Ontology learning from text based on multi-word term concepts: The *OntoGain* method

Efthymios G. Drymonas

A Thesis presented for the degree of
Master of Science

Intelligent Systems Laboratory
Department of Electronics & Computer Engineering
Technical University of Crete
Greece

June, 2009

Abstract

We propose *OntoGain*, a method for ontology learning from multi-word concept terms extracted from plain text. *OntoGain* follows an ontology learning process defined by distinct processing layers. Building upon plain term extraction a concept hierarchy is formed by clustering the extracted concepts. The derived term taxonomy is then enriched with non-taxonomic relations. Several different state-of-the-art methods have been examined for implementing each layer. OntoGain is based upon multi-word term concepts, as multi-word or compound terms are vested with more solid and distinctive semantics than plain single word terms. We opted for a hierarchical clustering method and Formal Concept Analysis (FCA) algorithm for building the term taxonomy. Furthermore an association rule algorithm is applied for revealing non-taxonomic relations. A method which tries to carry out the most appropriate generalization level between a relation's concepts is also implemented. To show proof of concept, a system prototype is implemented. The OntoGain allows transformation of the derived ontology into OWL using Jena Semantic Web Framework[1]. OntoGain is applied on two separate data sources (a medical and computer corpus) and its results are compared with similar results obtained by Text2Onto, a state-of-the-art-ontology learning method. The analysis of the results indicates that OntoGain performs better than Text2Onto in terms of precision (extracts more correct concepts) while being more selective (extracts fewer but more reasonable concepts).

[1] http://jena.sourceforge.net/

Acknowledgements

I would like to thank my supervisor, Dr. Eupipides G.M. Petrakis for his invaluable advice and support. Also i would like to thank Dr. Kalliopi Zervanou and Spyros Argyropoulos for their encouragement and their contribution in this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web the content, links, and transactions between people and computers. A Semantic Web, which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The intelligent agents people have touted for ages will finally materialize."

Tim Berners-Lee, 1999 [T. Berners-Lee, 1999] *(Inventor of the World-Wide-Web)*

1.1 Semantic Web

The World Wide Web (WWW) is nowadays unbreakably connected with our daily life. It consists of billions of static and dynamically generated Web pages, changing drastically the availability of electronically accessible information. Various automatic services aim at making access to information even more pervasive. However, this enormous amount of data has made it increasingly difficult to find, present and maintain the information required by a wide variety of users. Humans can read web pages and understand them, but the inherent meaning is not available in a structured form that computers can interprete. This fact makes the access to information

less quick and accurate.

In response to this problem, many new research initiatives have been set up to enrich available information with machine-processable semantics. Tim Berners-Lee, Director of the World Wide Web Consortium, referred to the future of the WWW as the semantic web - an extended web of machine-readable information and automated services [Berners-Lee et al., 2001]. The Semantic Web aims at defining ways to allow web information to be used by computers for interoperability and integration purposes between systems and applications. Consequently, information must be provided in such a way that computers can understand it. To give meaning to Web information, new standards and languages are being investigated and developed. Well-known examples include the Resource Description Framework [RDF, 2002] and the Web Ontology Language [OWL, 2004]. The descriptive information made available by these languages allows for characterizing individually and precisely the type and the relationships between the resources in the Web.

The main international standards organization for the World Wide Web [World Wide Web Consortium (W3C)] spots that Semantic Web technologies can be used in a variety of application areas. For example: in data integration, whereby data in various locations and various formats can be integrated in one, seamless application; in resource discovery and classification to provide better, domain specific search engine capabilities; in cataloging for describing content and content relationships available at a particular web site, page, or digital library; to facilitate knowledge sharing and exchange between intelligent agents; in describing collections of pages that represent a single logical document; for describing intellectual property rights of web pages (e.g. the "Creative Commons"[1]), and in many others.

1.2 Ontologies - A preface

From the ancient times, many cultures were concerned with the question of what is the essence of things. Many different answers to this question were proposed by

[1] http://creativecommons.org/

Greek philosophers, from Parmenides (5th and 4th century bc), to the precursor of ontology, Aristotle. The etymology of the term *ontology* means "talking" about "being". Ontology as a diachronic philosophical concept can be described as the *science of existence* as well as the *study of being*[2].

In modern computer science an ontology is not a science, but "ontologies" represent a formal representation of resources (i.e. *concepts*, different kinds of *relations*) that describe a certain domain. Gruber [Gruber, 1993][3] defines ontology as a "formal, explicit specification of a shared conceptualisation".
More precisely, from the W3C Recommendation [OWL Requirements Document, 2004]:
"*Ontologies are used by people, databases, and applications that need to share domain information (a domain is merely a specific subject area or area of knowledge, such as petroleum, medicine, tool manufacturing, real estate, automobile repair, financial management, etc.). Ontologies include computer-usable definitions of basic concepts in the domain and the relationships among them. They encode knowledge in a domain and also knowledge that spans domains. In this way, they make that knowledge reusable.*"

1.3 Motivation - Objectives

Semantic technologies and knowledge representation in general are gaining more and more importance during the last decade. Ontologies are applied in a variety of applications, including web service discovery [Paolucci et al., 2002], information integration [Alexiev et al., 2005], natural language processing [Nirenburg & Raskin, 2004] and dynamic composition of web services [Sirin et al., 2002].

At the same time, the construction of an ontology is a time and cost consuming task that involves specialists from several fields [Pinto & Martins, 2004]. This high development cost is a major barrier to the effort of building large scale intelligent

[2]http://en.wikipedia.org/wiki/Ontology

[3]Gruber's work was the first to describe the role of ontologies in supporting knowledge sharing activities

systems. The main difficulty lies in the fact that an ontology must have a significant coverage of the domain while, in parallel having a steady backbone with meaningful and consistent generalizations[4]. It is yet more difficult and complicated, due to the fact that usually many different specialists have to co-operate for this task, while they must agree on certain design choices[5]. In addition, it is hard to organise a group of experts for each possible domain.

An approach that could dramatically reduce the tedious work and the huge cost of building an ontology would be the automatic learning of ontologies from text documents. However, large documents are hard to comprehend and process their information. Standard text mining and information retrieval techniques usually rely on word matching and do not take into account the structure of the documents within the corpus. Hence, new methods that try to model a domain[6] are beggining to emerge during the last years. Given a certain critical amount of texts, these methods is expected to provide a reasonable coverage of the domain. A notable bottleneck lies to the fact that the consistency of the constructed domain model cannot be guaranteed. Postprocessing from human experts is considered inevitable.

Here comes the task of automatic ontology learning, which tries to assist an expert in the tedious task of modelling a domain. The aim of this thesis is to investigate methods for automatically learning ontologies from domain-specific text collections. Additionally, we built *OntoGain*, a system capable of automatically learning a domain ontology.

[4]The issue of determining the appropriate level of abstraction for binary relations extracted from a corpus with respect to a given concept hierarchy.

[5]Each designer may understand or consider a specific domain from a different angle of view.

[6]The domain model is created in order to document the key concepts and the domain-vocabulary of the corpus being modeled. The model identifies the relationships among all major entities and also identifies their important attributes.

1.4 Contributions

In this work we present *OntoGain*, a general architecture for automatic ontology acquisition from natural language resources. The input of our system is a plain text document (or a document collection) and the output is an OWL-formatted ontology. OntoGain relies on C/NC-Value [Frantzi, Ananiadou, Mima, 2000]. C/NC-Value is a domain independent method for identifying domain concepts in texts which are naturally represented as compound (mult-word) term expressions (i.e., phrases conveying specialized concepts). We prefer compound terms as they are vested with more solid and distinctive semantics than plain single word terms. The extraction of domain terms plays an important role towards better understanding of the contents of document collections. Hierarchical clustering and Formal cluster Analysis (FCA) are examined for building the concept hierarchy of terms.

Besides desribing their position in a hierarchy taxonomy, we examined methods relating concepts to other concepts (i.e. *non-taxonomic relations*, e.g., functional, part-of relations). We focused at learning binary relations and finding the appropriate labels and relation identifiers, on the basis of the examined corpus. Additionally we tried to discover the proper level of abstraction for the domain and range of each relation with respect to the derived concept hierarchy.

We applied OntoGain on the OhsuMed collection[7]. OhsuMed collection consists of 348,566 references from MEDLINE[8], covering all references from 270 medical journals over a five-year period (1987-1991) [Hersh et al., 1994], [Hersh & Hickam, 1994]. We also applied OntoGain on a specialized text corpus of computer science articles, suggested by Milios [Milios et al., 2003].

The results obtained by OntoGain on two separate data sources (the medical and computer corpus) are compared with similar results obtained by Text2Onto, a state-of-the-art-ontology learning method [Cimiano & Volker, 2005]. Text2Onto provides an extensible set of methods for extracting term concepts, learning class hierarchy, as well as object properties and instantiation.

[7] http://ir.ohsu.edu/ohsumed/ohsumed.html

[8] http://www.nlm.nih.gov/databases/databases_medline.html

The experimental results demonstrated significant performance improvements over methods such as Text2Onto, yielding fewer but more correct (reasonable) terms on both corpora.

The structure in this thesis is as follows:

- Chapter 2. Background knowledge and related research:
  State-of-the-art approaches for creating ontologies from text are reviewed first giving particular emphasis on methods for building the taxonomic and non-taxonomic ontology subtasks.

- Chapter 3. Our proposed method:
  We present *OntoGain*, a complete prototype ontology construction system. It consists of several modules for term extraction, to form the domain taxonomy and finally for the extraction of non-taxonomic relations.

- Chapter 4. Experimental results:
  In this chapter we compare the results of our system prototype with results obtained by Text2Onto along with a critical analysis of the results obtained by each method.

- Chapter 5. Conclusions, Future work:
  Finally we make an overall review of OntoGain. We also discuss on potential improvements and issues for further research.

Chapter 2

Background & Related Work

In this chapter we review existing principles and methodologies that focus on the automatic learning of a domain ontology.

Firstly we discuss about issues related to design and representation of ontologies such as ontology languages and terminology. Then we present the state-of-the-art in automatic ontology learning focusing on the main approaches associated with the main subtasks of ontology language development (i.e., taxonomy generation and mining of non-taxonomic relations) along with tools from text analysis and mining for implementing each task.

2.1 Ontologies

> *"The vision of the Semantic Web is to extend principles of the Web from documents to data; data should be related to one another just as documents (or portions of documents) are already. This also means creation of a common framework that allows data to be shared and reused across application, enterprise, and community boundaries, to be processed automatically by tools as well as manually, including revealing possible new relationships among pieces of data"*
>
> [World Wide Web Consortium (W3C)], [T. Berners-Lee, 1999]

2.1.1 Introduction - Ontology Languages

Towards the creation of a common framework that should allow data to be shared more efficiently between applications (as the W3C quotation denotes above), there were many efforts that consisted of building computable models in an ontology language. That means that a specific domain could now be decribed in terms of structured data, with the use of an ontology language. Examples of these languages are [RDF, 2002] - RDF(S), [OIL, 2000], [DAML+OIL, 2001]. Finally, in 2001, the W3C formed a working group called the Web-Ontology (WebOnt) Working Group[1]. The aim of this group was to devise a new ontology markup language for the Semantic Web, called [OWL, 2004] (Ontology Web Language). This language was proposed as a W3C recommendation in February 2004.

OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary. OWL can be used to represent the meaning of terms and the relationships between those terms. It is more expressive than XML, RDF and RDF-S, making it easier to represent machine interpretable content on the Web. It is a revision of DAML-OIL web ontology language and it has three sublanguages (species):

- **OWL Lite** that supports a classification hierarchy and simple constraints.
- **OWL DL** that supports maximum expressiveness while retaining computational completeness and decidability.
- **OWL Full** that supports maximum expressiveness but no computational guarantees.

OWL DL and OWL Lite semantics are based on Description Logics [Horrocks & Peter Patel-Schneider, 2004], which have attractive and well-understood computational properties, while OWL Full uses a semantic model intended to provide compatibility with RDF Schema.

[1] http://www.w3.org/2001/sw/WebOnt/

2.1.2 Main Components of an Ontology

Although different representation formalisms exist for the implementation of ontologies, they usually share the following minimal set of components, starting with classes.[2].

Classes describe the domain concepts. They form the backbone of the ontology (the taxonomy), through which inheritance mechanisms can be applied. The most basic concepts in a domain should correspond to classes that are the roots of various taxonomic trees. Additionally information describing the relations of different concepts that model the domain, are attached upon these classes.

Relations express association between domain concepts and are usually binary. The first argument is known as the domain of the relation while the second argument is referred to as its range. For example, the binary relation *Subclass-Of* is used for building the class taxonomy: `headache` is a subclass of `ache`, while `ache` is a subclass of `symptom`. In the relation *Subclass-Of(headache, ache)*, the domain is *headache* while *ache* constitutes the range of the relation respectively. Besides taxonomic, there are also non-taxonomic relations which are crucial for understanding other concept relationships (e.g. part-of, functional). *Relation learning* is the task of learning relation labels and/or identifiers as well as their appropriate domain and range.

In ontology engineering are often used more complex relation types between concepts or attributes that are called *axioms*. Such axioms may describe the properties of a relation, such as *transitivity* or *symmetry*. For concepts correspondingly we may have *disjointness* or *equivalence*[3]. Thus we may want to learn which pairs concepts are disjoint or which relations are symmetric. In the context of this thesis we restrict ourselves to binary relations.

Several techniques for learning (extracting) both taxonomic and non taxonomic

[2]According to each formalism, there are various component names. For example, classes are also known as concepts, entities and sets; relations are also known as roles, properties and slots; instances as individuals etc.

[3]For example the concepts *mountain* and *sea* are *disjoint*

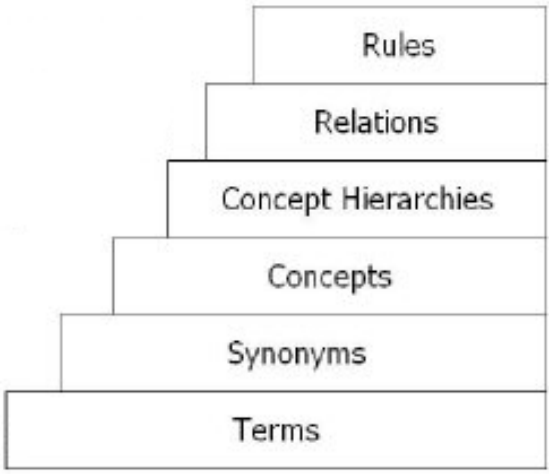


Figure 2.1: Ontology learning layers

relations are reviewed next.

2.2 Subtasks of Ontology Development

In the previous subsection we presented the main components that form an ontology. In what follows, we discuss about the subtasks that constitute the complex work of ontology construction.

Buitelaar [Buitelaar et al., 2005] and Cimiano [Cimiano et al., 2006] propose that all of the aspects of ontology development can be organized into a layer stack with the more complex tasks being at the top, as illustrated in Figure 2.1. The higher layers rely on the output of the lower layers. For example, we cannot discover relations between concepts before potential concepts are available from the extraction phase.

According to this layer structure, term extraction is the basic prerequisite for ontology learning from text. The main task in this phase lies in finding a set of relevant terms which are characteristic for the domain and will form the ontology lexicon. From our point of view, terms are multi-word compounds. Multi-word terms are not only vested with more compact and distinctive semantics (e.g. the term "carotid artery disease" distinguishes the document from any other document referring to other carotid artery information or diseases) but, they also present the advantage of lexically revealing their semantic content classificatory information,

by means of modifiers [Bourigault et al., 1996]. For example, the compound term "carotid artery disease" denotes a type of "artery disease", which in turn is a type of "disease". In this example, the single-word term "disease" has no apparent indication of its respective category, whereas for the multi-word terms, their modifiers, "carotid artery" and "artery", provide an indication of their respective reference to a specific disease type. Therefore, we try to retain more detailed and meaningful document semantics. That is also why our system is capable of processing relatively quickly and accurately big document collections contrary to other systems examined that exhausted the system resources while they worked out only with a few hundreds of plain text lines. It is important also to mention that in OntoGain terms are directly identified as concepts; this means that we do not take into consideration that terms can be polysemous.

The concept hierarchy layer constitutes the "backbone" of the ontology. A concept hierachy is a collection of the extracted concepts organized into a hierarchical structure. Each concept in a taxonomy belongs to one or more parent/child (broader/narrower) taxonomic relationships to other concepts in the taxonomy.

Concepts are also characterized by attributes as well as by relations to other concepts in the hierarchy (as discussed in the previous subsection). This is the relation layer where *non-taxonomic* relations are defined. These relationships are typically expressed by a verb relating pair of concepts [Kavalec et al, 2004].

In the last layer (top) layer, axioms or rules are defined. According to [Gruber, 1993], formal axioms serve to model sentences that are always true. They are normally used to represent knowledge that cannot be formally defined by the other components. In addition, formal axioms are used to verify the consistency of the ontology itself. The learning of axioms and rules at the top level is out of the scope of this work.

2.3 Natural Language (Pre)Processing

In order to be ready to be processed, the text collection must be relied on several preprocessing steps. A typical corpus preprocessing procedure consists of the

following substeps:

- Sentence Splitting: The initial step where sentences are recognized into the text.
- Tokenization: The process of breaking a text up into its constituent tokens. Tokenization can occur at different levels: a text could be broken up into paragraphs, sentences, words.
- Part-of-Speech (POS) taggging: It is the task of assigning to each token its corresponding syntactic word category (Part-of-speech, i.e. *noun*, *verb*, *adjective* etc).
- Lemmatization / Morphological Analysis: This is a normalization step, used to map morphological variants to their corresponding base form. For example the word "mice" becomes "mouse" or the word "travelling" becomes "travel".
- Shallow Parsing: This relates to extracting contextual features from text in order to extract syntactic dependencies. These dependencies are not obtained from parse trees[4] like full parsers, but match certain regular expressions over part-of-speech tags. The shallow parsing is easier to implement and more efficient than using a full parsing approach [Cimiano & Volker, 2005]. Cimiano and Volker used such an approach to extract contextual features, which they called "pseudo-syntactic dependencies" in their named entity classification task. They showed that the pseudo-syntactic dependencies results were very good in contrast to the results of a full parser in their experiments on a tourism domain [Cimiano et al., 2005]. In a work that automatically clusters adjectives, Hatzivassiloglou also reported that syntactic features from shallow parsing outperformed window-based co-occurrences [Hatzivassiloglou, 1996].

[4]Parse tree is an (ordered, rooted) tree that represents the syntactic structure of a string according to some formal grammar.

2.4 Concept Extraction

In this subsection we try to make an overview of the methods for both extracting concept terms and for building the ontology lexicon.

A purely linguistically based tool to term extraction, created by the French Electricity Board for thesauri updating and creation, is Lexter [Bourigault et al., 1996]. Lexter works in two stages. During the first stage, it extracts all word sequences that on linguistic rules could not constitute a terminological unit. Studies of the linguistic properties of terms have shown that certain word sequences rarely constitute a term, for example sequences comprising conjugated verbs, pronouns, conjunctions or certain strings of prepositions + determiners. Next, based again on linguistic information on the prevailing term formation patterns of the special language under study, Lexter extracts subsets from maximal length noun phrases that most likely constitute a terminological unit. The resulting list is then submitted to an expert for validation. Another approach which makes use both of word repetition and negative knowledge is proposed by Oueslati et al. (1996). Their algorithm extracts repeated word sequences and then uses a stop list and a domain verb list to filter the extracted data. The result is a list of noun-noun combinations. Additionally, FASTR, an algorithm proposed by [Jacquemin, 2001], attempts to retrieve both terms and term variants in an attempt to improve recall (i.e. to reveal even more terms).

There are also approaches applying dimension reduction techniques. Latent Semantic Analysis (LSA)[5] is a method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text [Landauer & Dumais, 1997], [Landauer et al., 1998]. LSA works by analyzing the relationships between terms in documents. It is based on the idea of producing a thesaurus of co-occuring terms. Starting from a term co-cocurrence matrix, terms that often occur together are grouped into concepts so that every time a user asks for a term, the system determines the relevant concepts. Thus the terms and

[5]In the context of its application to information retrieval, it is sometimes called Latent Semantic Indexing (LSI).

documents are now indirectly related through the concepts.

A method for extracting multi-word terms is [KEA, 1999]. KEA automatically extracts keyphrases from the full text of documents. The set of all candidate phrases in a document are identified using rudimentary lexical processing, features are computed for each candidate, and machine learning is used to generate a classifier that determines which candidates should be assigned as keyphrases. Two features are used in the standard algorithm: $tf \cdot idf$ and position of first occurrence. The $tf \cdot idf$ requires a corpus of text from which document frequencies can be calculated; the machine learning phase requires a set of training documents with keyphrases assigned.

Another method for the automatic extraction of multi-word terms is C/NC-Value [Frantzi, Ananiadou, Mima, 2000]. C-NC/Value constitutes a domain-independent method, combining linguistic and statistical information. It enhances the common statistical measure of frequency of occurrence and incorporates information from context words to the extraction of terms.

Milios et al. [Milios et al., 2003] demonstrated that C/NC-Value outperforms KEA. C/NC-Value achieves significantly better precision and recall in identifying terms upon special text corpora. Based on this result, the extraction phase of *OntoGain* is implemented by *C/NC-Value*.

2.4.1 The C/NC-Value method

The C-Value method is a hybrid domain-independent method combining linguistic and statistical information (with emphasis on the statistical part) for the extraction of multi-word and nested terms (i.e. terms that appear within other longer terms, and may or may not appear by themselves in the corpus). This method takes as input a corpus and produces a list of candidate multi-word terms, ordered by the likelihood of being valid terms, namely their C-Value measure.

The C/NC-Value method comprises of a linguistic part (for extracting candidate terms) and of the statistical part (computing relative importance values to these terms). More specifically, term importance is a function of two values, namely C-

Value and NC-value.

The Linguistic Part

Terms consist mostly of nouns and adjectives [Sager, 1990] and sometimes prepositions [Justeson, Katz, 1995]. The statistical information , without any linguistic filtering, is not enough to produce useful results. Without any linguistic information, undesirable strings such as of the, is a, etc., would also be extracted. The linguistic filter is used to extract noun phrases that constitute multi-word terms discarding such undesirable strings. Part-of-Speech(POS) is applied prior to linguistic filters.

Additionally, C/NC-Value uses a stop list for discarding terms that are not expected to occur as concept words in that domain, improving the precision of the output.

The Statistical Part

The C-value constitutes a measure of the importance of each candidate term extracted in the previous steps. The higher the C-Value measure the more likely it is the candidate term to be a valid term of the corpus. The C-Value of a multi-word term is computed as follows:

$$f(\alpha) = \begin{cases} log_2|\alpha| * f(\alpha) & \text{if } \alpha \text{ is not nested} \\ log_2|\alpha| * (f(\alpha) - \frac{1}{P(T_\alpha)} \sum_{b \in T_\alpha} f(b)) & \text{otherwise} \end{cases} \tag{2.1}$$

The negative effect on the candidate string being a substring of other longer candidate terms is reflected by the negative sign '-' in the formula above. The independence of α from these longer candidate terms is given by $P(T_\alpha)$. The greater this number the bigger its independence (and the opposite) is reflected by having $P(T_\alpha)$ as the denominator of a negatively signed fraction. The measure is built using several statistical characteristics of the candidate string. These are:

1. The total frequency of occurrence of the candidate string in the corpus.

2. The frequency of the candidate string as part of other longer candidate terms.

3. The number of these longer candidate terms.

4. The length of the candidate string (in number of words).

The higher the number of distinct longer terms that our string appears as nested in, the more certain we can be about its independence (i.e. that the candidate term extracted is a real term). The fact that a longer string appears X times is more important than that of a shorter string appearing X times.

NC-Value is an enhancement to C-Value that is computed based on context information. NC-Value creates a list of important term context words. Term context words are words that appear in the vicinity of terms in texts. These will be ranked according to their "importance" when appearing with terms. The criterion for the extraction of a word as a term context word is the number of terms it appears with. The higher this number is, the higher the likelihood that the word is "related" to terms (it occurs with other terms in the same corpus). Hence, with help from terms' NC-Value ranks again the list of candidate concepts, trying to bring higher terms that is more likely to consist valid concepts for the domain. Each candidate term in the C-Value list appears in the corpus with a set of context words. From these context words, the nouns, adjectives and verbs are retained for each candidate term. NC-Value provides a method for the extraction of term context words (words that tend to appear with terms) and incorporates this information (from term context words) into the term extraction process. This above criterion is more formally expressed as :

$$weight(w) = \frac{t(w)}{n} \tag{2.2}$$

where:

- w is the context word (noun, verb or adjective) to be assigned a weight as a term context word
- $weight(w)$ is the assigned weight to the word w
- $t(w)$ is the number of terms the word w appears with
- n is the total number of terms considered

The purpose of the denominator n is to express this weight as a probability (the probability that the word w might be a term context word). The NC-value measure is then computed as :

$$NC - Value = 0.8C - Value(a) + 0.2 \sum_{b \in C_\alpha} f_a(b) weight(b) \tag{2.3}$$

where:

- a the candidate term
- C_α the set of distinct context words of a
- b is a word from C_α
- $f_a(b)$ the frequency of b as a term context word of a
- $weight(b)$ is the weight of b as a term context word

The two factors of NC-value, i.e. C-value and the context information factor, have been assigned the weights 0.8 and 0.2 respectively. These have been chosen among others after experiments and comparisons of the results [Frantzi, Ananiadou, Mima, 2000].

2.5 Concept Hierarchies

In this section we review common approaches for the extraction of concept hierarchies. There are methods using co-occurrence analysis and lexico-syntactic patterns ([Hearst, 1992], [Iwanska et al., 2000], [Cederberg & Widdows, 2003]), as well as methods based on concept similarity and clustering ([Hindle, 1990], [Caraballo, 1999], [Faure & Nedellec, 1999], [Bisson et al., 2000]). There is no single correct class hierarchy for any given domain. The hierarchy depends on the possible uses of the ontology, the level of the detail that is necessary for the application, the personal preferences, while sometimes depends on requirements for compatibility with other models [Noy&McGuinness, Stanford].

2.5.1 Lexico-Syntactic Patterns

An early approach for the induction of concept hierarchies from textual data relies on the application of lexico-syntactic patterns [Hearst, 1992]. Hearst defined patterns that exploit hyponymy relations as in the following example:

$$\text{such } NP_0 \text{ as } \{NP_1,\}^* \{(\text{and} \mid \text{or})\} \ NP_2$$

where NP stands for a noun phrase. This denotes that if such a pattern is matched in a text, then we can deduce that NP_0 is a hypernym of NP_1 and NP_2.
More specifically, Hearst proposed the following patterns:

1. NP such as {NP,}* {(and | or)} NP
2. such NP as {NP,}* {(and | or)} NP
3. NP {,NP}* {,} or other NP
4. NP {,NP}* {,} and other NP
5. NP including {NP,}* NP {(and | or)} NP
6. NP especially {NP,}* {(and—or)} NP

Lexico-syntactic patterns such as the above can be easily identified and extracted from texts. However, lexico-syntactic patterns occur rarely in corpora, thus large amounts of texts are needed in order to find a reasonable amount of patterns capable of forming is-a relations.

Many researchers improved Hearst's initial work. For example Cederberg and Widdows [Cederberg & Widdows, 2003] indicated that the precision of Hearst patterns can be improved by filtering the results via Latent Semantic Analysis. Iwanska [Iwanska et al., 2000] also defined additional patterns, while other researchers proposed automatic learning of the patterns ([Downey et al., 2004], [Ravichandran & Hovy, 2002]).

2.5.2 Formal Concept Analysis

Formal Concept Analysis (FCA) [Ganter et al., 1999] forms a very interesting approach for building concept hierarchies [Haav, 2003], [Cimiano et al., 2005], [Jian Wang & Keqing He, 2006]. FCA relies on the idea that the objects are connected with their characteristics. FCA takes as input a matrix specifying a set of objects and attributes. Then it finds all the "natural" clusters of attributes and all the "natural" clusters of objects in the input data. By saying "natural" object cluster we mean the set of all objects that share a common subset of attributes, while a "natural" property cluster is the set of all attributes shared by one of the natural object clusters. In the scope of FCA, objects are called by *formal objects*, while their characteristics are named *formal attributes*. For the purpose of our work, the object-verb pairs extracted by the shallow parsing process are correlated to the formal objects and their respective formal attributes.

Which attributes hold for each object is depicted by a binary relation called the *incidence relation*. Objects, attributes as well as incidence relations constitute a *formal context*, as it will described in an example below (Table 2.1, Table 2.2). FCA can be seen as an unsupervised conceptual clustering technique and more broadly, a method of data analysis. For a more detailed description of the FCA method the reader is referred to [Ganter et al., 1999], [Ganter et al., 1999].

Table 2.1: Example concepts and verbs extracted from a corpus

Verb	Concepts
book	hotel, apartment, car, bike, excursion, trip
rent	apartment, car, bike
drive	car, bike
ride	bike
join	excursion, trip

The notion of a *formal context* is of key importance:

Definition 2.5.1 `[Formal Context]` *A formal context is a triple K := (G,M,I), if G and M are sets and $I \subseteq G \times M$ is a binary relation between G and M. The elements of G and M are called objects and attributes respectively. I is the incidence relation of the context.*

For a set $A \subseteq G$ of objects, we define

$$A' = \{m \in M | \forall g \in A : (g, m) \in I\} \tag{2.1}$$

We define for a set $B \in M$ of attributes respectively:

$$B' = \{g \in G | \forall m \in B : (g, m) \in I\} \tag{2.2}$$

Thereafter we define a *formal concept* as:

Definition 2.5.2 `[Formal Concept]` *A pair (A,B) is a formal concept C of (G,M,I) if and only if $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$. The set A is called the extent and the set B the intent of the concept C.*

In other words, a set of objects A and a set of attributes B constitute a *formal concept* (A, B) if the attributes in B are exactly those that are common to all objects in A and, conversely, the objects in A are exactly those that have all the attributes in B. Formal concepts of a given context are ordered by the subconcept-superconcept relation as follows:

$$(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 (\Leftrightarrow B_2 \subseteq B_1) \tag{2.3}$$

We will use an example from [Cimiano et al., 2005] to illustrate the FCA definitions. In a tourism domain, let us suppose that the concepts along with the corresponding verbs have been identified, as depicted in table 2.1. Table 2.2 illustrates its corresponding formal context which is then transformed into a "context lattice"[6].

[6] *From Wikipedia*: A lattice is a partially ordered set in which subsets of any two elements have a unique supremum (the elements' least upper bound; called their join) and an infimum (greatest lower bound; called their meet)

Lets consider the set of objects {excursion, trip} as an example. Both share the attributes *bookable* and *joinable*. However, the only objects (concepts) that have both these attributes in common are *excursion* and *trip*. Thus ({*excursion, trip*}, {*bookable, joinable*}) are formal concepts. Additional formal concepts are ({*car, bike*}, {*rentable, driveable, bookable*}) and its sub-concept ({*bike*}, {*rideable, rentable, driveable, bookable*}).

Table 2.2: Tourism domain knowledge as a formal context

	bookable	rentable	driveable	rideable	joinable
hotel	⋆				
apartment	⋆	⋆			
car	⋆	⋆	⋆		
bike	⋆	⋆	⋆	⋆	
excursion	⋆				⋆
trip	⋆				⋆

A concept lattice is represented by a line diagram, consisting of nodes and links. Each node represents a formal concept. Each link connecting two nodes represents the subconcept-superconcept relation between them. The diagram demonstrates that the attributes are inherited by the intents of all nodes downwards. Similarly, objects are also carried in the extent of all formal concepts upwards (see Definition 2.5.2).

Notice that there is redundant information in Figure 2.2. The lattice in Figure 2.3 (left) emerges from figure 2.2 using "reduced labeling" [Ganter et al., 1999]. For example, for a pair (A, B), B (the intent, as defined above) will appear in every descendant. The inherited elements may be eliminated. In our example, bike keeps only *ridable*, as the others (*bookable, drivable, rentable*) appear already in its ancestors.

Finally, in Figure 2.3 we demonstrate the lattice of formal concepts. The corresponding concept hierarchy for our example is shown on the right. Details on our

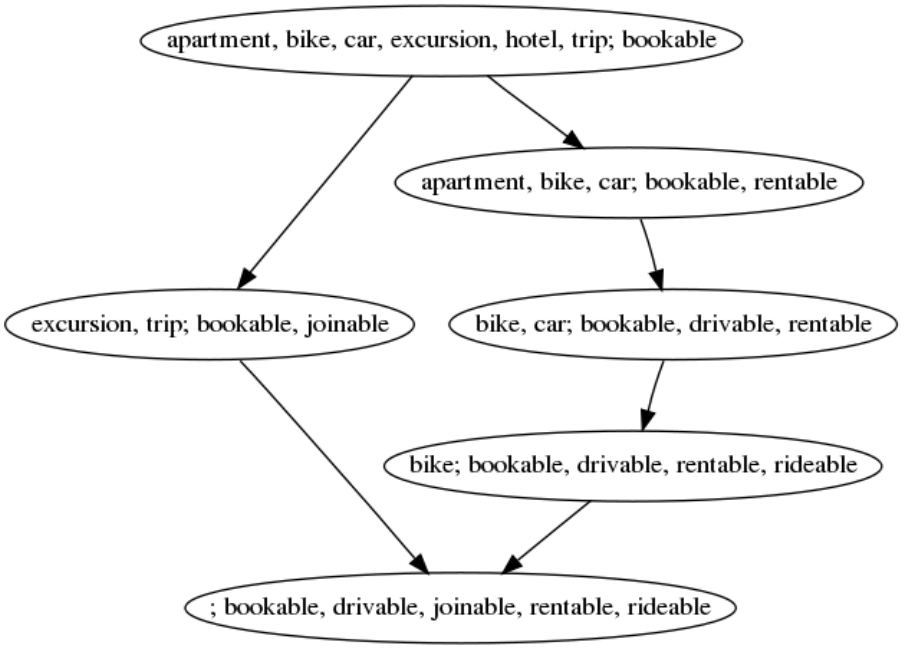


Figure 2.2: Lattice of formal concepts

implementation of the FCA method are provided in Chapter 3.

2.5.3 Clustering & other approaches

The induction of concept hierarchies is usually based on clustering. The main idea here is that similar words tend to occur in similar linguistic contexts [Harris, 1968]. Next, we discuss approaches based on clustering as well as approaches relying on linguistic criteria.

Caraballo [Caraballo, 1999] suggests a bottom-up clustering method for building a concept hierarchy from nouns extracted from the Wall Street Journal Corpus. These nouns are single-word terms, as she only considers the lemmatized head of noun phrases. Similarity between clusters of nouns is computed by a a combination of cosine and average linkage measures. Hearst patterns are applied for extracting hypernym relations and for labeling the tree nodes of the derived hierarchy.

Cimiano [Cimiano et al., 2005], the ASIUM system [Faure & Nedellec, 1999] and Mo'K Workbench [Bisson et al., 2000] use agglomerative clustering to induce a concept hierarchy.

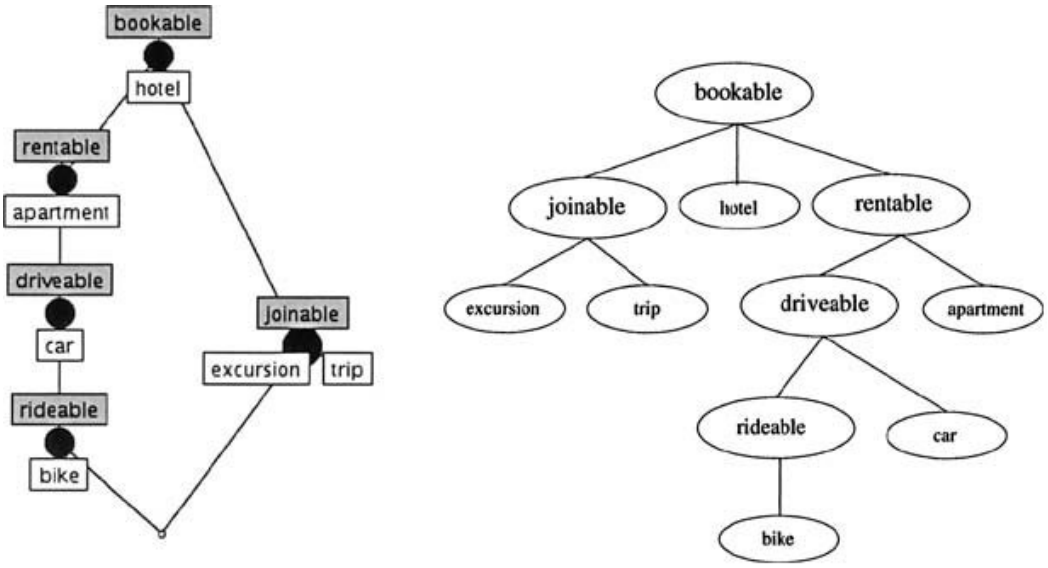


Figure 2.3: Lattice of formal concepts compacted - Reduced Labeling

All approaches based on clustering more or less share the problem of not being able to label the produced clusters directly. In addition, they often suffer from data sparseness, as many syntactic relations are spurious, leading to wrong data results.

There are also approaches that exploit linguistic criteria for deriving taxonomic relations. The OntoLT [Buitelaar et al., 2004] performs linguistic annotation and chunking using a shallow parser. The annotations are then mapped to ontological structures. In addition OntoLearn [Velardi et al., 2002] relies on WordNet to form a taxonomy. WordNet provides different synonym sets for terms in many cases, leading to different hyponym or hypernym trees. Without help from an expert it is difficult to choose which path is correct. For this purpose they propose an algorithm for sense disambiguation in order to define patterns that represent correct paths in WordNet. Recently, Sanchez & Moreno [Sanchez & Moreno, 2004], [Sanchez & Moreno, 2005] proposed an approach for inducing concept hierarchies in the WWW given a certain seed word. Also Sabou [Sabou, 2005] suggests extraction of taxonomic relations for the modelling of web services.

2.6 Learning of Relations

In the following we review techniques addressing the problem of extracting *non-taxonomic* relations. In contrast to *taxonomic* (is-a) which establish abstraction hierarchies, these are relationships which express that one concept is logically related to another. In addition, unlike taxonomic relations for which it is often possible to take advantage of external linguistic resources or the internal structure of terms to uncover hierarchical structures, non-taxonomic relationships tend to be domain-dependent: They can only be understood in the context of a specific domain application [Gulla & Brasethvik, 2008].

2.6.1 Association Rules

Association rules are commonly used to discover data, text elements or patterns that co-occur frequently within a dataset. Such patterns can be used to make predictions on data. They were first introduced by Agrawal [Agrawal et al., 1993] as a technique for market basket analysis. The aim here was to find association rules that predict the purchasing behavior of customers. The following statement is an example of such an association rule: *90% of the transactions that purchased bread and butter also purchased milk.*
Mining of association rules can be formally expressed as follows:

Definition 2.6.1 `[Association Rules]` Let I be a set of literals, called items. Let D be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. A transaction T contains X, a set of some items in I, if $X \subseteq T$.
An association rule is an implication of the form:
$X \Rightarrow Y$, where $X \subset I$,$Y \subset I$, $X \cap Y = \oslash$.
A rule $X \Rightarrow Y$ holds in the transaction set D with *confidence c* if c% of the transactions in D that contain X also contain Y. The rule $X \Rightarrow Y$ has *support s* in the transaction set D if s% of the transactions in D contain $X \cup Y$.

More precisely, the confidence of an association rule is a percentage value that shows how frequently the rule head (i.e. X) occurs among all the groups containing

the rule body (i.e. $X \Rightarrow Y$). The confidence value indicates how reliable this rule is. The higher the value, the more often these items are associated together. The support of an association rule is the percentage of groups that contain all of the items listed in that association rule. The percentage value shows how often the joined rule body and rule head occur among all the groups that were considered.

In general, only rules that achieve a certain minimum level of confidence and support are included in the mining model. This ensures a definitive result and it is one of the ways in which the created rules can be evaluated.

In ontology learning, transactions are defined in terms of words occurring together in certain syntactic dependencies. If the rule $X \Rightarrow Y$ has been generated and stored, we can conclude that there is a relationship between the concepts in X and the concepts in Y. [Gulla & Brasethvik, 2008], [Madche & Staab, 2000].

2.6.2 Linguistic Criteria

In this approach the task of learning relations from corpora is based on verbal expressions. The main idea lies on the extraction of verb frames. The verbal frames indicate that different concepts are connected with some relation ([Ciaramita et al., 2005], [Gamallo et al., 2002], [Buitelaar et al., 2004], [Schutz & Buitelaar, 2005]).

Cimiano [Cimiano et al., 2006] focuses also on the appropriate generalization of the arguments of a relation with respect to a given taxonomy. For instance speaking for the relation *suffer_from*, the pairs *suffer_from(older man, head ache)*, *suffer_from(ill person, head ache)*, *suffer_from(woman, stomach ache)* are certainly valid. However, from an ontology point of view we are interested in finding the most general relation that describes all the relation instances with respect to the concepts described on our given taxonomy, to avoid the necessity of representing each case explicitly. In our case this generalization shall be *suffer_from(patient, ache)*. The issue of determining a suitable level of abstraction between concepts in a hierarchy has been explored also by Clark & Weir [Clark & Weir, 2002].

2.6.3 Other methods

Yamaguchi [Yamaguchi, 2001] uses Schutze's [Schutze, 1993] word space method for discovering similar terms and suggests potential relations to the user. Claveau et al. [Claveau et al., 2003] apply Inductive Logic Programming (ILP) [Lavrac & Dzeroski, 1994] for discovering verbs which are "qualia" [7] elements.

Tegos et al. [Tegos et al, 2008] present an approach for ontology learning from texts which are semantically annotated with instances of ontologies' concepts. Statistical techniques are applied to metadata extracted from the annotated texts to discover semantic relations among the annotated concepts as well as to find cardinality restrictions to these concepts and their relations.

2.7 Corpora - Gold Standard Ontologies

In order to benchmark our method against other methods, we tried to find a standard ontology with respect to a corresponding dataset. The *Genia corpus* and its corresponding Genia ontology are commonly used as a reference[8]. The Genia ontology, developed by Tsuji Labs[9], comprises of a taxonomy developed from the semantic classification used in the GENIA corpus[10]. In OntoGain, unlike Genia, the input is non-annotated text. In addition, the Genia ontology contains mostly application specific terms (from medical and genomic texts) making it difficult to work with it and compare the results obtained by different methods, as this would require relevance judgement from domain experts. For the performance comparison experiments in this work we decided to work with a medical (OhsuMed) and a computer science corpus. Other golden standard taxonomies that could be used to compare with our results, are provided in the work by Madche and Staab [Mad-

[7] The qualia structure [Pustejovsky, 1995], gives access to relational information that is crucial both for NLP applications and for linguistic analysis. In particular, the qualia roles express the basic features (telic, agentive, constitutive, formal) of the semantics of nouns.

[8] http://www-tsujii.is.s.u-tokyo.ac.jp/ genia/topics/Corpus/

[9] http://sys.pwr.eng.osaka-u.ac.jp/home.html

[10] as shown in: http://www-tsujii.is.s.u-tokyo.ac.jp/ genia/topics/Corpus/genia-ontology.html

che & Staab, 2002]. In the context of their study, they asked ontology engineers to model an ontology from a tourism dataset[11]. The corpus texts are written in German, comprising a barrier to our efforts. We were seeking for a corpus with a corresponding extracted ontology.

OhsuMed collection contains a total of 348,566 references from Medline, the online medical information database. It consists of titles and/or abstracts from 270 medical journals over a five-year period (1987-1991). The OhsuMed document collection was compiled by William Hersh and colleagues, for the experiments described in [Hersh et al., 1994], [Hersh & Hickam, 1994]. OhsuMed was a first corpus used to benchmark our method.

We also applied our system on a specialized text corpus of computer science articles, used in the work of Milios et al. [Milios et al., 2003]. We used this corpus because the results were more easily evaluated by colleague domain experts.

2.8 Ontology Construction Methodologies

OntoLearn [Velardi et al., 2002] aims at extracting relevant domain terms from a corpus of text, relating them to appropriate concepts, while in addition detects relations among the concepts. OntoLearn uses natural language, statistical techniques and WordNet [Miller, 1995] lexical knowledge base. Concepts are related according to taxonomic and other semantic relations, using a rule-based inductive learning method.

ASIUM [Faure & Nedellec, 1999], [Faure et al., 2000] implements a bottom-up and breadth-first clustering strategy. At the same time, it uses also subcategorization frames with generalized selectional restrictions to form the concept hierarchy. ASIUM takes as input French texts and assigns a frequency of occurrence to each word in the text. The learning method is based on conceptual and hierarchical clustering.

Text2Onto [Cimiano & Volker, 2005] is an ontology learning framework which

[11] http://www.aifb.uni-karlsruhe.de/WBS/pci/TourismGoldStandard.isa

has been developed to support the acquisition of ontologies from textual documents. Like its predecessor TextToOnto [Madche & Staab, 2000], it provides an extensible set of methods for learning atomic classes, class subsumption, as well as object properties and instantiation. It includes term extraction, concept association extraction and ontology pruning algorithms. The rules extraction part consists of two distinct algorithms that extract potential taxonomic and non-taxonomic relationships between existing ontology concepts. The association rules extraction algorithm takes the proximity of two terms in the text to denote the correlation between concepts between these terms. The linguistic patterns algorithm analyses the text for common patterns that signal relationships between concepts. Text2Onto further relies on more involved user interation.

OntoLT [Buitelaar et al., 2004] relies on linguistic knowledge and uses built-in patterns that map possibly complex linguistic structure directly to concepts and relations. The approach provides a plug-in for Protege [Knublauch, 2003], with which concepts (Protege classes) and relations (Protege slots) can be extracted automatically from annotated text collections. For this purpose, the plug-in defines a number of linguistic patterns over an annotation format that will automatically extract class and slot candidates. Alternatively, the user can define additional rules, either manually or by the integration of a machine learning process.

Chapter 3

The OntoGain system - Implementation

OntoGain is complete prototype ontology construction system. Compared to its most successful counterparts in the literature (e.g. Text2Onto), OntoGain exhibits the following two important advantages: (a) produces an ontology of multi-word - rich in semantics - domain concepts rather than an ontology of mere single word terms and (b) produces an ontology in OWL. For this purpose, we used *Jena Semantic Web Framework*[1]. Jena provides a programmatic environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine. With OWL's help the ontology can be visualized and inspected using a common ontology editor such as *Protege*[2].

Figure 3.1 illustrates OntoGain architecture. The system consists of several modules, the most important of them being:

(a) The preprocessing module whose purpose is to prepare the corpus documents for the next modules that will form the domain taxonomy and will discover relations between concepts.

(b) The concept extraction module which identifies multi-word terms corresponding

[1]http://jena.sourceforge.net/

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

to the distinct and important corpus concepts.

(c) The taxonomy construction module which produces the back-bone generalization hierarchy of the concepts derived in the previous step.

(d) The non-taxonomic hierarchy generation module whose purpose is to enrich the taxonomy with domain specific concept relationships.

The modules are implemented in a sequence of steps. Various methods have bee considered as candidates for implementing each step and are described below.

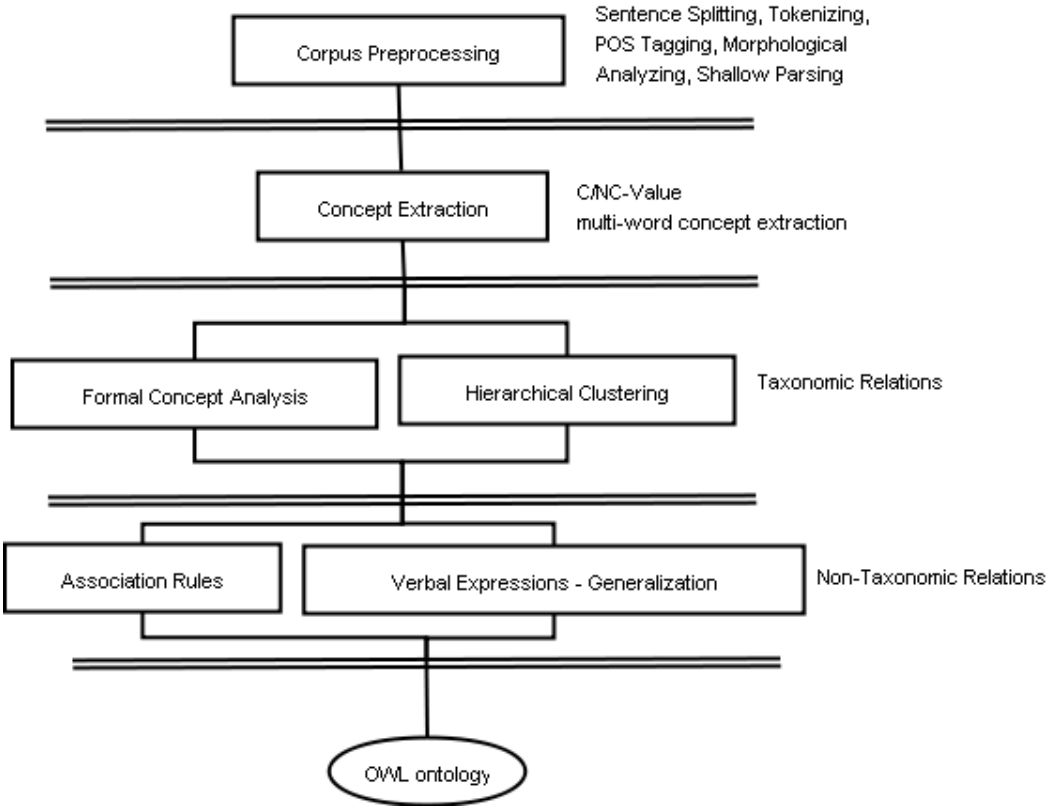


Figure 3.1: OntoGain layers

OntoGain receives a text corpus as input. After the preprocessing step, multi-word term concepts are extracted by C/NC-Value. FCA or Agglomerative clustering may be applied for generating the concept hierarchy. Within OntoGain, both methods are implemented. In turn, extraction of non-taxonomic relations relies on the application of the association rules method or on a method that extracts non-taxonomic relations based on verbal expressions and determines the right level of

abstraction for the domain and range of the relation respectively. The user may choose only one method for the extraction of taxonomic and non-taxonomic relations, as the results from each method may overlap each other.

3.1 Concept Extraction

We applied the C/NC-Value method for the extraction of multi-word concepts as described in the previous chapter.

3.1.1 Preprocessing

We used initially the tools from GATE[3] for Part-of-Speech Tagging , tokenizing and sentence splitting. However, our first experiments showed that the corresponding collection from OpenNLP[4] was faster and more stable. For this reason we decided to use the OpenNLP tools for the preprocessing stage. The OpenNLP POS tagger that we used is the MX-POST, based on the work of Ratnaparkhi on maximum entropy models[5] for natural language processing [Ratnaparkhi, 1996].

Towards the extraction of verbal dependencies (subject - verb - object) we used a chunker - shallow parser from OpenNLP. We tried to apply a full syntactic parser like Stanford parser[6] developed by the Stanford Natural Language Processing Group or the very good Enju[7] parser, developed by Tsujii laboratory in the university of Tokyo. Unfortunately with both of them we experienced problems with the

[3]http://gate.ac.uk :: GATE is an infrastructure for developing and deploying software components that process human language.

[4]http://www.opennlp.org :: OpenNLP is an organizational center for open source projects related to natural language processing. Hosts a variety of java-based NLP tools which perform sentence detection, tokenization, pos-tagging, chunking and parsing, named-entity detection, and coreference using the OpenNLP Maxent machine learning package.

[5] [Manning, Schutze, 1999] (page 589): Maximum entropy modeling is a framework for integrating information from many heterogeneous information sources for classification. The data for a classification problem is described as a (potentially large) number of features.

[6]http://nlp.stanford.edu/software/lex-parser.shtml

[7]http://www-tsujii.is.s.u-tokyo.ac.jp/enju/

identification of multi-word terms and we will try to integrate a full parsing approach in a next version of OntoGain.

3.1.2 Detection of morphological variants

We proposed and implemented an enhancement to C-Value, using the morphological processor from WordNet[8] Java Library[9] (JWNL). The Morphological Processor attempts to match the form of a word or phrase to its respective lemma, i.e. base form, in WordNet. For example, if one calls lookupBaseForm(POS.VERB, "running"), the lemma "run" should be returned. This enhancement has been thought important because it allows the C/NC Value tool to handle morphological variants of terms, for example web page, web pages.

3.1.3 Linguistic Filtering

A linguistic filter is applied on each sentence to extract potential multiword terms. Moreover, if a constituent word in each candidate extracted multi-word term resides in a simple stop list, then the candidate term is rejected. The choice of linguistic filter affects the precision and recall of the output list. A 'closed' filter which is strict about the strings it permits, will have a positive effect on precision but a negative effect on recall. As an example, consider the "Noun+" filter that Dagan, Church used [Dagan I. , Church K., 1995]. This filter permits only noun sequences and as a result produces high precision since noun sequences in a corpus are the most likely to be terms. At the same time, it negatively affects recall, since there are many noun compound terms that consist of adjectives and nouns, which are excluded by this filter. We implemented all the filters below and we chose the third one for our experiments, the open filter which results in augmented recall over precision (to reveal more candidate terms):

1. Noun+Noun

[8]http://wordnet.princeton.edu/
[9]http://jwordnet.sourceforge.net/

2. (Adj | Noun)+Noun

3. ((Adj | Noun)+| ((Adj | Noun)*(NounPrep)?)(Adj | Noun)*) Noun

We used a simple stop list to filter unwanted strings. The list was manually constructed based on domain observation. After this step, main process of C-Value takes place.

3.1.4 C-Value Implementation

We describe the steps taken in the C-value method to construct a list of candidate multi-word terms from a corpus.

Step 1

We tag the corpus. As mentioned earlier, we need the tagging process since we will use a linguistic filter to restrict the type of terms to be extracted.

Step 2

This stage extracts strings satisfying the linguistic filter and frequency threshold. The terms will be extracted from among these strings. According to [Nenadic et al., 2004] the maximum length of the extracted strings depends on:

1. The working domain. In arts for example, terms tend to be shorter than in science and technology.

2. The type of terms we accept. Terms that only consist of nouns for example, very rarely contain more than 5 or 6 words.

   The process of finding the maximum length is as follows: We attempt to extract strings of a specific length (7 in our work). If we do not find any strings of this length, we decrease the number by 1 and make a new attempt. We continue in this way until we find a length for which strings exist. At this point, extraction of the candidate strings can take place. Initially, a list of strings of every possible length up to the maximum length set to above (i.e., 7 in this work) is created, (i.e. a list for the bigrams, a list for the trigrams, etc.). For each candidate string, its frequency of occurrence is computed as well. The lists are then filtered through the stop-list and are concatenated.

The longest strings appear at the top, and decrease in size as we move down, with the bigrams being at the bottom.

3. The C-value for each of the candidate strings is computed, starting with the longest ones and finishing with the bigrams. The C-value for the longest terms is given by the top branch of formula described in the C-Value section (section 3.2.2) and we quote again:

$$f(\alpha) = \begin{cases} log_2|\alpha| * f(\alpha) & \text{if } \alpha \text{ is not nested} \\ log_2|\alpha| * (f(\alpha) - \frac{1}{P(T_\alpha)} \sum_{b \in T_\alpha} f(b)) & \text{otherwise} \end{cases} \quad (3.1)$$

We set a C-value threshold, so that only those strings with C-value above this threshold are added onto the list of candidate terms. For the evaluation of C-value for any of the shorter strings, we need two more parameters:

- their frequency as part of longer candidate terms
- the number of these longer candidate terms

These two parameters are computed as follows: For every string a, that it is extracted as a candidate term and for each substring b of a, we compute tuples $(f(b), t(b), c(b))$, where

- $f(b)$ is the total frequency of b in the corpus
- $t(b)$ is the frequency of b as a nested string of candidate terms
- $c(b)$ is the number of these longer candidate terms

When this triple is first created, $c(b) = 1$ and $t(b)$ equals the frequency of α. Each time β is found after that, $t(b)$ and $c(b)$ are updated, while $f(b)$, its total frequency, does not change. c(b) and t(b) are updated in the following manner:
c(b) is increased by 1 every time b is found within a longer string a that is extracted as a candidate term. t(b) is increased by the frequency of the longer candidate term a, f(a), every time b is found as nested. If n(a) is the number of times a has appeared as nested, then t(b) will be increased by f(a) n(a). Now in order to compute C-value

for a string a which is shorter by one word, we distinguish between the following two cases: we either already have a triple $(f(a), t(a), c(a))$ for this string, or we do not. If we do not, we calculate the C-value from the top branch of the above C-Value formula. If we do, we use the bottom branch respectively. In that case, $P(T_\alpha) = c(a)$ and $\sum_{b \in T_\alpha}$.

After the calculation of C-value for strings of length l finishes, we move to the computation of C-value for strings of length $l - 1$. This way it is evident whether the string to be processed has been found nested in longer candidate terms. At the end of this step, a list of candidate terms has been built. The strings of the list are ranked by their C-value. The higher a term is in this hierarchy , the higher its probability of being a real concept for our corpus.

3.1.5 NC-Value

NC-Value is used to rerank and improve the list of the extracted multi-word terms based on information from term's neighborhood. This involves the extraction of the term context words and their weights. In order to extract the term context words, we need a set of top C-Value terms, according to [Frantzi, Ananiadou, Mima, 2000]. We use the 'top' candidate terms from the C-value list, which are expected to present very high precision on domain terms. We chose the first 1500 terms for running our experiments because those 'top' terms contain enough information to produce a list of term context words. We assign to each of them a weight following the process described in the previous chapter about weighting the context words.

The reranking takes place in the following way: Each candidate term from the C-value list appears in the corpus with the set of context words. From these context words, we retain the nouns, adjectives and verbs for each candidate term. These words may or may not have been met before, during the second stage of the creation of the list with the term context words. In the case where they have been met, they retain their assigned weight. Otherwise, they are assigned zero weight. For each candidate term, we obtain the context factor by summing up: the weights for its term context words, multiplied by their frequency of co-occurrence with this

candidate term.

The NC-Value equation is: (discussed in chapter 2 in more detail)

$$NC - Value = 0.8C - Value(a) + 0.2 \sum_{b \in C_\alpha} f_a(b) weight(b) \tag{3.2}$$

where:

- a is the candidate term
- C_α is the set of distinct context words of a
- b is a word from C_α
- $f_a(b)$ is the frequency of b as a term context word of a
- $weight(b)$ is the weight of b as a term context word

For more information about C/NC-Value the reader is referred to [Frantzi, Ananiadou, Mima, 2000].

3.2 Taxonomic Relations

We already have extracted multi-word term concepts from the previous concept extraction step. These concepts will be used to construct a domain taxonomy. OntoGain offers two options for building a domain taxonomy. The user can choose between Formal Concept Analysis or the hierarchical agglomerative clustering module. This backbone hierarchy which is created from either FCA module or agglomerative clustering module, shall be enriched later with non-taxonomic relations. The experimental results, as will be discussed in Chapter 4, demonstrate that clustering results outperform Formal Concept Analysis on both corpora.

3.2.1 Formal Concept Analysis

The input of FCA module is a set of formal objects along with their corresponding formal attributes, as discussed in Chapter 2.5.2. The formal objects consist of the extracted multi-word term-concepts, while their corresponding formal attributes are

the verbs that were recognized with the terms. More precisely, the terms are given from the C/NC-Value module, so we need to find the corresponding attributes in order to build a context lattice as discussed in 2.5.2. For this purpose we make use of syntactic dependencies between the verbs and their concepts, as they were extracted from the OpenNLP tools' shallow parser. We use these dependencies to form formal contexts which will provide the input for the *Formal Concept Analysis* algorithm (Table 3.1). Table 3.2 summarizes the steps followed for the Formal Concept Analysis module.

Table 3.1: Computer Science knowledge as a formal context

	submit	test	describe	print	compute	search
html form	⋆			⋆		⋆
hierarchical clustering					⋆	⋆
text retrieval						⋆
root node		⋆	⋆		⋆	⋆
single cluster			⋆		⋆	⋆
web page				⋆		⋆

However, not all the inferred verb-argument dependencies are valid. Moreover, not all the extracted dependencies are important for discriminating between the different concepts. Cimiano proposes *conditionaly probability*, an information measure that tries to deal with these problems [Cimiano et al., 2005]. He compares conditional probability with the *pointwise mutual information (PMI)* proposed by Hindle [Hindle, 1990] and the measure proposed by Resnik [Resnik, 1997] for computing the importance of each verb-concept pair (i.e. the *selectional strength*) upon the document collection. In particular *conditional probability* (Eq. 3.1),

$$Conditional(n, v_{arg}) = P(n|v_{arg}) = \frac{f(n, v_{arg})}{f(v_{arg})} \tag{3.1}$$

computes the selectional strength of a term n with respect to the corresponding verb v that appears with.

Table 3.2: OntoGain FCA algorithm module

Input	Verb - object pairs found in documents
Output	taxonomy in OWL language
Step 1	Verb - Object pairs input
Step 2	Computation of conditional probability for each pair
Step 3	Import all pairs with higher probabibility than threshold t
Step 4	Formal context formulation
Step 5	Computing the concept lattice
Step 6	Reduced labeling
Step 7	OWL statements creation

Hindle [Hindle, 1990], suggests the pointwise mutual information measure below for measuring the importance of a verb-term pair appearing together in a corpus sentence:

$$PMI(n, v_{arg}) = log_2 \frac{P(n|v_{arg})}{P(n)} \tag{3.2}$$

where

- $P(n|v_{arg})$ the same as conditional probability above
- $P(n)$ is the relative frequency of a term n compared to all other terms

Finally, Resnik [Resnik, 1997] suggests computing the importance of a verb-term pair as:

$$Resnik(n, v_{arg}) = S_R(v_{arg})P(n|v_{arg}) \tag{3.3}$$

where $S_R(v_{arg}) = \sum_{n'} P(n'|v_{arg}) log \frac{P(n'|v_{arg})}{P(n')}$ and

- $f(n, v_{arg})$ is the total number of occurences of a term n as argument arg of a verb v
- $f(v_{arg})$ is the number of occurences of verb v with an argument
- $P(n)$ is the relative frequency of a term n compared to all other terms

Cimiano [Cimiano et al., 2005] claims that the conditional probability measure outperforms the other two information measures. Based on this result, we decided to adopt conditional probability as a measure of the importance of verb-tem pairs. We consider only those verb-argument relations that exceed some threshold t, in order to try to overlap the problems regarding the computation of dependencies and data sparseness. Next we present an example showing how we compute the weights to objects-attributes. Table 3.3 illustrates the occurences of each concept in the corpus with respect to its corresponding verb, while Table 3.4 illustrates the conditional probability measure computed to each term.

Table 3.3: Occurences of multi-word term concepts as objects of verbs

	submit	test	describe	print	compute	search
html form	8			4		2
hierarchical clustering					4	7
text retrieval						9
root node		6	8		4	5
single cluster			10		6	8
web page				9		4

We experimented with several different threshold values in order to discover noticeable discrimination between important and non-important verb-concept relations. We finally choose a threshold $t = 0.003$ to apply in both corpora. The dependencies below this threshold were discarded and were not considered during the process of building the taxonomy. This threshold can be altered in OntoGain according to user's preferences.

Several algorithms have been proposed for computing concept lattices. One of the most efficient is the *Next-Closure algorithm* by Ganter, [Ganter & Reuter, 1991]. For the implementation of FCA we used the *colibri* Java library by Christian Lindig[10] which implements the Next-Closure algorithm. We show the output lattice

[10]http://www.st.cs.uni-saarland.de/ lindig/

Table 3.4: Conditional Probabilities

	submit	test	describe	print	compute	search
html form	$\frac{8}{8} = 1$			$\frac{4}{13} = 0.31$		$\frac{2}{35} = 0.06$
hierar clustering					$\frac{4}{14} = 0.29$	$\frac{7}{35} = 0.2$
text retrieval						$\frac{11}{35} = 0.31$
root node		$\frac{6}{6} = 1$	$\frac{8}{18} = 0.44$		$\frac{4}{14} = 0.29$	$\frac{5}{35} = 0.14$
single cluster			$\frac{10}{18} = 0.56$		$\frac{6}{14} = 0.43$	$\frac{8}{35} = 0.23$
web page				$\frac{9}{13} = 0.69$		$\frac{4}{35} = 0.12$

for this example in Figure 3.1 and the corresponding taxonomy in Figure 3.2 and Figure 3.3. The output taxonomy (Figure 3.3) consists of Jena OWL statements as discussed in the beginning of this chapter (Table 3.5).

3.2.2 Hierarchical Clustering

Hierarchical agglomerative clustering proceeds bottom-up. It starts with the documents as individual clusters and, at each step, computes the similarity between all pairs of clusters and merges the most similar pair. The algorithm typically continues until a single cluster is formed at the top of the hierarchy. We used the group average method to compute the similarity between two clusters. Being more specific, the group average method computes the average similarity across all pairs of concepts within the two clusters (C_i, C_j) that will be merged:

$$sim(C_i, C_j) = \frac{\sum_{x \in C_i, y \in C_j} sim(x, y)}{|C_i| * |C_j|} \tag{3.4}$$

where x is a concept in cluster C_i and y in cluster C_j correspondingly.

Table 3.6 demontrates the steps followed to implement the hierarchical clustering module. The time complexity of a typical hierarchical agglomerative clustering algorithm is $O(n^2)$ where n is the number of concepts.

Lexical similarity [Nenadic et al., 2004] was used to measure similarity between multi-word term cluster concepts. This idea was exploited by Bourigault

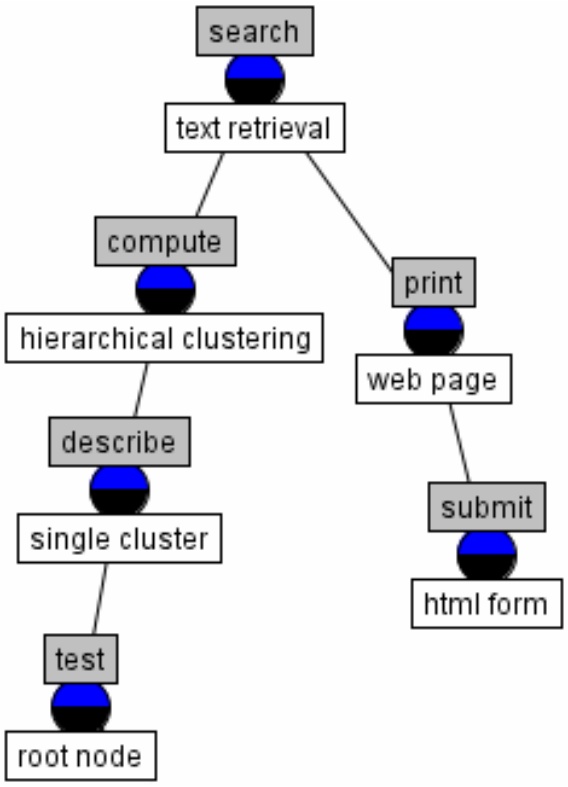


Figure 3.2: Sample FCA hierarchy of computer science terms

& Jacquemin [Bourigault & Jacquemin, 1999] by adapting the term variation process, and by Dagan & Church [Dagan I. , Church K., 1995] via "grouping" the list of term candidates according to their heads[11]. These approaches were generalized by considering constituents (head and modifiers) shared by terms. Therefore the rationale behind lexical similarity involves the following hypotheses [Nenadic et al., 2004]:

1. Terms sharing a head are assumed to be (in)direct hyponyms of the same term *(e.g. progesterone receptor and oestrogen receptor are both receptors).*

2. When a term is nested inside another term, we assume that the terms are related *(e.g. retinoic acid receptor and retinoic acid should be associated).*

The lexical similarity between term t_1 and term t_2 (whose heads are denoted by h_1 and h_2 respectively) is computed according to a Dice-like coefficient formula.

[11]For terms "web page" and "web snippet" the head comprises of the word "web".

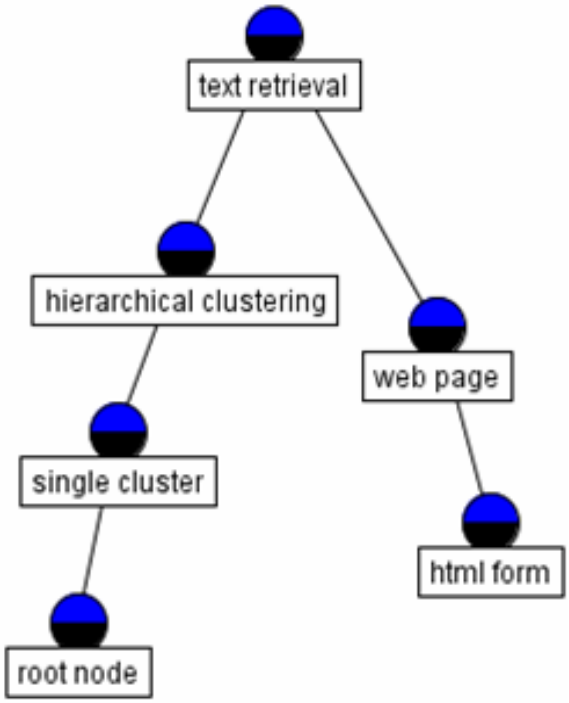


Figure 3.3: FCA taxonomy output without attributes

Building upon the idea of Dice coefficient, the lexical similarity between two terms is computed as :

$$LS(t_1, t_2) = \frac{P(h_1) \cap P(h_2)}{P(h_1) + P(h_2)} + \frac{P(t_1) \cap P(t_2)}{P(t_1) + P(t_2)} \tag{3.5}$$

The numerators in the previous formula (3.5) denote the number of shared constituents, while the denominators denote sums of total numbers of constituents. Table 3.7 illustrates some examples of lexical similarity between multi-word terms. For more details about lexical similarity the reader is referred to [Nenadic et al., 2004].

For creating a taxonomy of terms, the clustering process is terminated before a unique cluster remains. More specifically, clustering repeats as long as the merged clusters share common term heads. Furthermore, the lexical similarity measure gives credit to the shared heads between two similar multi-word terms. For this reason the created clusters consisted of mainly terms with shared heads. We used this feature to label appropriate the top clusters of the derived concept hierarchy. Figure 3.4 shows a part of the taxonomy output in Protege.

The clustering algorithm for creating the domain taxonomy resulted in better and more meaningful results than FCA as we will discuss in Chapter 4.

3.3 Non-Taxonomic Relations

In what follows we show how OntoGain extracts non-taxonomic relations. These relations will enrich our previously computed taxonomy. They will help towards the transformation of the taxonomy into an ontology. The use of the previously inferred taxonomy knowledge upon these methods is crucial, as both of them use information from the extracted hierarchies to form the final relationships between concepts.

3.3.1 Association Rules

Our learning approach is based on the algorithm for discovering generalized association rules proposed by Srikant & Agrawal [Srikant & Agrawal, 1995]. Their algorithm discovers associations between items in a set of transactions (e.g. supermarket products). It finds relationships between customers purchases aiming to describe them at the proper level of abstraction: "snacks are purchased together with drinks" comprises a valid relation, rather than "chips are purchased with beer" and "peanuts are purchased with soda".

We consider the extension of Srikant and Agrawal [Srikant & Agrawal, 1995] to determine the associations at the appropriate level of generalization with respect to a given taxonomy [Madche & Staab, 2000]. Each "subject-verb-object" triple found in text is enhanced with more general terms (super concepts) of the concepts it contains. Further, we eliminate rules $X \Rightarrow Y$ where Y contains of some element in X. Additionally we also exclude those rules that are included by an ancestral rule $X' \Rightarrow Y'$. Such ancestral rule means that X contains only subconcepts for all concepts in X' (analogously for Y and Y').

Initially, the taxonomy is loaded in memory. Upon this taxonomy the extracted rules will be attached to form the ontology in OWL. Next we load the system with all the subject - verb - object triples that were extracted from the shallow parsing

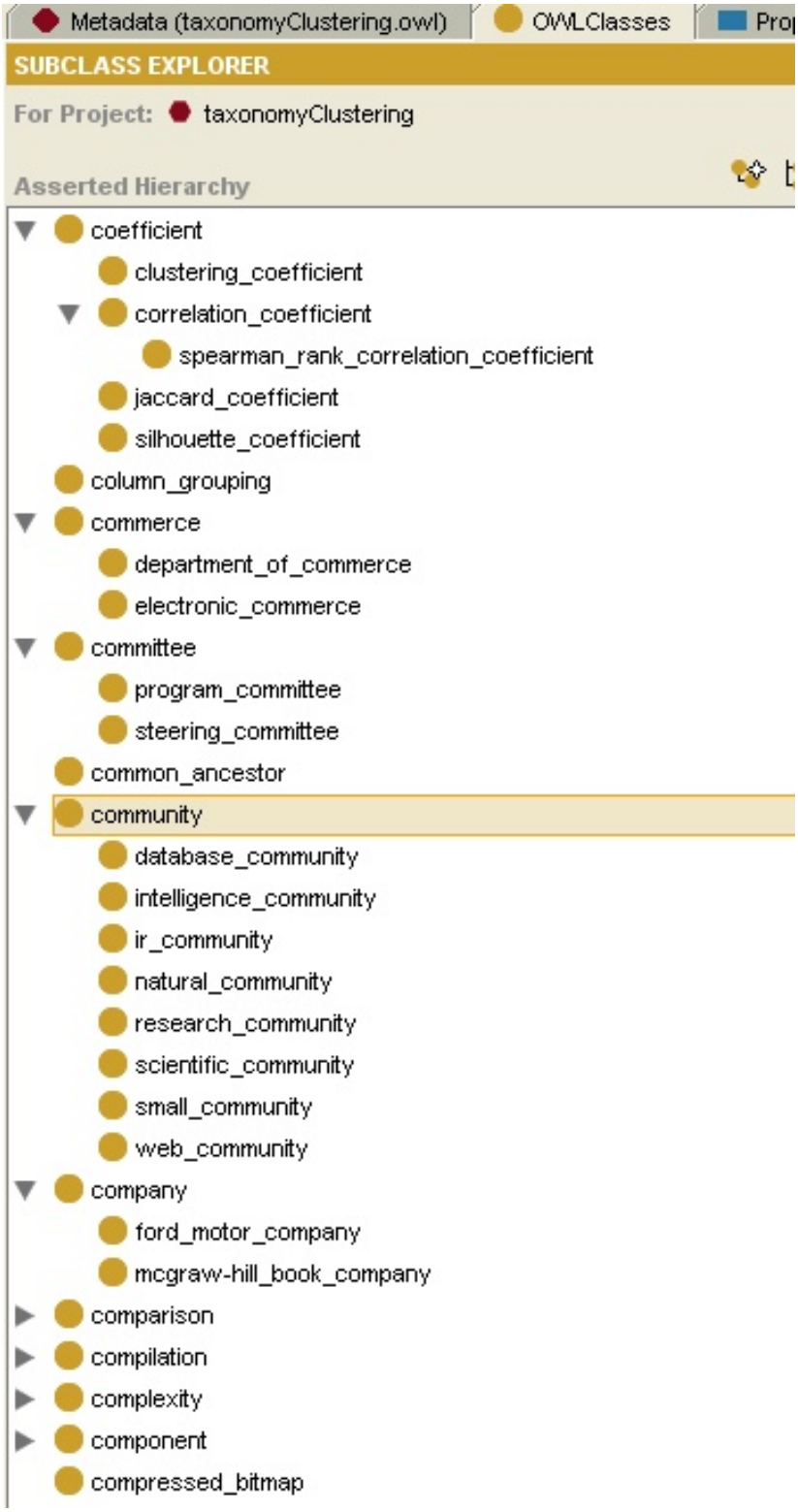


Figure 3.4: Clustering taxonomy OWL output in Protege

process. For the mining of rules we used WEKA[12], an open source software which comprises of a collection of machine learning algorithms for data mining tasks. *Apriori* is a classic algorithm for extracting association rules [Agrawal & Srikant, 1994]. OntoGain provides the *predictive apriori* algorithm to mine association rules which is an enhancement to the classic apriori algorithm [Scheffer, 2001]. As mentioned in Chapter 2.6.1, in order to select interesting rules from the set of all possible rules, constraints on various measures of significance and interest can be used. The best-known constraints are minimum thresholds on support and confidence. Predictive apriori algorithm proposes a trade-off between confidence and support which maximizes the chance of correct predictions. Each rule then is not followed by a confidence and support value to imprint its importance but is evaluated by a metric called *predictive accuracy*[13]. We used this measure for detecting rules exceeding a defined predictive accuracy threshold. After several experiments we accept rules that have a predictive accuracy higher than the threshold $acc = 0.3$. In OntoGain the user may alter this value according to his/her needs.

Table 3.8 summarizes the steps of the rule mining algorithm. Table 3.9 illustrates example rules extracted by the algorithm.

3.3.2 Verbal Expressions

This OntoGain module implements a method for finding the right level of abstraction between the domain and range of a relation with respect to a given concept hierarchy. The relations are extracted via the shallow parsing process. It is necessary for an ontology to allow representing relations (or rules) at the appropriate level of generalization, thus eliminating the necessity of representing each case explicitly and so avoiding redundant information [Cimiano et al., 2006]. This approach does not filter any of the extracted verb-concept relations (like the association rules module for deriving non-taxonomic relations) but tries to provide the proper level of

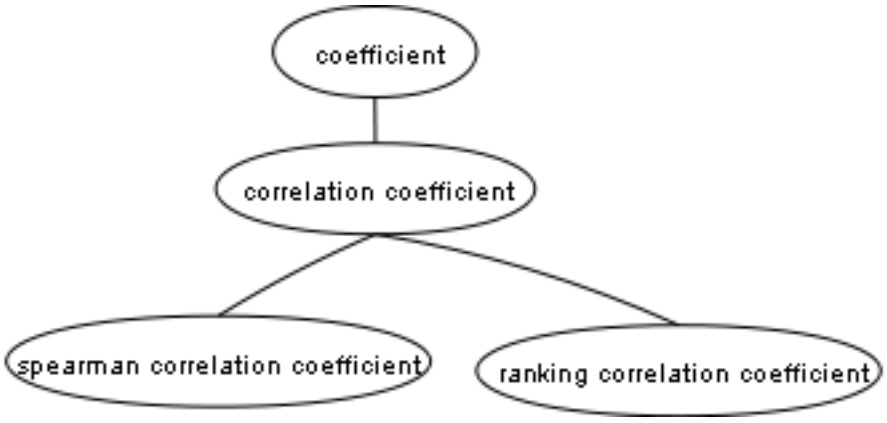
[12]http://www.cs.waikato.ac.nz/ml/weka/

[13]For more information about the combination of support and confidence that form the predictive accuracy metric, the reader is referred to [Scheffer, 2001].

abstraction for each of the relation's concepts (for the domain and range of each relation respectively). The user can choose only one of these methods (association rules or verbal expressions) for extracting non-taxonomic relations as the two methods produce overlapping results. We will discuss in Chapter 4 about the experimental results of each method.

We apply the output of the shallow parser module to determine verbal relations. These binary relations are labeled using the lemmatized verb and the corresponding multi-word terms as domain and range of the relation respectively. We have now collected various relations from the corpus, hence we target in finding the most proper generalization for the concepts within the domain and range of each relation. Cimiano et al. [Cimiano et al., 2006] experimented with different measures that can be used for this purpose including *conditional probability*, *pointwise mutual information* and a χ^2*-based* measure. According to Cimiano, the conditional probability outperforms the others, thus we applied the *conditional probability* measure to detect the most appropriate generalization for the concepts.

As an example, the concepts that appear in the corpus together with the verb *rank* are "correlation coefficient", "spearman correlation coefficient" and "ranking correlation coefficient" with frequencies 15, 10 and 5 respectively.



According to this approach the frequencies are propagated through all the ancestral concepts. If two or more concepts share the same conditional probability, we choose the most specific one according to the domain taxonomy. Thus *coefficient* has probability $30/30 = 1$, *correlation coefficient* $= 30/30 = 1$, *spearman correlation coefficient* $= 10/35 = 0.29$, *ranking correlation coefficient* $= 5/35 = 0.14$.

Two concepts have the same highest probability, *coefficient* and *correlation coefficient*. Finally we keep *correlation coefficient* as the proper generalization for the object position of *rank*. We choose the most specific one between concepts sharing the same value.

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <owl:Class rdf:about="#hierarchical clustering">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#text retrieval"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#web page">
    <rdfs:subClassOf rdf:resource="#text retrieval"/>
  </owl:Class>
  <owl:Class rdf:about="#single cluster">
    <rdfs:subClassOf rdf:resource="#hierarchical clustering"/>
  </owl:Class>
  <owl:Class rdf:about="#html form">
    <rdfs:subClassOf rdf:resource="#web page"/>
  </owl:Class>
  <owl:Class rdf:about="#root node">
    <rdfs:subClassOf rdf:resource="#single cluster"/>
  </owl:Class>
</rdf:RDF>
```

Table 3.5: OWL output for the FCA described example

Table 3.6: OntoGain hierarchical clustering algorithm module

Input	Multi-word term concepts
Output	Taxonomy in OWL
Step 1	Load multi-word term concepts from previous step
Step 2	Treat each concept as a cluster on its own
Step 3	Compute the similarity between all pairs of clusters, calculate the similarity matrix whose i^{th} entry gives the similarity between the i^{th} and j^{th} clusters.
Step 4	Merge the most similar two clusters (Group Average).
Step 5	Update the similarity matrix entries for the newly formed cluster and the other clusters.
Step 6	Repeat steps 4 and 5 until desired clustering level (or when one cluster remains).

Table 3.7: Lexical Similarity Examples

i	t_i	$P(t_i)$
1	*nuclear receptor*	{*nuclear, receptor, nuclear receptor*}
2	*orphan receptor*	{*orphan, receptor, orphan receptor*}
3	*orphan nuclear receptor*	{*orphan, nuclear, receptor, orphan nuclear, nuclear receptor, orphan nuclear receptor*}
4	*nuclear orphan receptor*	{*nuclear, orphan, receptor, nuclear orphan, orphan receptor, nuclear orphan receptor*}
LS(t_1, t_2)=0.67, LS(t_1, t_3)=0.83, LS(t_1, t_4)=0.72, LS(t_2, t_3)=0.72, LS(t_3, t_4)=0.75		

Table 3.8: OntoGain Association Rules algorithm module

Input	Subject - Verb - Object triples found in documents
Output	Taxonomy enrichment - Ontology formulation in OWL
Step 1	Load taxonomy in memory
Step 2	Subject - Verb - Object input
Step 3	Step 2 enrichment with hypernyms from the taxonomy
Step 4	Compute predictive accuracy for all relations
Step 5	Allow relations above a threshold t
Step 6	Prune relations subsumed by an ancestral rule
Step 7	Inject relations in the preloaded taxonomy from step 1

Table 3.9: Sample Association Rules module output

Domain	*Range*	*Label*
chiasmal syndrome	pituitary disproportion	cause by
medial collateral ligament	surgical treatment	need
blood transfusion	antibiotic prophylaxis	result
lipid peroxidation	cardiopulmonary bypass	lead to
prostate specific antigen	prostatectomy	follow
chronic fatigue syndrome	cardiac function	yield
right ventricular infarction	radionuclide ventriculography	analyze by
creatinine clearance	arteriovenous hemofiltration	achieve
sudden cardiac death	tachyarrhythmias	cause
cardioplegic solution	superoxide dismutase	give
bacterial translocation	antibiotic prophylaxis	decrease
accurate diagnosis	clinical suspicion	depend
ultrasound examination	clinical suspicion	give
total body oxygen consumption	epidural analgesia	attenuate by
coronary arteriography	physician assistant	perform by

Chapter 4

Evaluation

Evaluating an automatically crafted ontology is admittedly a difficult and challenging task [Brank et al., 2005]. In general there is not a standard way to model a domain in question. Different engineers could produce contradictory results and this is reasonable as knowledge is not described explicitly upon free texts. Ontology learning results are typically evaluated using *recall* which is defined as the amount of knowledge correctly identified with respect to all the knowledge that exists in the corpus. However, the notion of *all knowledge* that is described in a text collection is extremely subjective [Brewster et al., 2004]. Two main approaches dealing with this problem are the comparison with a hand-crafted "Gold Standard" ontology and the evaluation by a domain specialist, decomposing the extracted ontology to its constituent parts (i.e. concept terms, taxonomic & non-taxonomic relations) [Brank et al., 2005].

We seeked for a golden standard ontology as a benchmark for our comparisons. *Genia corpus* (see Chapter 2.7) contains a reference ontology which could be used for our purpose, but this corpus is structured and annotated. Consequently, we could not use the Genia corpus, as one of our main goals is to evaluate ontology construction starting from plain text. Furthermore GENIA contains very specialized terminology about gene names. That makes it hard for the preprocessing and the concept extraction step to work correctly[1]. Also Madche [Madche, 2002] used a

[1] a sample phrase: *Transient transfection experiments show that several elements in the*

gold standard ontology for his experiments, but the corresponding corpus consists of texts in German.

According to Sabou [Sabou, 2005], the use of a golden standard ontology for the evaluation of an automatically extracted ontology can lead sometimes to erroneous conclusions regarding the quality of the learned ontology. Another way to evaluate an automatically built ontology is the judgement of the results by domain experts [Brank et al., 2005]. For OntoGain, relevance judgements are provided by domain experts and only for the computer corpus, since for OHSUMED and medical texts we have no solid domain expertise. We evaluate each module of OntoGain in terms of *precision*, namely the ratio of the number of correct (relevant) output results with respect to the total number of the results examined. The domain expert evaluated the first 150 output concepts, on the main OntoGain modules: concept extraction, clustering, FCA, association rules and verbal expressions. The results showed significant performance and very reasonable results, as we will see in Section 4.3.

We also compare OntoGain with results obtained by Text2Onto[2] system on two separate text corpora (as discussed in Chapter 2.7), a medical (OhsuMed collection) and a computer science corpus. OhsuMed collection ([Hersh et al., 1994], [Hersh & Hickam, 1994]) contains a total of 348,566 references from MEDLINE[3], the on-line medical information database[4]. The computer science corpus constitutes of various computer science papers and articles and was used in the work of Milios et al. [Milios et al., 2003].

The experimental results will demonstrate the advantages of OntoGain over Text2onto: OntoGain builds compact ontologies with more solid and distinctive semantics compared with Text2Onto which produces lots of term concepts and relations. However, only a few of them are representative of the text domain at hand.

---

promoter-proximal region of the IL-2R alpha gene contribute to IL-1 responsiveness, most importantly an NF-kappa B site.

[2]http://ontoware.org/projects/text2onto/

[3]http://www.nlm.nih.gov/databases/databases_medline.html

[4]http://ir.ohsu.edu/ohsumed/ohsumed.html

4.1 Concept Extraction

The extraction of term concepts is a prequisite for all aspects of ontology learning from text. Terms represent important information related to a corpus, as they linguistically represent the concepts in documents, express the semantic content of texts and characterize the documents semantically. This process is considered a difficult task and it is usually carried out by human experts. However, this process tends to be slow and subjective and does not scale-up well for large document collections.

For this purpose we used the *C/NC-Value* method [Frantzi, Ananiadou, Mima, 2000] to deal with the extraction of multi-word term concepts. C/NC-Value is a domain-independent method for the automatic extraction of multi-word terms, combining linguistic and statistical information. It enhances the common statistical measure of frequency of occurrence and incorporates information from context words to the extraction of terms. Tables A.1 and A.2 illustrate top results of the output of C/NC-Value in the computer science corpus and OhsuMed respectively. The value beside each multi-word concept denotes the likelihood of each one being a valid term, namely its C-Value measure. An expert in the computer domain evaluated the first 150 terms extracted by C/NC-Value and resulted in a precision of 86,67% which shows that the output is highly reasonable (as shown in Table 4.1).

Formulating an ontology lexicon with reasonable terms is an important prequisite for the determination of relations that model appropriately the domain in question.

4.2 Taxonomic & Non Taxonomic Relations

In OntoGain we implement and compare FCA, a set-theoretic approach, with an hierarchical clustering approach regarding to speed, effectiveness and the ability to create understandable classes. An advantage of the FCA technique is that due to attribute inheritance provides with a reasonable description for the clusters (formal concepts) in the hierarchy. At the same time clustering-based methods are based on a plain numerical value denoting the similarity between clusters, thus resulting

in non-labeled concept clusters. Theoretically an ontology engineer will be assisted by Formal Concept Analysis in terms of better formal interpretation of taxonomy clusters.

However the experimental results did not show good results for the Formal Concept Analysis method. FCA is based on corpus verb dependencies and due to the big size of our input many dependencies were spurious and led to erroneous results. Most of concepts appeared with many different verbs, so the corresponding concept lattices were huge and did not lead to reasonable results on both corpora. FCA takes into account verb appearances, attaching them to the candidate terms and gives credit to shared verb attributes in order to establish a taxonomic relation. However, we observed that on several potential related concepts were assigned distinct different verb attributes so that most of the times either no single relation could be identified as candidate for forming taxonomic relations, or after observation by a domain expert there were erroneous relations with no meaning.

We tried to deal with this problem with two ways: The first idea relies on the application of conditional probability in an effort to measure the importance of the different verb attributes associated with each concept (as discussed in Section 3.2.1). We experimented with different thresholds. The second approach applies a categorization of the candidate verb attributes based on their synonym sets in WordNet. For example, the verbs {supply, give, provide} were thought as synonyms and were considered as the same verb. Despite all these efforts it seemed that hierarchical clustering outperformed FCA, which extracted more reasonable taxonomies. Appendix contains examples of derived concepts, taxonomic and non-taxonomic relations from computer science and OhsuMed corpora. Another disadvantage of FCA is the exponential time $O(2^n)$ worst case complexity $O(2^n)$, as opposed to the quadratic time complexity $O(N^2)$ of agglomerative clustering. Considering all the above and based on measurement of precision, we conclude to the fact that clustering outperformed Formal Concept Analysis on both corpora.

We considered two approaches towards the extraction of non-taxonomic relations. The method based on verbal relations does not filter the relations that are extracted from the shallow parsing process, contrarily to the association rules method. This

method tries to find the proper generalization level for concepts that form the already discovered relations from the shallow parsing process. As observed by Gulla [Gulla & Brasethvik, 2008], the evaluation of ontology relationships constitutes a difficult task, as there are many potential relationships between concepts and mostly subjective judgment from a domain expert can reveal which ones are of most importance. As we will show in Section 4.3, the domain expert assigned for evaluating the extracted relations, found them highly reasonable.

The first idea for deriving non-taxonomic relations originates from data mining, using association rules discovery algorithms. According to this approach, OntoGain attempts to predict valid rules, taking corpus concept dependencies as input. Although this approach cannot predict all valid relations, it is highly sufficient in identifying at least some of the most important concept relationships. The output rules are compact and have 'dense' meaning, in regard to the effort of modelling the domain in question.

A second approach attempts to discover the appropriate generalization level for verb-based relations that appear in the corpus, with respect to a given domain taxonomy. From an ontology point of view it is very important to discover the most general relation that describes all the relation instances with respect to the concepts described on our given taxonomy, to avoid representing each case explicitly (as shown through the examples in Chapter 2.6.2). We applied the conditional probability measure in order to find the correct level of generalization in the concept hierarchy. It appears difficult to reveal the appropriate generalization level of non-taxonomic relations due to fact that the concept hierarchies are in most cases shallow taxonomies. Based on our results, we prefer the association rules method, as it seems to outperform the method based on the generalization of verbal relations. It produces highly reasonable dependencies and it prunes less significant ancestral rules and relations for modelling the domain.

4.3 Results assessment by domain experts

As discussed in the beginning of this chapter, we asked a domain expert to provide with human relevance judgments on results obtained by OntoGain on the computer science corpus. We took the first 150 terms extracted by C/NC-Value and we collected all the results obtained by OntoGain modules for these 150 terms. The domain expert had to decide whether the terms and the relations were correct or not. In the following, we report results of *precision*. Precision is defined as the ratio of the number of correct (relevant) output results to the total number of the results examined. In our case, we examined 150 results. Table 4.1 illustrates the performance of OntoGain modules. The results show that C/NC-Value performs very well in the task of identifying concepts that will form the ontology lexicon, the basis of the ontology construction process. For forming the concept hierarchy, clustering clearly outperforms the Formal Concept Analysis module. Furthermore, association rules seem to deliver better non-taxonomic relations than the verbal expressions module.

Table 4.1: Results of human evaluation

Method	Precision
Concept Extraction	86.67%
Formal Concept Analysis	44.2%
Hierarchical Clustering	71.33%
Association Rules	72.85%
Verbal Expressions	61.67%

4.4 Comparison with other methods

Text2Onto[5] [Cimiano & Volker, 2005] is a framework for ontology learning from textual resources. OntoGain is capable of extracting concepts, subclass-of (taxonomic) relations, as well as properties and relations. It is further capable of concept instantiation.

For the concept extraction task Text2Onto, unlike OntoGain, extracts mostly single-word terms most of which are lacking semantic meaning and therefore, are not descriptive of the examined domain. Figures 4.1 and 4.2 show a snapshot of Text2Onto output respectively. Examining the sample snapshots (Figures 4.1 and 4.2), we observe that there are many extracted single word terms, with no interest to the ontology engineer, like 'ct', 'rcbf', 'acn'. Additionally, there are words with very general meaning, which also are of no use to the end user like 'pin', 'test', 'need'.

It is worth mentioning that in large corpora like OhsuMed, Text2Onto extracts too many concepts, resulting to chaotic lists of terms and relations. At the same time, multi-word terms are vested with more distinctive semantics for the purpose of forming a compact ontology lexicon. Multiword terms are therefore more suitable than single word terms for modeling an application domain. A sample concept output extracted by OntoGain is shown in Tables A.1 and A.2.

Text2Onto was not capable of processing full OhsuMed corpus even though we splitted to smaller segments and pass them as input. It was running out of memory even if we ran it on an 64-bit server reserving 3 GB of heap space. This owes to the fact that texts of ∼250 Mbytes approximately (like OhsuMed) consist of hundreds of thousands of single-word term concepts. Furthermore the effort of extracting taxonomic and non-taxonomic relations from such amount of data leads to program crash. This observation strengthens our assumption that multi-word terms lead to compact respresentation of the examined domain, yielding dense and meaningful listings of multi-word term concepts.

The following Figures (4.1 and 4.2) present a sample output from Text2Onto

[5]http://ontoware.org/projects/text2onto/

system for 1000 lines of OhsuMed input. Figure 4.1 illustrates sublass-of relations used to form a concept hierarchy, while Figure 4.2 indicates non-taxonomic relations. Notice that most relations do not actually correspond to significant information in the corpus while, the majority of term relationships are either wrong or meaningless. Notice finally that contrast to OntoGain, Text2Onto does not export results in OWL.

A major advantage of OntoGain over Text2Onto is that is outputs the results of each step into OWL, either when forming the taxonomy or when the ontology is formulated after the non-taxonomic relations are attached. At every step the ontology engineer can view the results in an ontology editor as Protege. For example, the ontology engineer can edit the output of the taxonomy extraction step and then load the new taxonomy again in OntoGain and therefore attempt to discover new types of non-taxonomic relations (as both non-taxonomic methods rely on the previously computed hierarchy).

Concept | Instance | Similiarity | SubclassOf | InstanceOf | Relation | SubtopicOf

Domain	Range
test	substitute
smoking	lifestyle aspect
autotransfusor	aid
arc welding	equipment
pigmentosa	dermatosis
acn	skin organism
acetate	salt
enzyme activity	change
system	model
regression	result
thrombosis	complication
need	complication
intervention	reason
withdrawal symptom	problem
lymph node metastasis	involvement
expression	datum
tumor	extirpation
physician	health professional
family support	part
provision	intervention technique
ventricular myocyte	mechanism
imazodan	agent
distortion	stimulus
anticoagulant therapy	problem
osteoma	bone tumor
synovitis	symptom
cream	preparation
flux	cause
colophony	flux
child	age group
risk	development
symptom	emergency room
progesterone receptor synthesis	function
prolactin	mitogen
somatostatin	vagal neuropeptide

Figure 4.1: Example of class-subclass relationships extracted by Text2Onto from HSUMED.

Concept | Instance | Similiarity | SubclassOf | InstanceOf | Relation | SubtopicOf

Label	Domain	Range
complement	ct	examination
count_before	platelet	treatment
present_from	paper	population survey
emphasize_of	case report	arterial pressure
change	rcbf	patient
decide	oma	[
add	study	credence
remain_without	patient	evidence
grow_as	neck	neoplasm
leave_per	tea	person
begin_at	depolarization	membrane
differ_in	vascular disease	man
occur_at	block	nanomolar concentration
explore_of	author	eve
explore	author	impact
stress_of	study	treatment history
stress	study	importance
generate_in	reticulum	rat
generate	reticulum	action potential
lack	kidney	reserve capacity
avoid	exposure	injury
record	uclum	anteroposterior force
approach	deformity	ankle
reaffirm	case report	difficulty
include	adhesive capsulitis	prevention
process	family interactional	infant feeding
mature_as	sport medicine	discipline
concentrate_from	vitro	bead
involve_from	infection	extraarticular focus
use_of	fixator	pin
present_with	case report	osteoid osteoma
present_for	investigation	importance
establish_of	arthrography	adhesive capsulitis
establish	arthrography	diagnosis
constitute	aberration	challenge

Figure 4.2: Example of non-taxonomic relationshipes extracted by Text2Onto from HSUMED.

Chapter 5

Summary and Future Work

In this thesis we focused in methods for ontology learning from texts and we proposed OntoGain, a complete prototype system consisting of distinct layers. Building upon plain term extraction, a concept hierarchy is initially formed which is then enhanced with non-taxonomic relations. Several state-of-the-art methods are examined as candidates for implementing each step. We concentrate on multi-word term concepts, as multi-word or compound terms are vested with more solid and distinctive semantics than plain single word terms. OntoGain is capable of of producing ontologies of reasonable size even for large text inputs.

OntoGain extracts significant multi-word concepts rather than large sets of meaningless single-word terms lacking semantics as most of its competitors do, helping the ontology engineer to model a domain in a more solid and compact way. We used the *C/NC-Value* method for the multi-word concept extraction. C/NC-Value is a hybrid domain-independent method that combines linguistic and statistical criteria. We have further introduced two different approaches for forming the backbone of the domain ontology, *Formal Concept Analysis* and *hierarchical clustering*. We implemented a method for the extraction of non-taxonomic relations, with respect to the previously inducted taxonomy. This method is based on *data mining* and more specifically on the *association rules* method [Agrawal et al., 1993]. We also implemented a method for learning automatically the appropriate level of abstraction for the domain and range of each relation.

Issues that merit further investigation include: Experimentation with alternative clustering algorithms such as the one suggested by Cimiano [Cimiano et al., 2005] which reveals superclasses with the help of a "hypernym oracle". Additionally alternative similarity measures can be used for computing the level of synonymity between multi-word terms, probably in combination with "lexical similarity" that was used in this thesis. It would be interesting to try discovering which similarity measures and weighting measures are working better for the process of forming the concept taxonomy. Probably we could yield improved results with a better measure that takes more factors into consideration than plain sharing of common constituents and modifiers. Of course corpus selection is related with this hypothesis.

A combination of Formal Concept Analysis with clustering, in an effort to produce better domain taxonomy representations, is also an issue for future research. Additionally, we may experiment with different measures than conditional probability to try dealing with the problems described in Chapter 3.2.1 that led in spurious results on the application of FCA in both corpora.

We could also exploit knowledge from WordNet dictionary, in an attempt to enrich the taxonomy extracted from the clustering process. Because of the multiple senses and paths for each concept in WordNet dictionary that leading to ambiguation, we could experiment with methods as the one suggested by OntoLearn [Velardi et al., 2002] that deals with the problem of ambiquity. The so called SSI (Structural Semantic Interconnection) method is applied for sense disambiguation on the basis of patterns representing paths in WordNet (for details refer to [Navigli & Velardi, 2004]). We argue for the inclusion of such a word sense disambiguation algorithm in OntoGain. The user should have the option to choose if knowledge from WordNet would be entered in the taxonomy, or the method would be left unsupervised as it is now.

In addition, we could use Hearst lexico-syntactic patterns (see Chapter 2) to reveal "subclass-of" or "part-of" relations and enrich our domain ontology. Hearst patterns can applied to derive more hypernyms to the previously extracted taxonomy. Further enrichment can be achieved with the learning of attributes in combination with non-taxonomic relations. Attributes as discussed in Chapter 2, are

relations with a datatype as range (ie. string, integer etc) and for example: *size* is one of {small, little, large}, *color* is one of {black, blue, red}. Also, we could use a full parser in order to analyze our texts, instead of tha shallow parsing technique that OntoGain uses in the current version.

For finding the most appropriate generalization level for our relations, more elaborate linguistic filtering could be used to reveal the proper level of abstraction. Furthermore OntoGain can be extended to support cardinality constraints which were not examined in the scope of this thesis, in the form of axioms (as discussed in Chapter 2.1.2) (e.g. every country has a unique capital), the same as *symmetry, transivity* for relations or *disjointness, equivalence* for concepts.

Bibliography

[Agrawal et al., 1993] Agrawal, R., Imielinski, T., Swami, A.N.: Mining Association Rules between Sets of Items in large Databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (1993)

[Agrawal & Srikant, 1994] R. Agrawal, R. Srikant (1994). Fast algorithms for mining association rules in large databases . Proc International Conference on Very Large Databases, pp. 478-499. Santiage, Chile: Morgan Kaufmann, Los Altos, CA.

[Alexiev et al., 2005] Alexiev, V., Breu, M., de Bruijn, J., Fensel, D., Lara, R., and Lausen, H., editors (2005). Information Integration with Ontologies: Experiences from an Industrial Showcase. Wiley.

[T. Berners-Lee, 1999] Berners-Lee, Tim; Fischetti, Mark (1999). Weaving the Web. HarperSanFrancisco. chapter 12. ISBN 9780062515872.

[Berners-Lee et al., 2001] T. Berners-Lee, J. Handler, and O. Lassila: The Semantic Web, Scientific American, May 2001.

[Bisson et al., 2000] Bisson et al. Designing clustering methods for ontology building, 2000

[Brand, 2006] Brand, M., "Fast Low-Rank Modifications of the Thin Singular Value Decomposition", Linear Algebra and Its Applications, Vol. 415, Issue 1, pp. 20-30, May 2006

[Brank et al., 2005] J. Brank, M. Grobelnik, and D. Mladenic. A survey of ontology evaluation techniques. In SIKDD 2005 at Multiconference IS 2005, 2005.

[Buitelaar et al., 2004] Buitelaar, P., Olejnik, D., and Sintek, M. (2004). A Protege plug-in for ontology extraction from text based on linguistic analysis. In Proceedings of the 1st European Semantic Web Symposium (ESWS), Heracleion, Greece. pages 31-44.

[Buitelaar et al., 2005] P. Buitelaar, P. Cimiano, and B. Magnini. Ontology Learning from Texts: An Overview. In P- Buitelaar, P. Cimiano, B. Magnini, Ontology Learning from Text: Methods, Evaluation and Applications volume 123 of Frontiers in Artificial Intelligence and Applications. IOS Press, July 2005.

[Bourigault et al., 1996] Bourigault, D., Gonzalez-Mullier, I., Gros, C.: LEXTER, a Natural Language Tool for Terminology Extraction. In: 7th EURALEX Intl. Congress on Lexicography, Part II, Goteborg University, Goteborg, Sweden (1996) 771-779

[Bourigault & Jacquemin, 1999] Bourigault, D. and C. Jacquemin. 1999. "Term extraction + term clustering: an integrated platform for computer-aided terminology." In Proceedings of the 8th Conference of the European Chapter of the Association for Computational Linguistics, Bergen, Norway, 15-22

[Brewster et al., 2004] Brewster, C., Alani, H., Dasmahapatra, S. and Wilks, Y. Data driven ontology evaluation. In Proceedings of International Conference on Language Resources and Evaluation, Lisbon, Portugal (2004).

[Caraballo, 1999] Caraballo S. (1999) Automatic construction of a hypernymlabeled noun hierarchy from text. in Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL), pages 120-126.

[Cederberg & Widdows, 2003] Cederberg, S. and Widdows, D. (2003). Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In Conference on Natural Language Learning (CoNNL), pages 111-118.

[Ciaramita et al., 2005] Ciaramita, M., Gangemi, A., Ratsch, E., Saric, J., and Rojas, I. (2005). Unsupervised learning of semantic relations between concepts of

a molecular biology ontology. In Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI), pages 659-664.

[Cimiano et al., 2005] Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. Philipp Cimiano and Andreas Hotho and Steffen Staab Journal of Artificial Intelligence Research (JAIR) 24 305-339 (2005)

[Cimiano & Volker, 2005] Cimiano P, Volker J. Towards large-scale, open-domain and ontology-based named entity classification. Proc Intl Conf Recent. 2005 .166 72.

[Cimiano et al., 2006] Philipp Cimiano, Matthias Hartung, E. Ratsch. Finding the Appropriate Generalization Level for Binary Relations Extracted from the Genia Corpus. In Proceedings of the International Conference on Language Resources and Evaluation (LREC), pp. 161-169. ELRA, May 2006.

[Cimiano et al., 2006] P. Cimiano, J. Volker, and R. Studer. Ontologies on Demand? A Description of the State-of-the-Art, Applications, Challenges and Trends for Ontology Learning from Text. In Information, Wissenschaft und Praxis 57 (6-7): 315-320. October 2006.

[Cimiano et al., 2005] P. Cimiano and S. Staab. Learning concept hierarchies from text with a guided hierarchical clustering algorithm. In C. Biemann and G. Paas, editors, Proceedings of the ICML 2005 Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods, Bonn, Germany, 2005.

[Cimiano & Volker, 2005] Philipp Cimiano and Johanna Volker. Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB), 3513: 227-238, 2005.

[Claveau et al., 2003] Claveau V., Sebillot P., Fabre C. and Bouillon, P. (2003). Learning semantic lexicons from a part-of-speech and semantically tagged cor-

pus using inductive logic programming. Journal of Machine Learning Research, 493-525.

[Clark & Weir, 2002] Clark, S. and Weir, D. (2002). Class-based probability estimation using a semantic hierarchy. Computational Linguistics, 187-206.

[Dagan I. , Church K., 1995] Dagan I. , Church K. : Termight: Identifying and translating technical terminology. In: Proc. 7th Conference of the european Chapter of the association for Computational Linguistics, 1995. EACL'95 , pp 34-40

[DAML+OIL, 2001] Horrocks I, van Harmelen F (eds) (2001) Reference Description of the DAML+OIL (March 2001) Ontology Markup Language. Technical Report. http://www.daml.org/2001/03/reference.html

[Downey et al., 2004] Downey, D., Etzioni, O., Soderland, S., and Weld, W. (2004). Learning text patterns for web information extraction and assessment. In Proceedings of the AAAI Workshop on Adaptive Text Extraction and Mining (ATEM).

[Faure & Nedellec, 1999] Faure, D and Nedellec, C, 1999, Knowledge acquisition of predicate argument structures from technical texts using machine learning: The system ASIUM in D Fensel and R Studer (eds.) Proceedings of the 11th European Workshop on Knowledge Acquisition, Modelling and Management (EKAW99), Dagstuhl, Germany (Lecture Notes in Artificial Intelligence, 1621). Berlin: Springer, pp. 329334.

[Faure et al., 2000] Faure D, Poibeau T (2000) First experiments of using semantic knowledge learned by ASIUM for information extraction task using INTEX. In: S. Staab, A. Maedche, C. Nedellec, P. Wiemer-Hastings (eds.), Proceedings of the Workshop on Ontology Learning, 14th European Conference on Artificial Intelligence ECAI00, Berlin, Germany

[Frantzi, Ananiadou, Mima, 2000] Frantzi, K., Ananiadou, S. Mima, H. (2000) Automatic recognition of multi-word terms. International Journal of Digital Libraries 3(2), Special issue edited by Nikolau, C. Stephanidis, C. (eds.), 117132

[Gamallo et al., 2002] Gamallo, P., Gonzalez, M., Agustini, A., Lopes, G., and de Lima, V. S. (2002). Mapping syntactic dependencies onto semantic relations. In Proceedings of the ECAI Workshop on Machine Learning and Natural Language Processing for Ontology Engineering, pages 15-22.

[Ganter & Reuter, 1991] B. Ganter, K. Reuter, Finding all closed sets: A general approach, in: Order, Kluwer Academic Publishers, Amsterdam, 1991, pp. 283-290.

[Ganter et al., 1999] Ganter, B. and Wille, R.: Formal Concept Analysis, Mathematical Foundations, Springer, (1999)

[Ganter et al., 1999] G anter, Bernhard; Stumme, Gerd; Wille, Rudolf, eds. (2005), Formal Concept Analysis: Foundations and Applications, Lecture Notes in Artificial Intelligence, no. 3626, Springer-Verlag, ISBN 3-540-27891-5

[Gruber, 1993] Tom Gruber - A translation approach to portable ontology specifications. Knowledge Acquisition, 5:199-220, 1993.

[Gulla & Brasethvik, 2008] Jon Atle Gulla, Terje Brasethvik: A Hybrid Approach to Ontology Relationship Learning. NLDB 2008: 79-90

[Haav, 2003] Haav, H.-M. (2003). An application of inductive concept analysis to construction of domain-specific ontologies. In Proceedings of the VLDB Preconference Workshop on Emerging Database Research in East Europe.

[Harris, 1968] Harris, Z. (1968). Mathematical Structures of Language. Wiley.

[Hatzivassiloglou, 1996] Hatzivassiloglou V. Do we need linguistics when we have statistics? A comparative analysis of the contributions of linguistic cues to a statistical word grouping system. In: Klavans JL, Resnik P (editors). The

Balancing Act: Combining Symbolic and Statistical Approaches to Language, pp. 6794, Cambridge(MA): MIT Press; 1996.

[Hearst, 1992] Hearst, M. (1992). Automatic acquisition of hyponyms from large text corpora. In Proceedings of the 14th International Conference on Computational Linguistics (COLING), pages 539-545.

[Hersh et al., 1994] Hersh, W., Buckley, C, Leone, T., and Hickam, D. (1994). Ohsumed: An interactive retrieval evaluation and new large text collection for research. In Proceedings of the 17th ACM SIGIR Conference on Research and Development in Information Retrieval, pages 192-201.

[Hersh & Hickam, 1994] Hersh WR, Hickam DH, Use of a multi-application computer workstation in a clinical setting, Bulletin of the Medical Library Association, 1994, 82: 382-389.

[Hindle, 1990] Hindle, D. (1990). Noun classification from predicate-argument structures. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), pages 268-275.

[Horrocks & Peter Patel-Schneider, 2004] Ian Horrocks and Peter Patel-Schneider. Reducing OWL entailment to description logic satisfiability. J. of Web Semantics, 1(4):345-357, 2004.

[Iwanska et al., 2000] Iwanska, L., Mata, N., and Kruger, K. (2000). Fully automatic acquisition of taxonomic knowledge from large corpora of texts. In Iwanksa, L. and Shapiro, S., editors, Natural Language Processing and Knowledge Processing, pages 335-345. MIT/AAAI Press.

[Jacquemin, 2001] Jacquemin, C. 2001, Spotting and Discovering Terms through NLP. Cambridge MA: MIT Press

[Jian Wang & Keqing He, 2006] Jian Wang, Keqing He, "Towards Representing FCA-based Ontologies in Semantic Web Rule Language," cit,pp.41, Sixth IEEE International Conference on Computer and Information Technology (CIT'06), 2006

[Justeson, Katz, 1995] Justeson, J.S., Katz,S.M : Technical terminology : some linguistic properties and an algorithm for identification in text. Natural Language Engineering 1(1):9-27,1995

[Kavalec et al, 2004] Kavalec, M., Maedche, A., Skatek, V.: Discovery of Lexical Entries for Non-taxonomic Relations in Ontology Learning. In: SOFSEM04. Volume 2932 of LNCS. (2004) 249256

[KEA, 1999] Frank E., Paynter G.W., Witten I.H., Gutwin C. and Nevill-Manning C.G. (1999) Domain specific keyphrase extraction Proc. Sixteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers, San Francisco, CA, pp. 668-673.

[Knublauch, 2003] Knublauch H. An AI Tool for the Real World: Knowledge Modeling with Protg JavaWorld, June 20, 2003.

[Landauer & Dumais, 1997] Landauer, T. K. & Dumais, S. T. (1997). A solution to Plato's problem: The Latent Semanctic Analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review , 104 , 211-140

[Landauer et al., 1998] Landauer, T. K., Foltz, P. W., & Laham, D. (1998). Introduction to Latent Semantic Analysis. Discourse Processes, 25, 259-284.

[Lavrac & Dzeroski, 1994] Lavrac, N. and Dzeroski, S. (1994). Inductive Logic Programming: Techniques and Applications. Ellis Horwood.

[Madche & Staab, 2000] Madche, A. and Staab, S. (2000). Discovering conceptual relations from text. In Proceedings of the 14th European Conference on Artificial Intelligence (ECAI), pages 321-325.

[Madche & Staab, 2000] Maedche, A., Staab, S.: Semi-automatic Engineering of Ontologies from Text. In: Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering, 2000.

[Madche & Staab, 2002] Madche, A. and Staab, S. (2002). Measuring similarity between ontologies. In Proceedings of the European Conference on Knowledge Acquisition and Management (EKAW), pages 251-263.

[Madche, 2002] Madche A. Ontology learning for the semantic web. Kluwer Academic Publishers, 2002

[Manning, Schutze, 1999] Foundations of statistical natural language processing . Christopher D. Manning, Hinrich Schutze. Cambridge, Mass. : MIT Press, c1999.

[Miller, 1995] WordNet: A lexical database for English. Communications of the ACM 38(11) 3941.

[Milios et al., 2003] E.Milios, Y.Zhang, B.He, L.Dong: Automatic term extraction and document similarity in special text corpora. Proceedings of the 6th conference of the Pacific Association for Computational Linguistics (PACLing'03), Halifax,Nova Scotia, Canada , August 22-25, 2003 , pages 275-284

[Navigli & Velardi, 2004] Navigli, R. and Velardi, P. (2004). Learning domain ontologies from document warehouses and dedicated websites. Computational Linguistics, 30(2):151-179.

[Nenadic et al., 2004] Nenadic, G., Spasic, I. , Ananiadou, S. (2004) Automatic Discovery of Term Similarities Using Pattern Mining, in International Journal ofTerminology.10:1, 55-80

[Nirenburg & Raskin, 2004] Nirenburg, S. and Raskin, V. (2004). Ontological Semantics. MIT Press.

[Noy&McGuinness, Stanford] N. Noy, D. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology, Stanford.

[OIL, 2000] Horrocks I, Fensel D, Harmelen F, Decker S, Erdmann M, Klein M (2000) OIL in a Nutshell. In: Dieng R, Corby O (eds.) 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW00).

Juan-Les-Pins, France. (Lecture Notes in Artificial Intelligence LNAI 1937) Springer-Verlag, Berlin, Germany, pp. 116

[OWL, 2004] OWL Web Ontology Language, 2004. http://www.w3.org/TR/owl-features/

[OWL Requirements Document, 2004] OWL Requirements Document, 2004. http://www.w3.org/TR/2004/REC-webont-req-20040210/

[Paolucci et al., 2002] Paolucci, M., Kawmura, T., Payne, T., and Sycara, K. (2002). Semantic matching of web services capabilities. In Proceedings of the 1st International. Semantic Web Conference (ISWC).

[Pinto & Martins, 2004] Pinto, H. and Martins, J. (2004). Ontologies: How can they be built? Knowledge and Information Systems, 6(4):441-464.

[Pustejovsky, 1995] Pustejovsky James, 1995. The Generative Lexicon. Cambridge MIT Press

[Ratnaparkhi, 1996] A. Ratnaparkhi, A maximum entropy model for part-of-speech tagging. In Proceedings of Conference on Empirical Methods in Natural Language Processing, University of Pennsylvania, 1996.

[Ravichandran & Hovy, 2002] Ravichandran, D. and Hovy, E. (2002). Learning surface text patterns for a question answering system. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), pages 41-47.

[Resnik, 1997] Resnik, P. (1997). Selectional preference and sense disambiguation. In Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?

[RDF, 2002] Lassila O, Swick R (1999) Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation. http://www.w3.org/TR/REC-rdf-syntax/

[Sager, 1990] Juan C. Sager. A Practical Course in Terminology Processing John Benjamins Publishing Company, 1990.

[Sabou, 2005] Sabou, M. (2005). Learning web service ontologies: an automatic extraction method and its evaluation. In Buitelaar, P., Cimiano, P., and Magnini, B., editors. Ontology Learning from Text: Methods, Applications and Evaluation, Frontiers in Artificial Intelligence and Applications, pages 125-139. lOS Press.

[Sanchez & Moreno, 2004] Sanchez, D. and Moreno, A. (2004a). Automatic generation of taxonomies from the WWW. In Proceedings of the Conference on Practical Aspects of Knowledge Management (PAKM)

[Sanchez & Moreno, 2005] Sanchez, D. and Moreno, A. (2005). Web-scale taxonomy learning. In Biemann, C. and Pass, G., editors. Proceedings of the Workshop on Extending and Learning Lexical Ontologies using Machine Learning Methods.

[Scheffer, 2001] T.Scheffer (2001). Finding Association Rules that Trade Support Optimally against Confidence. Proc of the 5th European COnf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'01), pp. 424-435. Freiburg, Germany: Springer-Verlag

[Schutz & Buitelaar, 2005] Schutz, A. and Buitelaar, P. (2005). RelExt: A tool for relation extraction from text in ontology extension. In Proceedings of the International Semantic Web Conference, pages 593-606.

[Schutze, 1993] Schutze, H. (1993). Word space. In Advances in Neural Information Processing Systems 5, pages 895-902.

[Sirin et al., 2002] Sirin, E., Hendler, J., and Parsia, B. (2002). Semi-automatic composition of web services using semantic descriptions. In Proceedings of the ICEIS Workshop on Web Services: Modeling, Architecture and Infrastructure.

[Srikant & Agrawal, 1995] R. Srikant and R. Agrawal, Mining generalized association rules, in Proc. of VLDB 95, pp. 407419, (1995).

[Steinbach, Karypis & Kumar, 2000] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In KDD Workshop on Text Mining, 2000.

[Tegos et al, 2008] A. Tegos, V. Karkaletsis, and A. Potamianos, "Learning of Semantic Relations Between Ontology Concepts Using Statistical Techniques", in Proc. HLIE Workshop , Antwerp, Belgium, Sept. 2008.

[Velardi et al., 2002] Velardi, P, Navigli, R and Missikoff, M, 2002, 'Integrated approach for Web ontology learning and engineering' IEEE Computer 35(11) 6063.

[Wille, 1982] Wille, R. (1982). Restructuring lattice theory: an approach based on hierarchies of concepts. Ordered Sets, pages 445-470.

[World Wide Web Consortium (W3C)] World Wide Web Consortium - Web Standards. http://www.w3.org/

[Yamaguchi, 2001] Yamaguchi, T. (2001). Acquring conceptual relationships from domain-specific texts. In Proceedings of the IJCAI Workshop on Ontology Learning.

Appendix A

Sample Results

Table A.1: Top Extracted Terms from C/NC-Value - Computer science corpus

output term	C-Value
web page	1740.11
information retrieval	1274.14
search engine	1103.99
machine learning	727.70
computer science	723.82
experimental result	655.125
text mining	645.57
natural language processing	582.83
world wide web	557.33
large number	530.67
artificial intelligence	515.73
relevant document	468.22
similarity measure	464.64
information extraction	443.29
knowledge discovery	435.79
entity class	426.75
web site	409.43
support vector machine	376.11
xml document	375.41
small number	364.0
search result	359.91
clustering algorithm	355.98
relevant page	353.81
web document	350.61
information system	324.94
computational linguistics	324.30
technical report	320.55

Table A.2: Top Extracted Terms from C/NC-Value - OhsuMed corpus

output term	C-Value
heart rate	66.58
blood pressure	63.575
magnetic resonance imaging	44.37
coronary artery disease	34.60
normal subject	34.0
extracorporeal shock wave lithotripsy	33.5
case report	31.0
carotid endarterectomy	31.0
prospective study	29.0
plasma concentration	28.0
significant change	26.0
myocardial infarction	23.0
risk factor	22.789
blood flow	22.745
congestive heart failure	21.661
arterial pressure	21.333
tricuspid regurgitation	21.166
surgical treatment	21.0
acute myocardial infarction	20.604
mean age	20.0
visual acuity	20.0
mast cell	19.90
renal failure	18.3
severe congestive heart failure	17.43
sudden cardiac death	17.43
systolic blood pressure	17.03
radiation therapy	16.666

Table A.3: Sample relations extracted from computer science corpus

information retrieval ***taxonomic relations***: (retrieval, modern information retrieval, cross information retrieval, research in information retrieval, web information retrieval, intelligent information retrieval, information retrieval community, information retrieval system, development in information retrieval) ***non-taxonomic relations (domain, range, label)***: search engine, information retrieval, search information retrieval, document matrix, use information retrieval, text document, analyze information retrieval, query term, establish
search engine ***taxonomic relations***: (search, search engine form, search engine google, search engine index, search engine result, web search engine, web search engine result) ***non-taxonomic relations (domain, range, label)***: user query, search engine, submit search engine, relevant page, return information retrieval, search engine, use search engine, relevant information, find search engine, linkage information, use
machine learning ***taxonomic relations***: (learning, program for machine learning, machine learning technique, machine learning algorithm, practical machine learning tool) ***non-taxonomic relations (domain, range, label)***: automatic image annotation, machine learning method, use machine learning, tagged corpus, use machine learning, information access, has