (SUBCLASS)
```

Figure 8.2: The class *Group* in MUSING.

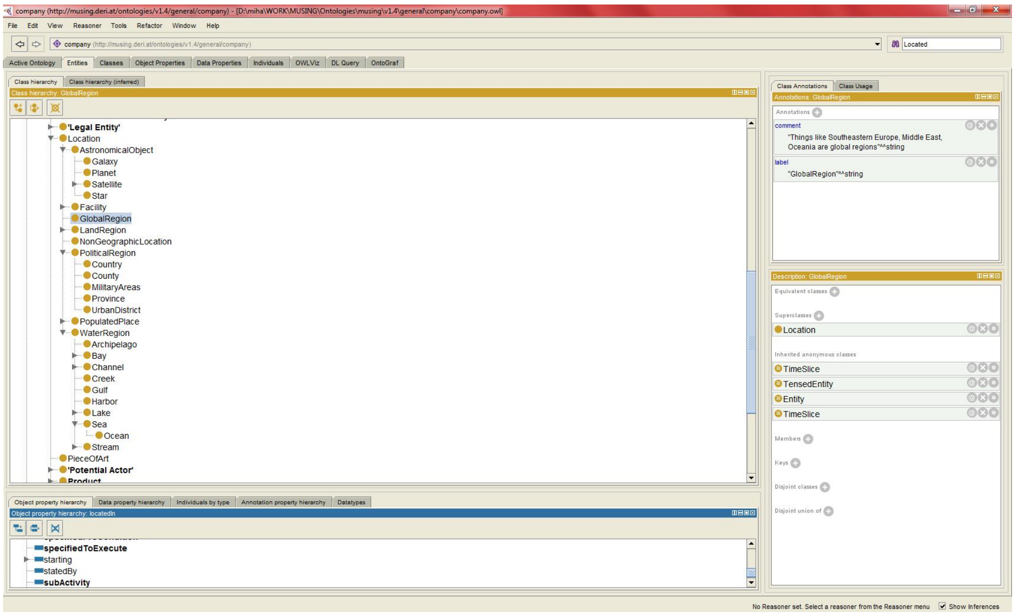
Concerning the subclasses for the different types of organizations, our results coincide with those in MUSING. We also extracted classes like *Firma* (*company*), *Bank* (*bank*), *Regierung* (*government*), *Unternehmen* (*company*) which appear in our ontology as subclasses of the superclass *Organization.* Although we did not use the semantic path in GermaNet for building our ontology[6], we can use it at this point to compare our results with those in MUSING.

Although the MUSING company ontology is really large, due to its goal of internationalization, it cannot cover the same aspects, as the one covered by our method. For example, the MUSING company ontology contains the class *Event*, which we also extract with our method. The MUSING class *Event* has as subclasses *Accident*, *Activity*, *ArtPerformance*, *Meeting*, *Military*, *Project*, *SportEvent*, but no subclass *EconomicEvent.* Although the class *activity* contains the subclasses *Manage*, *Promotion* and *Planning*, we suggest here a class *EconomicEvent*, as a subclass of *Event.* This class can than contain subclasses such as *Rezession* (*recession*), *Expansion* (*expansion*), *Produktion* (*production*) and *Ankurbelung* (*boost*). Besides the extension of the class *Event*, we can extend the MUSING company by the object property `hasEvent`.

The class *Location* is used by us for determining the geographical location, without taking into consideration that a galaxy also denotes a location. In MUSING, the class *Location* is classified a very detailed way, including the information that an ocean is a sea and the sea is a water region. For our corpus and our purpose the subclassification of the class *PoliticalRegion* would have been enough. Figure 8.3 shows the MUSING class *Location* in Protégé. Connected to the class *Location*, we introduce the object property `hasLocation`. In MUSING this relation is named `locatedIn`, but means in fact the same things as `hasLocation`.

The MUSING company ontology contains also a class *Feature*, which is in fact a synonym for the class *Attribute.* This way, we are able to extend the existing

[6] We did not consider this path because this would lead to an overgeneration of superclasses for the extracted nouns, without really introducing new knowledge into the ontology.



company (http://musing.deri.at/ontologies/v1.4/general/company) - [D:\miha\WORK\MUSING\Ontologies\musing\v1.4\general\company\company.owl]

company (http://musing.deri.at/ontologies/v1.4/general/company)

Located

Class hierarchy: GlobalRegion

```
'Legal Entity'
Location
    AstronomicalObject
        Galaxy
        Planet
        Satellite
        Star
    Facility
    GlobalRegion
    LandRegion
    NonGeographicLocation
    PoliticalRegion
        Country
        County
        MilitaryAreas
        Province
        UrbanDistrict
    PopulatedPlace
    WaterRegion
        Archipelago
        Bay
        Channel
        Creek
        Gulf
        Harbor
        Lake
        Sea
            Ocean
        Stream
PieceOfArt
'Potential Actor'
Product
```

Object property hierarchy: locatedIn

```
specifiedToExecute
starting
statedBy
subActivity
```

Annotations: GlobalRegion

Annotations
comment
"Things like Southeastern Europe, Middle East, Oceania are global regions"^^string
label
"GlobalRegion"^^string

Description: GlobalRegion

Superclasses
Location

Inherited anonymous classes
TimeSlice
TensedEntity
Entity
TimeSlice

No Reasoner set. Select a reasoner from the Reasoner menu

Figure 8.3: The class *Location* in MUSING.

MUSING class *Feature* with attributes such as *Geduld* (*patience*), *Kraft* (*power*), *Anspruch* (*demand*), *Haltung* (*attitude*), *Design* (*design*), *Disposition* (*disposition*). As this list shows, these are different types of attributes which we can further specify by using GermaNet's semantic path. We can distinguish between attributes of a person such as *Geduld*, *Kraft* and attributes of an object such as *Design*, *Volumen* or *Form*. Strongly connected to this class is the `hasAttribute` object property which extends the already existing set of object properties in MUSING.

The classes *Abstract* and *Object* do not offer information for extending the MUSING ontology. The class *Object* is in MUSING a superclass of the class *Location* and the class *Abstract* a superclass of the class *Feature*. We consider *Location*, *Object*, *Feature* and *Abstract* direct subclasses of *Thing*. Figure 8.1 lists the superclasses which are extended by our approach. The plus sign means that our method enriches the MUSING ontology. Minus means that the specific class exists in the MUSING ontology.

Class name	
Group	+
Person	+
Location	-
Event	+
Feature	+
Object	
Abstract	

Table 8.1: Superclasses which enrich the MUSING classes.

For the ontology classes we can conclude, that our method extends the MUSING ontology by subclasses The object property `hasAffiliation` is represented in MUSING by two more specific object properties, `hasMember` and `hasNationality`. This subclassification of the `hasAffiliation` relation we can achieve by using GermaNet's hyperonym path (for deducing the `hasMember` object property) and a more detailed semantic classification of adjectives (for deducing the `hasNationality` object property). Although not for all nouns applicable, we can make the semantic difference between nouns like *Manager* (*manager*), *Kunde* (*customer*) and *Mitarbeiter* (*collaborator*). This way we are able to classify the object property `hasMember` into `hasLeader`, `hasEmployee` and `hasAgent`. The `hasNationality` object property is strongly connected with the adjectives, which need to be further specified semantically. In order to make the distinction between *deutsch* (*German*), *heidelberger* (*from Heidelberg*) and *kirchlich* (*churchy*), which are all classified as affiliation adjectives, a more fine-grained classification is needed. We have to stress here that we use GermaNet to propose a more fine-grained classification of the relations and classes extracted by us. Figure 8.2 shows relations extracted from phrases which enrich the MUSING ontology.

The remaining two object properties `hasDimension` and `disposesOver` do not occur in the MUSING company ontology and can therefore extend the existing object properties in MUSING. The three datatype properties `hasMoneyValue`, `hasTimeUnitValue` and `hasQuantitativeValue` are available in the MUSING

Relation name	
hasDimension	+
hasAttribute	+
hasAffiliation	-
disposesOver	+
hasEvent	+
hasLocation	-
hasPosition	-
hasNumberValue	+
hasTimeUnitValue	+
hasQuantitativeValue	+
isOppositeTo	+
partOf	-

Table 8.2: Relations which enrich the MUSING relations.

ontology as the `hasValue` object property. Another aspect to be discussed here is concerned with the relations introduced by the antonyms and meronyms in GermaNet. The `isOppositeTo` property does not exist in the ontology and can extend the existing set of object properties in MUSING, whereas the `partOf` object property is available in the MUSING ontology.

Besides these relation, we extracted relations from premodification phenomena and grammatical functions. From the premodification phenomena we enrich the MUSING ontology by 23 relations. The top 10 most frequent verbs introduce 22 new relations into the MUSING ontology.

The comparison with the MUSING ontology can be carried on with the instantiations. In MUSING, only a few classes are instantiated and most of these instantiations are NACE codes. With our method we are able to instantiate persons, organizations and locations and the currency class. The classes *Person*, *Organization* and *Location* do not contain any instance, so we are able to introduce our instantiations into the ontology. Concerning the locations, we also noticed that our classes coincide with those in MUSING, respectively, the classes *City*, *Country*, *Province* are subclasses of the generic class *Location*. Our object property `hasLocation` corresponds to the MUSING object property `locatedIn`. In contrast, the class *Currency* is instantiated with the instances *Euro* and *US_Dollar*. With our method we are able to extend the instantiations with the Russian ruble, the Swedish krone and the German Mark, which are also instantiations of the class *Currency*.

8.3 Evaluation Against a Manually Annotated Test Suite

The evaluation of the method presented here was performed on a manually annotated test suite. The test suite consists of 200 randomly selected sentences (out of over 11000) which were annotated by a student of business informatics. In this way we ensure that the annotator is familiar with the financial language and the ontological constructs which may appear in an ontology. For building the test suite we used a similar method to the one used for the semantic annotation of the CLEF corpus (Roberts et al., 2007). We deliberately do not use the formulation "gold standard" here, because our methodology for building the manually annotated test suite differs from the NLP standards for building a gold standard. According to Boisen et al. (2002) when building a gold standard the annotators use annotation guidelines and the annotation is performed by more than one annotator. Furthermore, an annotation is considered a good annotation only if they pass a threshold. The annotation differences between annotators are resolved by a third experienced annotator. Our manual annotator annotated also according to annotation guidelines, but he was the only annotator. In order to ensure the quality of the manual annotation after the first 20 manually annotated sentences we carried out a refinement session. We checked the annotation and, depending on its quality, we instructed the annotator to correct the annotation.

8.3.1 Guidelines for the Manual Annotation

The role of annotation guidelines is to ensure the consistency of the annotation. When building a manually annotated test suite, as we did, it is important that the same phenomena are annotated by using the same standard, especially if more than one annotator is used to construct the test suite. The sentences to be annotated were presented to the annotator one after another, in a column, in an Excel table. He was instructed to annotate all semantic relations between concepts in the sentence and the instantiation of these concepts. The annotator wrote the extracted semantic relations in the same row as the input sentence. Each new relation discovered in the sentence was written in a new column.

In the following we will explain, based on examples, what we expected from the annotator when we say that we want to annotate the semantic relation between concepts. For example, from the compound *Konzernchef* (*chief of the corporation*) we expect that the annotator detects a relation between the two entities *Konzern* (*corporation*) and *Chef* (*chief*), such as `subClassOf(Konzernchef, Chef)` and `Konzern hasPosition Chef`. From linguistic constructions like *Experten der Bank* (*bank experts*), *Aktie der Bank* (*bank share*) or *Wohnung im Westen* (*apartment in the west*) the annotators should also detect a relation between the pairs *Experten-Bank*, *Aktie-Bank* and *Wohnung-Westen.* Since the relation names depend on the GermaNet classification of the nouns, we instructed the annotator to mark the relation with a generic name `hasProperty`. For us it is important to see whether we covered these relations and not whether the naming of the relations was similar. The relations above are not exhaustive, they are only examples of how the relation extraction can function.

(21) Der Konzern verdiente Millionen.

The corporation earned millions.

Another type of semantic relation concerns the verbs. For example, one semantic relation which can be extracted from sentence 21 concerns the entities *Konzern* (*corporation*) and *Million* (*million*). In this cases, the semantic relation is named as the verb, `earns(Konzern, Million)`.

As mentioned above, we are also interested in the instantiation of entities. By instantiation we mean the identification of the names of these concepts in the corpus. For example, *George W. Bush* is an instantiation of a person, whereas *Berlin* is the instantiation of a location. We have to notice here, that entities like attribute, state, feeling, motive or process do not have instantiations. Only entities which exist in the real world can have instantiations.

The further specification of these entities should be fulfilled from structures like *deutsche Firma* (*German company*), *größte deutsche Firma*, *deutsche und französische Firmen*, *deutsche, englische französische Firmen*, *sehr große Firma.*

From the structure *deutsche, englische französische Firmen*, the annotator will write each of the relations `hasProperty(Firma, Deutsche)`, `hasProperty(Firma, Französische)` and `hasProperty(Firma, Französische)` in a different column. The format of the extracted relations will correspond to the triple relationName(relationSubject, relationObject), such as `hasProperty(Firma, Deutsche)`.

8.3.2 The Results

The 200 sentences selected for manual annotation were also processed with our method and the corresponding tools. The quantitative evaluation was performed in two stages, and after each stage we measured the performance of our method. We compared the results of our method with the manual annotation by counting precision and recall scores. For a document containing t semantic relations, from which m were extracted correct, n incorrect and some not at all, the recall is m/t and precision is m/(m+n). The best score is 1, the worst value is 0. We did not use the core precision and recall numbers. Instead we use the F-measure which combines precision and recall. The general formula for F-measure is depicted in below (see Figure 8.4). Depending on the value assigned to β we can determine whether precision or recall is weighted more. The most common usage for F-measure is F_1, when β is assigned to 1 and F_1 corresponds to the harmonic mean of precision and recall.

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

Figure 8.4: General formula for F-measure.

When β is assigned to 2 (F_2) recall is weighted twice as much as precision and $F_{0.5}$ weights precision twice as much as recall. From the results in Table 8.3 we notice that we have the best results when it comes to extracting the `subClassOf` relation. The good results are not a surprise, since the `subClassOf` relation is extracted mainly from compounds. Because our compound filtering process relies both on PoS and a noun lexicon, the compounds filtered out are indeed the ones from which we can extract the `subClassOf` relation. In this way we ensure that only real noun compounds from which correct `subClassOf` relations can be extracted. It seems that the 200 manually annotated sentences contain only determinative compounds. The `subClassOf` relation is extracted not only from compounds but is introduced into the ontology from GermaNet. In this case the left-hand side argument of the `subClassOf` relation differs from the one chosen by the manual annotator. This we will not weight here as negative, since we found it totally normal that a human being produces semantic annotations in a different way to GermaNet. For example, we introduce the noun *Wohnung* (*apartment*), based on GermaNet, into the ontology as a subclass of the more general class *Object*. The manual annotator allocated *Wohnung* to the superclass *Immobilie*. Both assignments are correct, but we notice that the manual annotator has chosen a more specific superclass than the one we use.

Phenomenon	Precision	Recall	F_1	F_2	$F_{0.5}$
SubClassOf	1	1	1	1	1
Modification phenomena	1	0,52	0,68	0,84	0,64
Relations from phrases	1	0,23	0,37	0,60	0,33
Grammatical functions	0,5	0,30	0,38	0,38	0,38
Instantiation	1	0,82	0,89	0,95	0,87

Table 8.3: Precision and recall scores for our approach in the first evaluation round.

The results from the modification phenomena show that we have a very good precision. This means that we either find a true relation or we do not find it at all. This corresponds to the methodology applied: if a modifier is in our modifier lexicon it produces a true relation, if not it does not produce anything and these we can read from the recall score. The adjectives *westdeutsch* or *größter* are in the first evaluation stage not written in our lexicon and are consequently not covered by our rules. For the relations extracted from phrases we achieve the lowest scores concerning the recall. This low score is due to three factors: there is no rule for extracting a relation, the implemented rule does not work properly and the rule exists but it does not fire because of lack of semantic information. The first two factors we can influence by writing new rules or improving the implementation of the existing rules. In fact the GermaNet lookup fails because certain nouns do not have a stem and the GermaNet lookup is based on stems. For example, the noun *Beschäftigter* (*employee*) has no stem in the input file for our rules. The missing stem feature makes it than impossible to find in GermaNet its semantic class and to apply our rules.

The scores for ontology extraction from grammatical functions show one characteristic common to all other phenomena: the relation is either not found or if it is found than, it is correct. The precision and recall (and consequently the F-measure) scores are not necessary influenced by our rules, but by the assignment of grammatical functions by the the SCHUG parser (Declerck, 2002). By incorrect grammatical function assignment we mean in fact the ambiguous assignment. When applying our rules, we have to decide automatically for one of the variants and it sometimes happens, that we do not choose the correct variant. Because we cannot influence the ambiguity of the grammatical function assignment, in the second evaluation round we manually corrected the ambiguities provided by the SCHUG parser.

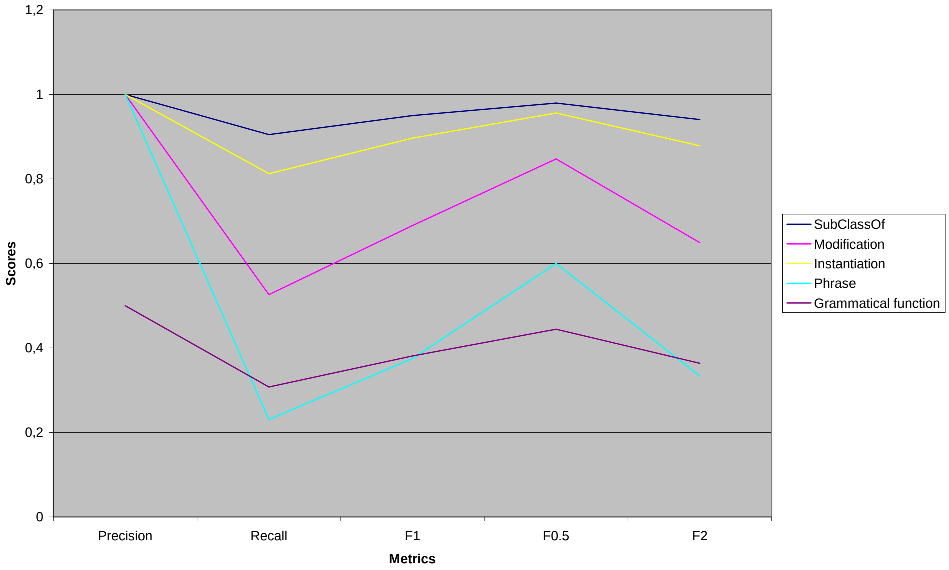


Figure 8.5: Graphical representation of the measured scores in the first evaluation round.

The scores for instantiations show that instantiation works fine. Only a small set of instantiations cannot be found by us and this is motivated by the unexpected input format for the implemented extraction rule.

In a second evaluation round we concentrated on relations from phrases and modification phenomena and were able to improve their shortcomings from the first

Phenomenon	Precision	Recall	F_1	F_2	$F_{0.5}$
SubClassOf	1	1	1	1	1
Modification phenomena	1	1	1	1	1
Relations from phrases	1	0,61	0,76	0,88	0,72
Grammatical functions	1	0,61	0,80	0,90	0,70
Instantiation	1	0,82	0,89	0,95	0,87

Table 8.4: Precision and recall scores for our approach in the second evaluation round.

evaluation round. Therefore we improved the scripts implementing the rules for ontology extraction from phrases and enlarged our lexicons for ontology extraction from modification phenomena. We decided to not write new extraction rules, because they might interfere with the existing ones. So the scores for the relations extracted from phrases are due to the missing rules and the missing stem for nouns. Figure 8.6 depicts the overlap between the scores for the different phenomena used for ontology extraction.

The manual disambiguation of the grammatical function assignment provided a considerable improvement on the measured scores. Also part of the evaluation is Appendix D, a simple quantitative evaluation based on frequencies.

The comparison of our evaluation results with other studies is a difficult task, since we need similar data and phenomena to be compared. Ciaramita et al. (2008) evaluated their rule-based ontology learning approach from the biomedical domain by measuring precision. Their method differs from ours in three aspects: relations are extracted from a different corpus (the GENIA corpus Tomoko Ohta (2002)), from a different domain (molecular biology) and, the extracted relations are based only on dependency structures. They propose a method for ontology learning based on dependency structures and evaluate the extraction potential of the found patterns from verb-argument structures. From 287 patterns, 91 were impossible to evaluate and excluded from evaluation. From the remaining 196, 150 patterns could be evaluated as correct (76,5%). A direct comparison of the

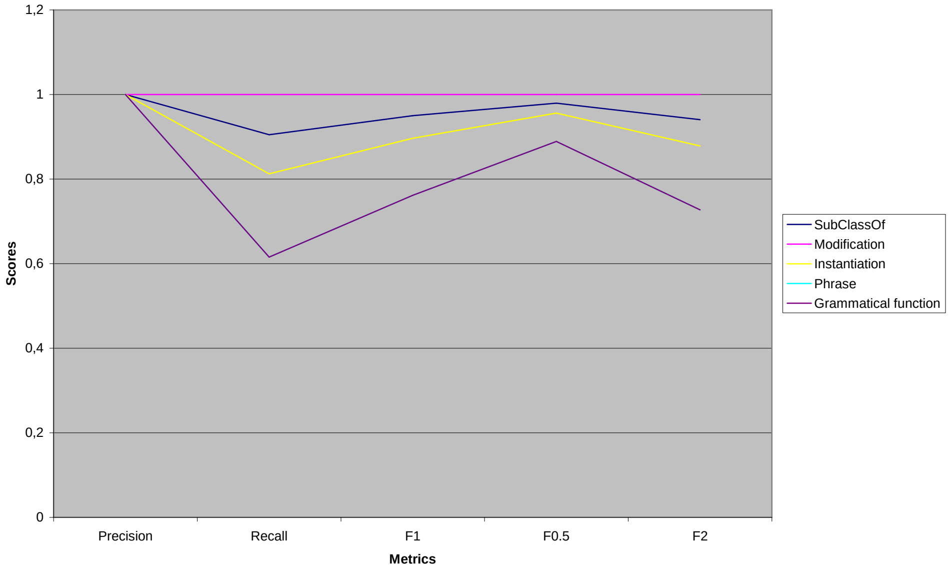


Figure 8.6: Graphical representation of the measured scores in the second evaluation round.

methods is not feasible, but their patterns are comparable with our rules for ontology extraction from grammatical functions. These rules we can also compare with Cimiano et al. (2005)'s approach. Cimiano et al. (2005) propose a machine learning approach for ontology learning from dependency structures. They evaluated their machine learning on two domains, the finance and the tourism domain, achieving F-measure scores of 40,5%, respectively 33,1%. Our evaluation scores are higher than the ones achieved by Ciaramita et al. (2008) and Cimiano et al. (2005), but objective and correct comparison can be performed only by testing the three approaches on the same corpus by using the same linguistic analysis tools.

8.4 Conclusion

This chapter is about evaluating our approach. In Section 8.3.1, we analyze the compatibility of our results with the MUSING ontology. In Section 8.2, we compare our results with a manually built test suite. Concerning the first aspect, we have shown that the MUSING company ontology can be extended by our approach. The extension of the MUSING ontology is possible for certain classes (such as the *Person*) and object properties (such as `hasAttribute`).

For the evaluation based on the F-measure metric we can conclude the following: either we do not find a specific relation (visible in the recall) or we find it and then it is correct (visible in the precision). For the `subClassOf` relation the very good scores are due to our method, whereas for the relations extracted from modification phenomena the results depend on the semantic resources. For the relations extracted from phrases, we have to notice here that these good results are due to the fact that we do not evaluate relations names, but only relations. This means that for us it is important to know whether we discovered a relation or not, and not how this relation is named. For the ontological knowledge extracted from grammatical functions, we need to say here that the results are strongly connected to the capacities of the parser, more precisely its capability to disambiguate. As a final remark we have to notice that the linguistic-based developed rules for ontology extraction cover 80% of the ontological knowledge annotated by our manual annotator.

Chapter 9

Outlook

In this chapter we first summarize the work described in this thesis (Section 9.1). Then we outline some possibilities for the extension and reusability of the work presented. Section 9.1 summarizes the preceding chapters in this thesis. Section 9.2 describes the linguistic phenomena which have not been considered for the process of ontology extraction and Section 9.3 deals with the integration of the work presented here in the broader area of ontology extraction.

9.1 Summary

We have described an incremental multi-layer rule-based methodology for the extraction of ontology schema components from German financial newspaper text. We concentrated on describing both the process of rule generation for the extraction of ontology schema components and the application of the developed rules.

Chapter 2 provided definitions and descriptions of the different linguistic and semantic analysis steps. In Chapter 3, we presented the state of the art with reference to the work presented in this thesis.

In Chapter 4, we presented the methodology applied for accomplishing our approach.

Chapter 5 gave a detailed description of the designed rules and their application for the extraction of ontological knowledge.

Chapter 6 dealt with the formalization of the ontological knowledge extracted by the method presented in this thesis.

Chapter 7 concentrated on demonstrating the expandability of the approach presented in this thesis.

Chapter 8 compared our results with the MUSING ontology and presented the results of the numeric evaluation.

9.2 Linguistic Phenomena not Covered Yet

There are several linguistic and semantic phenomena which can be annotated and used for different purposes in Computational Linguistics. The phenomena which we consider in this thesis (compounding, nominalization, premodification, postmodification, phrase-structure, as well as lexical semantics) are very important for our work but are not exhaustive. From a purely linguistic point of view we do not take into consideration the peculiarities of relative clauses. We also do not handle with the semantic and linguistic properties of the negation particle or coreference. These phenomena are not treated here because of a more pragmatic and practical reason: the linguistic tools we have at hand in this thesis do not annotate these kind of phenomena. To integrate these phenomena into the approach presented here remains an issue for future work.

9.3 Integration as Future Work

In this section we concentrate on sketching how our work can be integrated in a broader research context. With broader context we mean on the one hand existing upper level ontologies[1], such as SUMO, and on the other hand research in the area of ontology learning.

9.3.1 Integration into Upper Level Ontologies

The Suggested Upper Merged Ontology (SUMO)[2] is an ontology consisting of several domain ontologies. SUMO is in fact the largest free available ontology. Another important characteristic of SUMO is the fact that it has been mapped to the whole lexicon of WordNet. From this perspective SUMO is the ontology which fits our approach when it comes to integrate our work into a broader ontology. It is true, that there is no direct mapping between GermaNet and SUMO. This situation can be solved by first mapping from GermaNet to WordNet and then to SUMO. The direct mapping between GermaNet and WordNet is possible since both have the same general structure concerning the semantic tree. So, for example, the more general concepts such as `group`, `person`, `attribute` are integrated both in GermaNet as well as in WordNet. And since our ontological knowledge is always connected to the more general nodes in the semantic network, we can easily transpose our results into WordNet and from there to SUMO. For example, each subclass of the generic classes in Table A.1 in Appendix A.2 can be mapped into SUMO. The integration of relations is a more complicated process, since we need to a very good relation alignment in order to map only new relations into SUMO.

[1] Upper level ontologies are general ontologies.

[2] http://www.ontologyportal.org/

9.3.2 Integration into NeOn

Here, we first give a detailed presentation of the ontology patterns in the NeOn project. In the next step we show how the ontology extraction rules presented in this thesis can extend the existing ontology patterns inventory.

NeOn Ontology Patterns

In this section we summarize sections of a NeOn deliverable dedicated to Ontology Design Patterns (ODP) (Gangemi et al., 2008). The aim of NeOn is to create a methodology for generating semantic applications. These applications rely on a network of contextualized ontologies. As reusable solutions for collaborative design of networked ontologies the NeOn project defined Ontology Design Patterns (ODPs). NeOn distinguishes six different types of Ontology Design Patterns (ODPs): Structural ODPs, Correspondence ODPs, Content ODPs, Reasoning ODPs, Presentation ODPs, and Lexico-Syntactic ODPs. Figure 9.1 depicts the NeOn ODP structure. Each family addresses different kinds of problems and can be represented with different levels of formality.

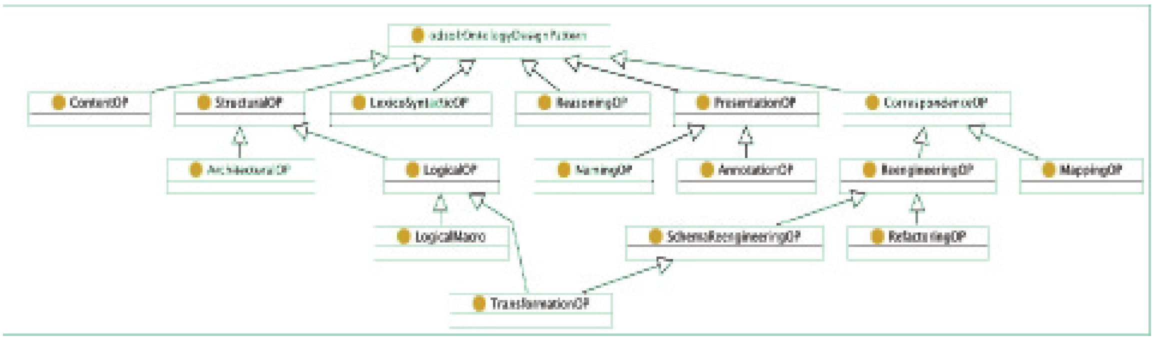


Figure 9.1: Ontology Design Patterns in NeOn

Structural ODPs include Logical ODPs and Architectural ODPs. A logical design pattern is a formal expression whose only parts are expressions from the logical vocabulary of OWL DL that solve a problem of expressivity. The current inventory of NeOn Ontology Modelling Components considered as Logical Patterns includes, as a sample, the following ones: primitive class, defined class, `subClassOf` relation between classes, multiple inheritance between classes (using `subClassOf`), equivalence relation between classes, `objectProperty`, `subPropertyOf` relation between object properties, datatype property, existential restriction, universal restriction, union of classes, individual, disjoint classes, covering axiom, defining n-ary relations, and representing specified values in OWL.

Architectural ODPs are defined in terms of composition of Logical ODPs. Their aim is to constrain 'how the ontology should look like'. They are used in the design of the ontology as a whole, by providing the composition of Logical ODPs that have to be exclusively employed when designing an ontology. The following NeOn Ontology Modeling Components are considered Architectural Patterns: tree structure, binary tree structure, graph structure, taxonomy structure, lightweight ontology and modular architecture.

Content ODPs encode conceptual, rather than logical design patterns. In other words, while Logical ODPs solve design problems independently of a particular conceptualization. Content ODPs propose patterns for solving design problems for the domain classes and properties that populate an ontology, therefore addressing content problems. The current inventory of NeOn Ontology Modeling Components considered as Content Patterns includes as a sample the following patterns: participation pattern, description-situation pattern, role-task pattern, plan-execution pattern, and simple part-whole relations pattern.

Lexico-Syntactic ODPs can be defined as linguistic structures or schemes that consist of certain types of words following a specific order, and permit one to generalize and extract some conclusions about the meaning they express. For example, in one of the patterns that corresponds to the `subClassOf` relation, *NP<subclass> be NP<superclass>*, a Noun Phrase (NP)[3] should appear before the verb - represented by its basic form or lemma, *be* in this example - and the verb should in turn be followed by another Noun Phrase. In this way, sentences in English like *Dolphins are warm blooded mammals* could be asserted by expressing a hyponymy-hyperonymy relation between the two Noun Phrases.

Correspondence ODPs include Reengineering ODPs and Mapping ODPs. Reengineering Ontology Design Patterns (Reengineering ODPs) are transformation rules applied in order to create a new ontology (target model) starting from elements of a source model. The target model is an ontology, while the source model can be either an ontology, or a non-ontological resource e.g., a thesaurus concept, a data model pattern, a UML model, a linguistic structure, etc.

Mapping ODPs refer to the possible semantic relations between mappable elements. There are three basic semantic relations that are used for mapping assertions: equivalence, containment, and overlap. They can be supplemented by their negative counterparts i.e., not equivalent, not contained, and not overlap or disjoint, respectively. Mapping ODPs provide designers with solutions to relate two ontologies without changing the logical types (e.g. owl:Class) of the ontology elements involved.

Reasoning ODPs are applications of Logical ODPs oriented to obtain certain reasoning results, based on the behaviour implemented in a reasoning engine. Examples of Reasoning ODPs include: classification, subsumption, inheritance, materialization, de-anonymizing, etc.

Presentation ODPs deal with the usability and readability of ontologies from a user perspective. They are meant as good practices that support the reuse of patterns by facilitating their evaluation and selection. The ontological knowledge extracted in NeOn is based on ontology authoring extracted at the sentence level.

[3] A NP is a phrase whose head is a noun or a pronoun, optionally accompanied by a set of modifiers

Our Relations in NeOn

In the following, we will describe to what extent the ontology design patterns in NeOn can be extended with our method. The first two rules for the extraction of ontology schema components are the generic `objectProperty` and `subClassOf` relation in Section 4.1.1. These two ontological relations can be easily integrated into NeOn, since both the introduced classes and the generic relations correspond to the class of Logical Patterns (OWL elements). NeOn does not cover phenomena covered by our extraction rules from paraphrases. A reason for this is the fact, that NeOn does ontology authoring only on predicate-argument structures (either string-based or syntactic-based), without dealing with phenomena such as pre- and postmodification. Also, NeOn does not use any semantic information. Therefore, at the phrase level, there are several relations which are not covered in NeOn. For this case, we propose the extension of the Lexico-Syntactic Pattern set with new relations. Since we also perform ontology population, the results from ontology population can also be classified into the Lexico-Syntactic Pattern.

For the relations extracted at the sentential level (Section 5.3.1) we found that the modification rules and the apposition rule are not covered by NeOn, since they imply besides phrase structure information also lexical semantics. As already proposed above, the extension of the Lexico-Syntactic Pattern with the new relations will solve this kind of problem. The same applies for the relations such as `isa`, `cause`, `earn` which are not listed in the NeOn catalogue of Ontology Patterns.

A result from using GermaNet is that we can also introduce synonyms, hyponyms and meronyms into the ontology. With our method, hyponyms are covered by the `subClassOf` relation, meronyms by the `partOf` relation and the synonyms are introduced as labels. In NeOn, the hyponyms can be represented by the `subClassOf` ontology pattern and the meronyms can be represented by the `genericRelation` relation. The synonyms can be formalized in NeOn with the `equivalence between classes` rule, which is listed under the Logical Patterns. A reason why the relations extracted by us are covered only at the OWL generic level in NeOn is due to the fact that the NeOn project does ontology authoring (users are asked to formulate sentences describing specific phenomena), whereas we try to cover aspects found in the corpus. Nevertheless, the comparison has shown that our method can help to increase the number of ontology patterns in the NeOn repository by at least the seven patterns from paraphrases and the eight patterns from modification phenomena. By introducing the NeOn[4] project, we described how the existing ontology patterns can be extended and formalized (or not) with OWL DL. Concerning the interconnectivity between the method presented in this thesis and the NeOn project we state that the extension of the existing NeOn ontology patterns is possible because NeOn extracts the ontological knowledge by ontology authoring from sentential level. Since the majority of our ontology extraction rules handle compounds, phrases and modification phenomena, it is self-evident that our ontology extraction rules extend the ones in the NeOn project.

[4] http://www.neon-project.org

Appendices

Appendix A

Tools and Resources

A.1 Used Tools

A.1.1 SProUT

SProUT (Shallow Processing with Unification and Typed Feature Structures) is a platform for the development of multilingual shallow text processing and information extraction systems which incorporates in it a morphological analyzer and a PoS tagger. SProUT was developed at the Language Technology Lab of the DFKI (Deutsches Forschungszentrum für Künstliche Intelligenz) Saarbrücken.

A.1.2 SCHUG

SCHUG (Shallow and Chunk Based Unification Grammar) is a parser, implemented in Perl for German and English. The model of cascaded chunk processing adopted for SCHUG presupposes a sequence of levels. This means that the linguistic structures on one level are built on linguistic information from the previous level. SCHUG is applying higher-level linguistic knowledge to the morphologically annotated input delivered by SProUT and generates dependency structures between the various elements and syntactic constituents of the analyzed sentence. One kind of dependency structure is phrase internal and describes for example the modification relation between adjectives and the main noun of a nominal phrase. Another kind of dependency structure information provided by SCHUG is the one existing between a nominal phrase (NP) and the predicate of the sentence, whereas the NP can be for example the subject or the direct object of the predicate. The latter type of dependency structure is known as the grammatical function (GF) of linguistic constituents. SCHUG was developed at the Language Technology Lab of the DFKI Saarbrücken

A.1.3 Java GermaNet API

The Java GermaNet API (Gurevych and Niederlich, 2005) is an application interface for Java, which allows easy access to all information available in GermaNet. The API provides a set of software functions for parsing and retrieving information from GermaNet, such as synonyms and antonyms.

A.1.4 Protégé

Protégé[1] is a free, open source ontology editor and knowledge-base framework. The Protégé platform supports two main ways of modeling ontologies via the Protégé-Frames and Protégé-OWL editors. Protégé ontologies can be exported into a variety of formats including RDF(S), OWL, and XML Schema. Protégé is based on Java, is extensible, and provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development.

[1] http://protege.stanford.edu/

A.1.5 Used Database

In order to perform an easier query through the corpus, we modeled and built a database for the entire corpus. The database encodes the annotated corpus (PoS, morphology, NE, grammatical functions).

A.1.6 Implemented Java Scripts

Text-Based Processing

For the text-based processing we implemented a Java script which performs the following three steps:

- Extraction of all potential concepts.
- Extraction of all compounds in which the concepts from the previous point are part of.
- Extraction of all paraphrases for the compounds in the previous point.

Shallow Linguistic Processing Processing

For the text-based processing we implemented a Java script which performs the following three steps:

- Extraction of classes and relations from the two nouns of a genitive or prepositional phrases.
- Extraction of classes and relations from the modified nouns.
- Instantiation of the already extracted classes.

Deep Linguistic Processing Processing

For the text-based processing we implemented a Java script which performs the extraction of ontological knowledge from predicate-argument structures.

A.2 Resources

A.2.1 The Corpus

For the research presented in this we use a corpus of German financial newspaper text, more precisely the 1992 edition of the German newspaper "Wirtschaftswoche". The corpus comprises 200107 words, 11583 sentences and 121331 phrases.

A.2.2 GermaNet

GermaNet (Kunze and Lemnitzer, 2002) is a lexical-semantic net that relates German nouns, verbs, and adjectives semantically by grouping lexical units that express the same concept into synsets and by defining semantic relations between these synsets. GermaNet has much in common with the English WordNet (Fellbaum, 1998) and might be viewed as an on-line thesaurus or a light-weigt ontology.

GermaNet classifies all nouns into semantic fields (tops). Table A.1 list all semantic fields for nouns.

GermaNet's Semantic Fields for Nouns
artifact
attribute
possession
motion
relation
event
shape
feeling
group
body
cognition
communication
quantity
person
motive
food
object
phenomenon
location
plant
substance
animal
time

Table A.1: The list of semantic fields in GermaNet.

A.2.3 Semantic Classification of Adjectives

Number	Name of Class	Examples
1	Adjectives of Possession	dickköpfig, gutwillig, weitmaschig, lederartig
2	Adjectives of Tendency	naschhaft, ergiebig, anschmiegsam, spendabel
3	Adjectives of Possibility	erkennbar, denkbar, strahlungsfähig, erholsam
4	Adjectives of Necessity	vergänglich, anstrebenswert
5	Stative Adjectives	freudig, durstig, kundig, lustig
6	Dimensional Adjectives	lang, stark, schwer, breit, kurz
7	Adjectives of Privaticity	ärmellos, inhaltsleer, fehlerfrei
8	Objective plus Temporal Combination	verkehrsschwache Zeiten
9	Objective plus Locative Combination	einsamer Ort
10	Material Adjective	eisern, silbern, golden, gläsern
11	Quantitative Adjective	sämtlich, übrig, ganz, drei
12	Spatial Adjective	französisch, äußer, dortig, hinter
13	Temporal Adjective	baldig, heutig, morgig, sommerlich
14	Adjective of Affiliation	heidnisch, kirchlich, väterlich
15	Instrumental Adjective	nuklear, maschinell, mechanisch, brieflich
16	Adjective of Counterpart	gewohnheitsmäßig, naturgemäß, ordentlich, redlich
17	Actional Adjective	fachmännisch, polizeilich, fahrerflüchtig
18	Reference Adjective	biologisch, chronisch, klinisch, sportlich
19	Causative Adjective	monsunal, bakteriell, tuberkulos, nervös
20	Equivalence Adjective	katastrophal, trottelig, eklatant
21	Gradable Adjective	dick, erbärmlich, gewaltig, entsetzlich
22	Modal Adjective	bewußt, ambulant
23	Occurrence Adjective	ansichtig, fällig
24	Comparison Adjective	grippal, ledern, nonnenhaft

Table A.2: Semantic classification of adjectives.

Number	Name of Class	Introduced Relation
1	Adjectives of Possession	hasPossession
2	Adjectives of Tendency	hasTendency
3	Adjectives of Possibility	hasPosibility
4	Adjectives of Necessity	hasNecessity
5	Stative Adjectives	hasState
6	Dimensional Adjectives	hasDimension
7	Adjectives of Privaticity	hasPrivaticity
8	Objective plus Temporal Combination	-
9	Objective plus Locative Combination	-
10	Material Adjective	isMadeOf
11	Quantitative Adjective	hasQuantity
12	Spatial Adjective	hasLocation
13	Temporal Adjective	hasTime
14	Adjective of Affiliation	hasAffiliation
15	Instrumental Adjective	hasIntrument
16	Adjective of Counterpart	hasCounterpart
17	Actional Adjective	actsLike
18	Reference Adjective	hasReference
19	Causative Adjective	hasCause
20	Equivalence Adjective	hasEquivalence
21	Gradable Adjective	hasGrade
22	Modal Adjective	hasMode
23	Occurrence Adjective	hasFrequency
24	Comparison Adjective	hasComparison

Table A.3: Relations introduced by the adjectives.

A.2.4 Semantic Classification of Adverbs

Number	Name of Class	Examples
1	Possibility	vermutlich, eventuell, vielleicht
2	Attitude	sicher, unbedingt
3	Temporal	dann, heute, gestern, bislang
4	Aspect	weiterhin, derzeit, niemals, immer
5	Frequency	oftmals, oft, meistens, einmal
6	Manner	medikamentoes, rechts, rund, gerade
7	Focus	allerdings, soweit, dadurch, dagegen
8	Local	da, dahin, hierunter, darin
9	Pronominal	darüber, damit
10	Relative_WH	wovon, warum, weshalb, wie, wann, woran, wodurch
11	Numeral	trebly, vielfach, zweimal, dreimal
12	Causal	deshalb
13	Modal	sehr, ziemlich, gleichermassen, lange

Table A.4: Semantic classification of adverbs.

Number	Name of Class	Introduced Relation
1	Possibility	hasPossibility
2	Attitude	hasAttitude
3	Temporal	hasTime
4	Aspect	hasAspect
5	Frequency	hasFrequeny
6	Manner	hasManner
7	Focus	hasFocus
8	Local	hasLocation
9	Pronominal	
10	Relative_WH	
11	Numeral	hasNumber
12	Causal	hasCause
13	Modal	hasMode

Table A.5: Relations introduced by the adverbs.

A.2.5 Semantic Classification of Verbs

Number	Verb	Frequency	Number	Verb	Frequency
1	geben	179	18	stellen	47
2	liegen	177	19	finden	47
3	gehen	175	20	rechnen	44
4	kommen	174	21	halten	43
5	stehen	158	22	meinen	43
6	machen	143	23	drohen	42
7	gelten	143	24	fehlen	42
8	sehen	116	25	kosten	38
9	bleiben	114	26	nehmen	36
10	setzen	74	27	klagen	33
11	zeigen	70	28	bekommen	33
12	sagen	66	29	suchen	32
13	glauben	65	30	kaufen	32
14	brauchen	55	31	lassen	32
15	steigen	53	32	scheinen	32
16	bringen	51	33	bestehen	31
17	bieten	49	34	fallen	30

Table A.6: The list of verbs which appear more than thirty times in the corpus.

Number	Name of Class	Examples
1	General Existence	
1.1	Stative Existence	sein, passieren, geschehen
1.2	Active Existence	andauern, anhalten, bleiben
1.3	Causative Existence	anfertigen, erzeugen, gründen
2	Special Existence	
2.1	Constitutive Existence	anfangen, einsetzen, auftreten
2.2	Contextual Existence	erscheinen, fehlen, vorlegen
3	Difference	
3.1	General Difference	abheben von, unterscheiden von, varirreren
3.2	Change	ändern, schmelzen, sinken
3.3	Causative Change	reduzieren, anheben, kürzen
4	Relation	
4.1	General Relation	stehen in, verbinden mit, beziehen auf
4.2	Identity	gleich sein, kongruieren, übereinstimmen
4.3	Structure	abgrenzen, einordnen, eingliedern
4.4	Part Of	angehören, beinhalten, haben
4.6	Base	aufbauen auf, basieren auf, stützen auf
4.7	Result	ableiten aus, schlußfolgern aus, schließen aus
4.8	Scope	richten auf, zielen auf, abzielen auf
4.9	Evaluation	betrachten als, sehen als, ansehen als
4.10	Orientation	achten, folgen, richten an
4.11	Attention	achten auf, denken an, bedenken
4.12	Ignorance	absehen von, übergehen, übersehen
4.13	Intellectual Activity	beschäftigen mit, konzentrieren auf, tangieren
4.14	Investigation	ergründen, erkunden, unersuchen
4.15	Testing	kontrollieren, prüfen, erproben
5	Scope of Action	ablehnen, anraten, anweisen
6	Expression	
6.1	Tell	erzählen, anvertrauen, bekanntmachen
6.2	Communicate	ausrichten, betsellen, vermitteln
6.3	Discuss	besprechen, erörtern, beraten
7	Need	
7.1	Possession	verkaufen, verlieren, borgen
7.2	Consum	essen, kosten, speisen
7.2	Sleep	aufwachen, erwachen, schlafen

Table A.7: Semantic classification of verbs by Schumacher (1986).

Appendix B

The Generic Rules Used for Extraction

B.1 Genitive Paraphrases

```
disposesOver
 if one of the concepts has GN=group
 and the other GN=possession
 ==> concept[GN=group] disposesOver concept[GN=possession]

hasDimension
 if one of the nouns has GN=quantity
 and the second GN=person/possession
 ==> noun[GN=!quantity] hasDimension noun [GN=quantity]

hasEvent
 if one of the concepts has GN=event
 and the second GN=!event
 ==> concept[GN=!event] hasEvent concept[GN=event]

hasAttribute
 if one of the concepts has GN=attribute
 and the second GN=!attribute
 ==> concept[GN=!attribute] hasAttribute concept[GN=attribute]

hasLocation
 if one of the concepts has GN=location
 and the other has GN=!location
 ==> concept[GN=!location] hasLocation concept[GN=location]
```

B.2 Prepositional Paraphrases

```
disposesOver
 if one of the concepts has GN=group
 and the other GN=possession
 ==> concept[GN=group] disposesOver concept[GN=possession]

hasDimension
 if one of the concepts has GN=quantity
 and the other is different from quantity
 ==> concept[GN!=quantity] hasDimension concept[GN=quantity]

hasEvent
 if one of the concepts has GN=event
 and the second GN=!event
 ==> concept[GN=!event] hasEvent concept[GN=event]
```

```
hasAttribute
 if one of the concepts has GN=attribute
 and the second GN=!attribute
 ==> concept[GN=!attribute] hasAttribute concept[GN=attribute]

hasLocation
 if one of the concepts has GN=location
 and the other has GN=!location
 ==> concept[GN=!location] hasLocation concept[GN=location]

hasAffiliation
 if one of the concepts has GN=person/group
 and the other GN=group
 ==> concept[GN=person/group] hasAffiliation concept[GN=group]
```

B.3 Premodification Phenomena

```
  modifier1[PoS=adv/adj][SC=semanticClass]
 + modifier2[PoS=adj][SC=semanticClass]{0,1}
 + noun[PoS=noun][GN=semanticClass]

  (modifier[PoS=adj][SC=semanticClass]
 + separator[PoS=comma]){0,n}
 + modifier[PoS=adj][SC=semanticClass]
 + separator[PoS=conj/comma]
 + modifier[PoS=adj][SC=semanticClass]
 + noun[PoS=noun][GN=semanticClass]
```

```
modifier1[PoS=adj][SC=semanticClass]
+ noun[PoS=noun][GN=semanticClass]
==> relationDerived(noun, modifier1)

modifier1[PoS=adj][SC=semanticClass]
+ modifier2[PoS=adj][SC=semanticClass]
+ noun[PoS=noun][GN=semanticClass]
==> relationIntroducedByModfier2(noun, modifier2)
==> relationIntroducedByModfier1(modifier1, modifier2 noun)

modifier1[PoS=adv][SC=semanticClass]
+ modifier2[PoS=adj][SC=semanticClass]
+ noun[PoS=noun][GN=semanticClass]
==> relationIntroducedByModfier2(noun, modifier2)
==> relationIntroducedByModfier1(modifier2, modifier1)

modifier1[PoS=adj][SC=semanticClass]
+ separator[PoS=comma]
+ modifier2[PoS=adj][SC=semanticClass]
+ noun[PoS=noun][GN=semanticClass]
==> relationIntroducedByModfier1(noun, modifier1)
==> relationIntroducedByModfier2(noun, modifier2)

modifier1[PoS=adj][SC=semanticClass]
+ separator[PoS=conj]
+ modifier2[PoS=adj][SC=semanticClass]
+ noun[PoS=noun][GN=semanticClass]
```

```
  ==> relationIntroducedByModfier1(noun, modifier1)
  ==> relationIntroducedByModfier2(noun, modifier2)

  modifier1[PoS=adj][SC=semanticClass]
  + separator[PoS=punct]
  + modifier2[PoS=adj][SC=semanticClass]
  + separator[PoS=conj]
  + modifier3[PoS=adj][SC=semanticClass]
  + noun[PoS=noun][GN=semanticClass]
  ==> relationIntroducedByModfier1(noun, modifier1)
  ==> relationIntroducedByModfier2(noun, modifier2)
  ==> relationIntroducedByModfier3(noun, modifier3)
```

B.4 NE Instantiations

```
organization
  NE[ne-organization[descriptor, orgname]]
  ==> instanceOf(orgname, descriptor)
  ==> subClassOf(descriptor, organization)

  NE[ne-organization]
  ==> instanceOf(ne-orgname, organization)

  NE[ne-organization[ne-designator, orgname]]
  ==> instanceOf(ne-orgname, organization)
  ==> isOrganizedAs(organization, ne-designator)

location
```

```
  NE[ne-location[loctype, locname]]
  ==> instanceOf(locname, loctype)
  ==> subClassOf(loctype, location)

  NE[ne-location]
  ==> instanceOf(ne-location, location)

person
  NE[ne-person[position]]
  ==> instanceOf(ne-person, person)
  ==> subClassOf(ne-position, position)
  ==> occupiesPosition(ne-person, position)

  NE[ne-person]
  ==> instanceOf(ne-person, person)

time
  NE[ne-duration[date]]
  ==> hasTimeUnitValue(ne-duration, string)
  ==> instanceOf(string, stringValue)

quantity
  NE[quantity]
  ==> hasQuantityValue(ne-quantity, string)
  ==> instanceOf(string, stringValue)

money
  NE[ne-money[currency]]
  ==> hasNumberValue(ne-money, string)
  ==> instanceOf(string, stringValue)
  ==> instanceOf(ne-currency, currency)


hyphen compounds
  NE[ne-organization]-NE[ne-person[position]]
  ==> instanceOf(ne-person, person)
  ==> instanceOf(ne-organization, organization)
  ==> occupiesPosition(ne-person, position)
  ==> hasPosition(ne-organization, ne-person)
```

B.5 Grammatical Functions

```
GEBEN
   es[PoS=pron]
  + VG[STEM=geben]
  + NP[GF=DOBJ]
  ==>exists(DOBJ)

   es[PoS=pron]
  + VG[STEM=geben]
  + NP[GF=DOBJ]
  + PP[GF=PP_ADJUNCT]
  ==> exists(DOBJ, PP_ADJUNCT)

   es[PoS=pron]
  + VG[STEM=geben]
  + NP[GF=DOBJ][SC=change]
  + PP[GF=PP_ADJUNCT]
  ==> exists(DOBJ, PP_ADJUNCT)

   es VG[STEM=geben]
  + NP[GF=DOBJ][SC=information]
  + PP[GF=PP_ADJUNCT]
  ==> exists(DOBJ, PP_ADJUNCT)

  NP[GF=SUBJ][SC=person]
  + VG[STEM=geben]
  + PP[GF=IOBJ][SC=person]
  + NP[GF=DOBJ][SC=object]
  ==>changePossessionRelation(SUBJ, DOBJ, IOBJ)

  NP[GF=SUBJ][SC=person]
  + VG[STEM=geben]
  + NP[GF=DOBJ][SC=object]
  + PP[GF=IOBJ][SC=person]
  ==>changePossessionRelation(SUBJ, DOBJ, IOBJ)

  NP[GF=SUBJ][SC=person]
  + VG[STEM=geben]
  + PP[GF=IOBJ][SC=person]
  + NP[GF=DOBJ][SC=abstract]
  ==>changePossessionRelation(SUBJ, DOBJ, IOBJ)

  + NP[GF=SUBJ][SC=person]
  + VG[STEM=geben]
```

```
  + NP[GF=DOBJ][SC=abstract]
  + PP[GF=IOBJ][SC=person]
  ==>changePossessionRelation(SUBJ, DOBJ, IOBJ)

LIEGEN
  NP[GF=SUBJ]
  + VG[STEM=liegen]
  + PP[GF=PP_ADJUNCT][SC=location]
  ==>hasLocation(SUBJ, PP_ADJUNCT)



  NP[GF=SUBJ]
  + VG[STEM=liegen]
  + PP[GF=IOBJ][SC=quantity]
  ==>hasValue(SUBJ, IOBJ)

  NP[GF=SUBJ]
  + VG[STEM=liegen]
  + PP[GF=PP_ADJUNCT[prep=im/auf]][SC=artefact]
  ==>lies(SUBJ, PP_ADJUNCT)



  NP[GF=SUBJ]
  + VG[STEM=liegen]
  + PP[GF=IOBJ][SC=!(artefact|quantity|location)]
 ==>hasConnection(SUBJ, IOBJ)

GEHEN
```

```
  NP[GF=SUBJ]
  + VG[STEM=gehen]
  + ADVP[GF=DOBJ]
  ==>hasTendency(SUBJ, ADVP)

  NP[GF=SUBJ]
  + VG[STEM=gehen]
  + PP[GF=IOBJ][SC=location]
  ==> movesTo(SUBJ, IOBJ)

  NP[GF=SUBJ]
  + VG[STEM=gehen]
  + PP[GF=IOBJ[prep=an]][SC=person|group]
  ==> receives(IOBJ, SUBJ)

  NP[GF=SUBJ][STEM=es]
  + VG[STEM=gehen]
  + PP[GF=IOBJ[prep=um]][SC=artifact]
  ==> dealsWith(SUBJ, IOBJ)

KOMMEN
 NP[GF=SUBJ]
  + VG[STEM=kommen]
  + PP[GF=PP_ADJUNCT][SC=location]
  ==> hasLocation(SUBJ, PP_ADJUNCT)

  NP[GF=SUBJ]
  + VG[STEM=kommen]
  + PP[GF=PP_ADJUNCT][SC=quantity|time]
  ==> hasValue(SUBJ, PP_ADJUNCT)

  NP[GF=SUBJ]
  + VG[STEM=kommen]
  + NP[GF=DOBJ]
  + (PP[PP_ADJUNCT][SC=time])?
  ==> comes(SUBJ, DOBJ, PP_ADJUNCT?)

  NP[GF=SUBJ]
  + VG[STEM=kommen]
  + NP[GF=DOBJ][SC!=person]
  + (PP[GF=IOBJ])?
  + ADVP[STEM=hinzu]
  ==> isAddedTo(SUBJ, DOBJ, IOBJ?)

STEHEN
  NP[GF=SUBJ]
  + VG[STEM=stehen]
  + ADJP[GF=DOBJ]
  ==> isConsidered(SUBJ, DOBJ)

  NP[GF=SUBJ][SC=group|person]
  + VG[STEM=stehen]
  + PP[GF=IOBJ]
  ==> isInSituation(SUBJ, IOBJ)

  NP[GF=SUBJ]
  + VG[STEM=stehen]
  + PP[GF=IOBJ][SC=quantity]
  + PP[GF=PP_ADJUNCT]?
  ==> hasValue(SUBJ, IOBJ, PP_ADJUNCT?)

  NP[GF=SUBJ]
  + VG[STEM=stehen]
  + (NP[GF=DOBJ])?
  + PP[GF=IOBJ]
  ==> stands(SUBJ, DOBJ?, IOBJ)

MACHEN
  NP[GF=SUBJ]
  + VG[STEM=machen]
  + NP[GF=DOBJ]
  + (PP[GF=PP_ADJUNCT])?
  ==>does(SUBJ, DOBJ, PP_ADJUNCT)

  NP[GF=SUBJ]
  + VG[STEM=machen]
  + NP[GF=DOBJ][SC=person]
  + PP[GF=IOBJ[prep=zu|aus]][SC=#1]
  ==>change(SUBJ, DOBJ, IOBJ)

GELTEN
  NP[GF=SUBJ]
  + VG[STEM=gelten]
  + NP[GF=IOBJ]
  ==> concerns(SUBJ, DOBJ)

  NP[GF=SUBJ]
  + VG[STEM=gelten]
  + PP[GF=IOBJ[prep=als]]
  ==> isConsidered(SUBJ, IOBJ)

  NP[GF=SUBJ]
  + VG[STEM=gelten]
  + NP[GF=DOBJ]
  + PP[GF=PP_ADJUNCT[prep=als]]
  ==> isConsidered(SUBJ, DOBJ, PP_ADJUNCT)

SEHEN
  NP[GF=SUBJ][SC=person]
  + VG[STEM=sehen]
  + NP[GF=DOBJ][SC=!person]
  + (PP[GF=PP_ADJUNCT])?
  ==> hasOpinion(SUBJ, DOBJ, PP_ADJUNCT?)

  NP[GF=SUBJ][SC=person|group]
  + VG[STEM=sehen]
  + NP[GF=DOBJ]
  + PP[GF=IOBJ[prep=als]]
  ==> compares(SUBJ, DOBJ, IOBJ)

BLEIBEN
  NP[GF=SUBJ]
  + VG[STEM=bleiben]
  + NP[GF=DOBJ]
  + (PP[GF=PP_ADJUNCT])?
  ==> remains(SUBJ, DOBJ, PP_ADJUNCT)?

SETZEN
  NP[GF=SUBJ][SC=person|group]
  + VG[STEM=setzen]
  + PP[auf][GF=IOBJ[prep=auf]]
  + (PP[GF=PP_ADJUNCT])?
  ==> countsOn(SUBJ, IOBJ, PP_ADJUNCT?)
```

Appendix C

DL and Formalization

C.1 Description Logic Syntax and Semantics

Before describing the syntax and semantics of DL we have to mention some notational conventions: the letters A and B are atomic concepts, the letter R stands for atomic roles and the letters C and D are used for concept descriptions.

Concept descriptions in $\mathcal{AL}$ are formed according to the following concept roles (Baader et al., 2003):

$$\begin{aligned} C, D \quad \longrightarrow \quad & A \mid \top \mid \bot \mid \neg A \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C \mid \exists R.b \mid \\ & \geq nR \mid \ \leq nR \mid \ = nR \mid \ \geq nR.C \mid \ \leq nR.C \mid \ = nR.C \end{aligned}$$

Table C.1 shows the syntax of common concept constructors.

Role constructors are interpreted as binary relations which means that they can be used for usual operations on binary relations. Table C.2 lists some of them and the corresponding syntax.

As already mentioned DL relates the concepts and roles to each other by axioms. Table C.3 list some of them.

Abstract syntax	Concrete syntax	Name
$\top$	`TOP`	Top
$\bot$	`BOTTOM`	Bottom
$C_1 \sqcap ... C_n$	`(and C1 ... Cn)`	intersection
$C_1 \sqcup ... C_n$	`(or C1 ... Cn)`	union
$\neg C$	`(not C)`	negation
$\forall R.C$	`(all R C)`	value restriction
$\exists R.\top$	`(some R)`	limited existential quantification
$\exists R.C$	`(some R C)`	existential quantification
$\geq nR$	`(at-least n R)`	at-least number restriction
$\leq nR$	`(at-most n R)`	at-most number restriction
$= nR$	`(exactly n R)`	exact number restriction
$\geq nR.C$	`(at-least n R C)`	qualified at-least restriction
$\leq nR.C$	`(at-most n R C)`	qualified at-most restriction
$= nR.C$	`(exactly n R C)`	qualified exact restriction
$u_1 = u_2$	`(same-as u1 u2)`	same-as agreement
$R_1 \subseteq R_2$	`(subset R1 R2)`	role-value-map
$\exists R.I_1 \sqcap ... \sqcap \exists R.I_n$	`(fillers R I1 ... In)`	role fillers
$I_1 \sqcup ... \sqcup I_n$	`(one-of I1 ... In)`	one-of

Table C.1: Concrete syntax of concept constructors.

Abstract syntax	Concrete syntax	Name
$\top$	`TOP`	universal role
$R_1 \sqcap ... R_n$	`(and R1 ... Rn)`	intersection
$R_1 \sqcup ... R_n$	`(or R1 ... Rn)`	union
$\neg R$	`(not R)`	complement
R^-	`(inverse R)`	inverse
$R_1 \circ ... R_n$	`(compose R1 ... Rn)`	compose
R^+	`(transitive-closure R)`	transitive closure
R^*	(transitive-reflexive closure)	reflexive-transitive closure
$R\|_C$	(restrict R C)	role restriction
$id(C)$	(identity C)	identity

Table C.2: Concrete syntax of role constructors.

The interpretation of concepts and roles exhibits the connection between Description Logic and Predicate Logic. Since in the interpretation every atomic concept corresponds to an unary relation, and every role to a binary relation, concepts and roles can be viewed as unary and binary predicates, respectively.

Abstract Syntax	Concrete Syntax	Name
$A \equiv C$	(define-concept A C)	concept definition
$A \sqsubseteq C$	(define-primitive-concept A C)	primitive concept introduction
$C \sqsubseteq D$	(implies C D)	general inclusion axiom
$R \equiv S$	(define-role R S)	role definition
$R \sqsubseteq S$	(define-primitive-role R S)	primitive role introduction
$C(a)$	(instance a C)	concept assertion
$R(a, b)$	(related a b R)	role assertion

Table C.3: Concrete syntax of axioms.

An interpretation $\mathcal{I}$ consists of a non-empty set $\Delta^{\mathcal{I}}$ (the *domain* of the interpretation) and an interpretation function which assigns to every atomic concept A a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to every atomic role R a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Based on this definition each of the constructors listed in this section can be reformulated by using first-order predicate logic[1].

As already mentioned in Section 3.2.1 the basic DL language $\mathcal{AL}$ can be extended in order to obtain more expressive DL. The three main possibilities for extending $\mathcal{AL}$ are by introducing new concept constructors, role constructors and formulating restrictions on role interpretations. So for example, the extension of $\mathcal{AL}$ with the concept constructor negation is written as $\mathcal{ALC}$. The role hierarchy, which imposes restrictions on the interpretation of roles in a certain domain D, is indicated by appending a $\mathcal{H}$ to the DL. Table C.4 lists some of the possible extensions for DL.

OWL DL corresponds to $\mathcal{SHOIN}$(D).

[1] From the constructors presented in this section transitive and reflexive-transitive closure are the only constructors that cannot be expressed by first-order predicate logic.

Extension	**Symbol**
Negation	$\mathcal{C}$
Number restrictions	$\mathcal{N}$
Qualified number restrictions	$\mathcal{Q}$
Role hierarchy	$\mathcal{H}$
Role inverse	$\mathcal{I}$
Nominals	$\mathcal{O}$
Functional roles	$\mathcal{F}$
$\mathcal{ALC}$ + transitive roles	$\mathcal{S}$

Table C.4: DL extensions.

C.2 OWL's Functional Syntax

```
Namespace(=<http://dfki.lt.de/formOwlFull.owl#>)
Namespace(rdfs=<http://www.w3.org/2000/01/rdf-schema#>)
Namespace(owl2xml=<http://www.w3.org/2006/12/owl2-xml#>)
Namespace(owl=<http://www.w3.org/2002/07/owl#>)
Namespace(xsd=<http://www.w3.org/2001/XMLSchema#>)
Namespace(rdf=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
Namespace(formOwlFull=<http://dfki.lt.de/formOwlFull.owl#>)

Ontology(<http://dfki.lt.de/formOwlFull.owl>

SubClassOf(Größte Dimension)
SubClassOf(Deutsch Affiliation)
SubClassOf(DeutscheChemiekonzern Chemiekonzern)
SubClassOf(DeutscheChemiekonzern
        ObjectAllValuesFrom(hasDimension DimensionRelation))
SubClassOf(EarningRelation Relation)
SubClassOf(EarningRelation
        ObjectSomeValuesFrom(hasEarningValue Million))
SubClassOf(EarningRelation
        ObjectSomeValuesFrom(hasEarningTime Monat))
SubClassOf(Affiliation owl:Thing)
SubClassOf(Chemiekonzern
        ObjectAllValuesFrom(hasEarning EarningRelation))
SubClassOf(Chemiekonzern
        ObjectAllValuesFrom(hasAffiliation AffiliationRelation))
SubClassOf(Chemiekonzern Konzern)
SubClassOf(DimensionRelation
        ObjectSomeValuesFrom(hasDimensionValue Größte))
SubClassOf(DimensionRelation Relation)
SubClassOf(Monat TimeUnit)
SubClassOf(Number owl:Thing)
SubClassOf(Group owl:Thing)
SubClassOf(Konzern Group)
SubClassOf(AffiliationRelation Relation)
SubClassOf(AffiliationRelation
        ObjectSomeValuesFrom(hasAffiliationValue Deutsch))
SubClassOf(Relation owl:Thing)
SubClassOf(TimeUnit owl:Thing)
SubClassOf(Million Number)
SubClassOf(Dimension owl:Thing)


FunctionalObjectProperty(hasDimensionValue)
ObjectPropertyDomain(hasDimensionValue DimensionRelation)
ObjectPropertyRange(hasDimensionValue Dimension)
```

```
FunctionalObjectProperty(hasEarningTime)
ObjectPropertyDomain(hasEarningTime EarningRelation)
ObjectPropertyRange(hasEarningTime TimeUnit)
ObjectPropertyDomain(hasAffiliation Group)
ObjectPropertyRange(hasAffiliation AffiliationRelation)
ObjectPropertyDomain(hasDimension Affiliation)
ObjectPropertyDomain(hasDimension Group)
ObjectPropertyRange(hasDimension DimensionRelation)
FunctionalObjectProperty(hasEarningValue)
ObjectPropertyDomain(hasEarningValue EarningRelation)
ObjectPropertyRange(hasEarningValue Number)
ObjectPropertyDomain(hasEarning Group)
ObjectPropertyRange(hasEarning EarningRelation)
FunctionalObjectProperty(hasAffiliationValue)
ObjectPropertyDomain(hasAffiliationValue AffiliationRelation)
ObjectPropertyRange(hasAffiliationValue Affiliation)

DataPropertyDomain(hasNumberValue Number)
DataPropertyRange(hasNumberValue xsd:integer)
DataPropertyDomain(hasTimeUnitValue TimeUnit)
DataPropertyRange(hasTimeUnitValue xsd:string)

ClassAssertion(<http://dfki.lt.de/formOwlFull.owl#17> Million)
ClassAssertion(BASF Chemiekonzern)
ClassAssertion(neun Monat)
)
```

C.3 Logical Formalization

Classes

Chemiekonzern

Chemiekonzern $\sqsubseteq$ $\forall$ hasEarning EarningRelation

Chemiekonzern $\sqsubseteq$ $\forall$ hasAffiliation AffiliationRelation

Chemiekonzern $\sqsubseteq$ Konzern

Konzern

Konzern $\sqsubseteq$ Group

Million

Million $\sqsubseteq$ Number

Monat

Monat $\sqsubseteq$ TimeUnit

Thing

Affiliation

Affiliation $\sqsubseteq$ Thing

AffiliationRelation

AffiliationRelation $\sqsubseteq$ Relation

AffiliationRelation $\sqsubseteq \exists$ hasAffiliationValue Deutsch

Deutsch

Deutsch $\sqsubseteq$ Affiliation

DeutscheChemiekonzern

DeutscheChemiekonzern $\sqsubseteq$ Chemiekonzern

DeutscheChemiekonzern $\sqsubseteq \forall$ hasDimension DimensionRelation

Dimension

Dimension $\sqsubseteq$ Thing

DimensionRelation

DimensionRelation $\sqsubseteq \exists$ hasDimensionValue Grösste

DimensionRelation $\sqsubseteq$ Relation

EarningRelation

EarningRelation $\sqsubseteq$ Relation

EarningRelation $\sqsubseteq \exists$ hasEarningValue Million

EarningRelation $\sqsubseteq \exists$ hasEarningTime Monat

Group

Group $\sqsubseteq$ Thing

größte

Größte $\sqsubseteq$ Dimension

Number

Number $\sqsubseteq$ Thing

Relation

Relation $\sqsubseteq$ Thing

TimeUnit

TimeUnit $\sqsubseteq$ Thing

Object properties

hasAffiliation

$\exists$ hasAffiliation Thing $\sqsubseteq$ group

$\top \sqsubseteq \forall$ hasAffiliation AffiliationRelation

hasAffiliationValue

$\exists$ hasAffiliationValue Thing $\sqsubseteq$ AffiliationRelation

$\top \sqsubseteq \forall$ hasAffiliationValue Affiliation

hasDimension

$\exists$ hasDimension Thing $\sqsubseteq$ Affiliation

$\exists$ hasDimension Thing $\sqsubseteq$ Group

$\top \sqsubseteq \forall$ hasDimension DimensionRelation

hasDimensionValue

$\exists$ hasDimensionValue Thing $\sqsubseteq$ DimensionRelation

$\top \sqsubseteq \forall$ hasDimensionValue Dimension

hasEarning

$\exists$ hasEarning Thing $\sqsubseteq$ Group

$\top \sqsubseteq \forall$ hasEarning EarningRelation

hasEarningTime

$\exists$ hasEarningTime Thing $\sqsubseteq$ EarningRelation

$\top \sqsubseteq \forall$ hasEarningTime TimeUnit

hasEarningValue

$\exists$ hasEarningValue Thing $\sqsubseteq$ EarningRelation

$\top \sqsubseteq \forall$ hasEarningValue Number

Data properties

hasNumberValue

hasTimeUnitValue

Individuals

17

17 : Million

BASF

BASF : Chemiekonzern

neun

neun : Monat

Appendix D

Statistics

Relation	Number
hasDimension	683
hasLocation	153
hasEvent	370
hasAttribute	267
disposesOver	45
hasAffilitation	164

Table D.1: Frequencies for relations from prepositional phrases.

Relation	Number
hasDimension	428
hasLocation	143
hasEvent	222
hasAttribute	135
disposesOver	30

Table D.2: Frequencies for relations from genitival phrases.

Phenomenon	Pattern Match	Rule Match
Compounds	6776	22142
Modification phenomena	11178	2614
Prepositional phrases	2546	1684
Genitive phrases	1637	1137
Grammatical functions	12164	2459
Instantiations	7812	7812

Table D.3: Coverage of rules from patterns.

Structure	Frequency	Complete coverage	Partial coverage
Adjective noun	9478	2289	295
Adjective adjective noun	906	187	296
Adverb adjective noun	598	123	335
Adjective, adjective noun	44	1	14
Adjective conj adjective noun	97	12	23
Adjective, adjective conj adjective noun	55	2	1

Table D.4: Frequencies for modification phenomena.

Structure	Frequency
Subj DObj	1215
Subj DObj IObj	181
Subj DObj IObj PP$_A$djunct	193
Subj DObj PP_Adjunct	870

Table D.5: Frequencies for grammatical functions.

Phenomenon	Frequency
Organization	2765
Person	1345
Location	2659
Money	830
TimeUnit	195
Quantity	62

Table D.6: Frequencies for instantiations.

Phenomenon	Number of Classes	Number of Relations
Compounds	6386	1
Modification phenomena	5760	33
Prepositional phrases	2172	6
Genitive phrases	1535	5
Verbs	2900	1148

Table D.7: Number of unique classes and relations.

Number of Classes	Number of Relations
17462	1183

Table D.8: Number of unique classes and relations.

Bibliography

Guadalupe Aguado de Cea, Asunción Gómez-Pérez, Elena Montiel-Ponsoda, and María del Carmen Suárez-Figueroa. Natural language-based approach for helping in the reuse of ontology design patterns. In *Proceedings of the EKAW Conference*, 2008.

Niraj Aswani, Valentin Tablan, Katina Bontcheva, and Hamish Cunningham. Indexing and querying linguistic metadata and document content. In *Proceedings of Fifth International Conference on Recent Advances in Natural Language Processing (RANLP-2005)*, 2005.

Nathalie Aussenac-Gilles and Marie-Pauler Jacques. Designing and evaluating patterns for relation acquisition from texts with caméléon. *Terminology*, 5(1), 2008.

Franz Baader. Description logic terminology. In Franz Baader, Diego Clavanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors, *Description Logic Handbook: Theory, Impementation and Applications.* Cambridge University Press„ Cambridge, UK; New York, NY, USA, 2003.

Franz Baader and Werner Nutt. Basic description logics. In Franz Baader, Diego Clavanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors, *Description Logic Handbook: Theory, Impementation and*

Applications. Cambridge University Press,, Cambridge, UK; New York, NY, USA, 2003.

Franz Baader, Diego Clavanese, Deborah Mcguiness, Daniele Nardi, and Peter F. Patel-Schneider. *Description Logic Handbook: Theory, Impementation and Applications*. Cambridge University Press, 2003.

Daniel Bachlechner, Christian Leibold, Marcus Spies, Andrea Bellandi, Barbara Furletti, Silvia Figini, and Paolo Giudici. Knowledge modelling and management routines, tools and modules. Technical report, MUSING Deliverable D3.1, 2008.

Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. *OWL Web Ontology Language Reference*. W3C, 2003.

Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5), 2001.

Sean Boisen, Michael R. Crystal, Richard Schwartz, Rebecca Stone, and Ralph Weischedel. Annotating resources for information extraction. In *Proceedings of the LREC 2000*, 2002.

Thorsten Brants. Tnt - a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000*. Association for Computational Linguistics, 2000.

Eric Brill. A simple rule-bases part of speech tagger. In *Proceedings of the Third Annual Conference on Applied Natural Language Processing (ACL)*. Association for Computational Linguistics, 1992.

Paul Buitelaar and Thierry Declerck. Linguistic annotation for the semantic web. In *Annotation for the Semantic Web*, volume 96. IOS Press, 2003.

Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini. Ontology learning from text: An overview. In *Ontology Learning from Text: Methods, Evaluation and Applications. Frontiers in Artificial Intelligence and Applications*, volume 123. IOS Press, 2005.

Hadumod Bußmann. *Lexikon der Sprachwissenschaft.* Alfred Kröner Verlag, 2008.

Eugene Charniak. A maximum-entropy-inspired parser. In *Proceedings of the 38th Annual Meeting of the Association of Computational Linguistics*, Hong Kong, China, 2000.

Massimiliano Ciaramita, Aldo Gangemi, Esther Ratsch, Jasmin Saric, and Isabel Rojas. Unsupervised learning of semantic relations for molecular biology ontologies. In Paul Buitelaar and Philipp Cimiano, editors, *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 91–107. IOS Press, Amsterdam, 2008.

Philipp Cimiano and Johanna Völker. Towards large-scale, open-domain and ontology-based named entity classification. In *Proceedings of RANLP*, 2005.

Philipp Cimiano, Siegfried Handschuh, and Steffen Staab. Towards the self-annotating web. In *Proceedings of the 13th World Wide Web Conference*, 2004.

Philipp Cimiano, Andreas Hotho, and Steffen Staab. Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence Research*, 24, 2005.

Fabio Ciravegna and Yorick Wilks. Designing adaptive information extraction for the semantic web in amilcare. In Siegfried Handschuh and Steffen Staab, editors, *Annotation for the Semantic Web.* IOS Press, Amsterdam, 2003.

Stephen Clark and David Weir. Class-based probability estimation using a semantic hierarchy. *Computational Linguistics*, 28(2), 2002.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley-Interscience, New York, NY, USA, 1991.

Hamish Cunningham. Gate, a general architecture for text engineering. *Computers and the Humanities*, 36:223–254, 2002.

Éric De la Clergerie. Spécifications du service d'extraction supervisée d'ontologies. Technical report, SCRIBO Project, 2009.

Thierry Declerck. A set of tools for integrating linguistic and non-linguistic information. In *Proceedings of SAAKM (ECAI Workshop)*, 2002.

Elke Donalies. *Basiswissen Deutsche Grammatik.* Francke Verlag, Tübingen, 2007.

Witold Drozdzynski, Hans-Ulrich Krieger, Jakub Piskorski, Ulrich Schäfer, and Feiyu Xu. Shallow processing with unification and typed feature structures — foundations and applications. *Künstliche Intelligenz*, 1:17–23, 2004.

Duden. *Grammatik der deutschen Gegenwartssprache.* Dudenverlag, Mannheim/Wien/Zürich, 4 edition, 2006.

Johannes Erben. *Einführung in die deutsche Wortbildungslehre.* Erich Schmidt Verlag GmbH and Co., Berlin, 3 edition, 1993.

Christiane Fellbaum. *WordNet: An Electronic Lexical Database.* MIT Press, Cambridge, 1998.

Charles J. Fillmore. Frame semantics. *Linguistics in the Morning Calm*, pages 111–137, 1982.

Wolfgang Finkler and Günter Neumann. Morphix. a fast realization of a classification-based approach to morphology. In *Proceedings of 4th OFAI*, 1988.

Wolfgang Fleischer and Irmhild Barz. *Wortbildung der deutschen Gegenwartssprache.* Niemeyer, Tuebingen, 2 edition, 1995.

William Frawley. *Linguistic semantics.* Erlbaum, 1992.

Pablo Gamallo, Alexandre Agustini, and Gabriel P. Lopes. Selection restrictions acquisition from corpora. *Computer Science*, 2258, 2001.

Pablo Gamallo, Marco Gonzalez, Alexandre Agustini, Gabriel Lopes, and Vera S. de Lime. Mapping syntactic dependencies onto semantic relations. In *Proceedings of the ECAI Workshop on Machine Learning and Natural Language Processing for Ontology Engineering*, 2002.

Aldo Gangemi, Stefano David, Guadalupe Aguado de Cea, Mari Carmen Suárez-Figueroa, Elena Montiel-Ponsoda, and María Poveda. A library of ontology design patterns: reusable solutions for collaborative design of networked ontologies. Technical report, NeON Project, 2008.

Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis – Mathematical Foundations.* Springer, 1991.

Horst Geckeler and Wolf Dietrich. *Einführung in die französische Sprachwissenschaft.* Erich Schmidt Verlag, 2007.

Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

Iryna Gurevych and Hendrik Niederlich. Accessing germanet data and computing semantic relatedness. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'2005)*, 2005.

Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of 14th International Conference on Computational Linguistics (COLING-92)*, pages 539–545, Nantes, France, 1992.

Gerhard Helbig and Joachim Buscha. *Deutsche Grammatik*. Langenscheidt, 18 edition, 1998.

Atanas Kiryakov, Borislav Popov, Ivan Terziev, Dimitar Manov, and Damyan Ognyanoff. Semantic annotation, indexing, and retrieval. *ISWC 2003 Special Issue Journal of Web Semantics*, 1(2):671–680, 2004.

Paul Kogut and William Holmes. Aerodaml: Applying information extraction to generate daml annotations from web pages. In *First International Conference on Knowledge Capture (K-CAP)*, 2001.

Kimmo Koskenniemi. *Two-level Morphology: A General Computational Model for Word-Form Recognition and Production*. University of Helsinki: Department of General Linguistics, 1983.

Hans-Ulrich Krieger, Bernd Kiefer, and Thierry Declerck. A framework for temporal representation and reasoning in business intelligence applications. In Knut Hinkelmann, editor, *AI Meets Business Rules and Process Management. Papers from AAAI 2008 Spring Symposium*, pages 59–70. AAAI Press, 2008.

Claudia Kunze and Lothar Lemnitzer. Germanet - representation, visualization, application. In *Proceedings of the LREC 2002*, 2002.

Sun-Muk Lee. *Untersuchungen zur Valenz des Adjektivs in der deutschen Gegenwartssprache*. Lang, Frankfurt am Main, Germany, 1994.

Susanne Leischner. *Die Stellung des attributiven Adjektivs im Französischen*. Gunter Narr Verlag, 1990.

Dekang Lin. Dependency-based evaluation of minipar. In *Proceedings of the Workshop on the Evaluation of Parsing Systems*, Granada, Spain, 1998.

Anne Lobeck. *Discovering Grammar: An Introduction to English Sentence Structure.* Oxford University Press, 2000. URL `http://www.ac.wwu.edu/~annelob/TESOL402assignment3.htm`.

Michael Lohde. *Wortbildung des modernen Deutschen.* Francke, Tübingen, 2006.

Alexander Maedche and Steffen Staab. Measuring similarity between ontologies. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW).* Springer, 2002.

Alexander Maedche, Viktor Pekar, and Steffen Staab. Ontology learning part one - on discovering taxonomic relations from the web. In *Web Intelligence.* Springer, 2002.

Nuno Marques. *Uma Metodologia para a Modelação Estatística da Subcategorização Verbal.* Universidade Nova de Lisboa, Lisboa, Portugal, 2000.

Diana Maynard and Sophia Ananiadou. Identifying terms by their family and friends. In *Proceedings of the 18th International Conference on of Computational Linguistics (COLING)*, Saarbrücken, Germany, 2000.

Diana Maynard, Valentin Tablan, Hamish Cunningham, Cristian Ursu, Horacio Saggion, Katina Bontcheva, and Yorick Wilks. Architectural elements of language engineering robustness. *Journal of Natural Language Engineering - Special Issue on Robust Methods in Analysis of Natural Language Data*, 8(3), 2002.

Diana Maynard, Yaoyang Li, and Wim Peters. Nlp techniques for term extraction and ontology population. In Paul Buitelaar and Philipp Cimiano, editors, *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 171–199. IOS Press, Amsterdam, 2008.

Luke K. McDowell and Michael Cafarella. Ontology-driven information extraction with ontosyphon. In *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, volume 4273 of LNCS, pages 428–444, Athens, Greece, 2006. Springer.

Elena Montiel-Ponsoda. *Multilingualism in Ontologies.* PhD thesis, Universidad Politécnica de Madrid, 2011.

Wolfgang Motsch. *Deutsche Wortbildung in Grundzügen.* de Gruyter, Berlin, 2 edition, 2006.

MUSING-Annual Public Report. MUSING-Annual Public Report. Technical report, MUSING Consortium, 2009.

Daniele Nardi and Ronald J. Brachman. An introduction to description logics. In Franz Baader, Diego Clavanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors, *Description Logic Handbook: Theory, Impementation and Applications.* Cambridge University Press,, Cambridge, UK; New York, NY, USA, 2003.

Roberto Navigli and Paola Velardi. From glossaries to ontologies: Extracting semantic structure from textual definitions. In Paul Buitelaar and Philipp Cimiano, editors, *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 71–91. IOS Press, Amsterdam, 2008.

Patrick Pantel and Marco Pennacchiotti. Automatically harvesting and ontologizing semantic relations. In Paul Buitelaar and Philipp Cimiano, editors, *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 171–195. IOS Press, Amsterdam, 2008.

Dominique Petitpierre and Graham Rusell. Mmorph - the multext morphology program. multext deliverable report for the task 2.3.1, issco. Technical report, University of Geneva, Switzerland, 1995.

Valentina Presutti, Aldo Gangemi, Stefano David, Guadalupe Aguado de Cea, Mari Carmen Suárez-Figueroa, Elena Montiel-Ponsoda, and María Poveda. A library of ontology design patterns: reusable solutions for collaborative design of networked ontologies. Technical report, NeON Project, 2008.

Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the ACL Conference*, Philadelphia, PA, USA, 2002.

Thomas C. Rindflesch and Alan R. Aronson. Semantic processing for enhanced access to biomedical text. In *Real World Semantic Web Applications.* IOS Press, 2002.

Angus Roberts, Robert Gaizauskas, Mark Hepple, Neil Davis, George Demetriou, Yikun Guo, Jay (Subbarao) Kola, Ian Roberts, Andrea Setzer, Archana Tapuria, and Bill Wheeldin. The clef corpus: Semantic annotation of clinical text. *American Medical Informatics Association*, pages 625–629, 2007.

Vitor J. Rocio, Gabriel P. Lopes, and Éric de la Clergerie. Tabulation for multi-purpose partial parsing. *Journal of Grammars*, 4(1), 2001.

Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, 1994.

Helmut Schmid. Lopar: Design and implementation. Technical report, IMS Stuttgart, 2000.

Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26, 1991.

Helmut Schumacher. *Verben in Feldern.* de Gruyter, Berlin, 1 edition, 1986.

Wojciech Skut and Thorsten Brants. Chunk tagger - statistical recognition of noun phrases. In *Proceedings of the ESSLLI Workshop on Automated Acquisition of Syntax and Parsing*, 1998.

Steffen Staab and Rudi Studer. *Handbook on Ontologies. International Handbooks on Information Systems.* Springer, 2004.

Achim Stein. *Einführung in die französische Sprachwissenschaft.* J.B. Metzler Verlag, 2005.

Rudi Studer, V. Richard Benjamins, and Dieter Fensel. Knowledge engineering: Principles and methods. *Data Knowledge Engineering*, 25, 1998.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A large ontology from wikipedia and wordnet. *Elsevier Journal of Web Semantics*, 2008.

Hristo Tanev and Bernando Magnini. Weakly supervised approaches for ontology population. In Paul Buitelaar and Philipp Cimiano, editors, *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 129–143. IOS Press, Amsterdam, 2008.

Johannes Thiele. *Wortbildung der französischen Gegenwartsprache.* Langenschiedt Verlag, 1993.

Jin-Dong Kim Tomoko Ohta, Yuka Tateisi. The genia corpus: An annotated research abstract corpus in molecular biology domain. In *Proceedings of the second international conference on Human Language Technology Research*, pages 82–86, 2002.

Gisela Zifonun, Ludger Hoffmann, and Bruno Strecke. *Grammatik der deutschen Sprache*, volume 3. de Gruyter, Berlin/New York, 1997.