NEALT MONOGRAPH SERIES
VOL. 1

Cross-language Ontology Learning

Incorporating and Exploiting Cross-language Data in the Ontology Learning Process

Hans Hjelm

NORTHERN EUROPEAN ASSOCIATION FOR LANGUAGE TECHNOLOGY

NEALT Monograph Series, Vol.1

Dissertation for the degree of Doctor of Philosophy (PhD)
Defended on February, 6, 2009 in Stockholm University

ISSN 1736-6291

Published by
Northern European Association for Language Technology (NEALT)
http://omilia.uio.no/nealt

Electronically published at
Tartu University Library (Estonia)
http://dspace.utlib.ee/dspace/handle/10062/10126

Cross-language Ontology Learning

Incorporating and Exploiting Cross-language Data in the Ontology Learning Process

Hans Hjelm

Stockholm University

ISBN 978-91-7155-806-0

Printed in Sweden by US-AB, Stockholm 2009

Distributor: Department of Linguistics, Stockholm University

Abstract

An ontology is a knowledge-representation structure, where words, terms or concepts are defined by their mutual hierarchical relations. Ontologies are becoming ever more prevalent in the world of natural language processing, where we currently see a tendency towards using semantics for solving a variety of tasks, particularly tasks related to information access. Ontologies, taxonomies and thesauri (all related notions) are also used in various variants by humans, to standardize business transactions or for finding conceptual relations between terms in, e.g., the medical domain.

The acquisition of machine-readable, domain-specific semantic knowledge is time consuming and prone to inconsistencies. The field of ontology learning therefore provides tools for automating the construction of domain ontologies (ontologies describing the entities and relations within a particular field of interest), by analyzing large quantities of domain-specific texts.

This thesis studies three main topics within the field of ontology learning. First, we examine which sources of information are useful within an ontology learning system and how the information sources can be combined effectively. Secondly, we do this with a special focus on cross-language text collections, to see if we can learn more from studying several languages at once, than we can from a single-language text collection. Finally, we investigate new approaches to formal and automatic evaluation of the quality of a learned ontology.

We demonstrate how to combine information sources from different languages and use them to train automatic classifiers to recognize lexico-semantic relations. The cross-language data is shown to have a positive effect on the quality of the learned ontologies. We also give theoretical and experimental results, showing that our ontology evaluation method is a good complement to and in some aspects improves on the evaluation measures in use today.

Sammanfattning

En ontologi är en struktur som används för kunskapsrepresentation i maskinläsbart format. Ontologier innehåller koncept, termer eller ord som definieras genom inbördes hierarkiska relationer. De ges en alltmer framskjutande roll inom dagens språkteknologi, där vi ser tendenser till att integrera semantik i ett allt större antal applikationer, främst inom områden relaterade till informationsåtkomst. Ontologier, taxonomier och tesaurer (alla starkt förknippade med varandra) används också av människor för att exempelvis standardisera affärsprocesser eller för att hitta relationer mellan koncept eller termer inom medicinen.

Att skapa maskinläsbara, domänspecifika, semantiska resurser är en tidsödande och krävande process, där det är svårt för människor att alltid fatta konsekventa beslut. Därför försöker man inom fältet för ontologiinlärning utveckla verktyg för att automatisera skapandet av domänspecifika ontologier. Inlärningen görs huvudsakligen genom automatisk analys av stora samlingar domänspecifik text.

Denna avhandling studerar tre huvudsakliga frågor inom området för ontologiinlärning. För det första undersöker den vilka informationskällor som är användbara inom ett ontologiinlärningssystem, samt hur dessa källor kan kombineras på ett effektivt sätt. För det andra gör den detta med fokus på tvärspråkliga textsamlingar, för att se om det finns fördelar med att studera flera språk parallellt, jämfört med att bara använda ett språk i taget. För det tredje utforskar avhandlingen nya ansatser till formell automatisk utvärdering av kvaliteten hos en automatiskt inlärd ontologi.

Vi visar hur information från olika språk kan kombineras och användas av automatiska klassificerare för att känna igen lexikala semantiska relationer. Tvärspråkliga data visas ha en positiv effekt på kvaliteten hos de inlärda ontologierna. Vi ger också teoretiska och experimentella resultat där vår metod för utvärdering av inlärda ontologier visas vara ett gott komplement till, och i vissa fall förbättra, de utvärderingsmetoder som används idag.

Preface

During my first two years as a PhD student, I worked half-time to finance my studies. I worked mainly for Intrafind Software AG, based in Munich, Germany, and they showed a great deal of flexibility towards having me work out of Sweden and allowing me to work on my thesis on the side. I would especially like to thank Intrafind's Christoph Goller, also for valuable discussions and for commenting on the pre-final version of my thesis.

By my third year, I was able to start working full-time as a PhD student, thanks to a scholarship from the GSLT. I am glad to have been a part of the GSLT for many reasons, not the least of which has been the ability to meet so many nice and interesting people/colleagues. Thank you all – I don't dare to start making a list of people to thank, it would fill up the page! A scholarship from the DAAD (Deutsher Akademischer Austausch Dienst) enabled me to go to Saarbrücken for four months, as a guest researcher at the DFKI, under the knowing supervision of Paul Buitelaar. This was a great experience, during which I learned a lot and made many friends.

I have had the pleasure of sharing offices with some fantastic people here at Stockholm university: Henrik Oxhammar, Kristina Nilsson and Yvonne Samuelsson, all fellow PhD students. Thank you for all discussions and all coffee breaks; a special thanks to Henrik for continuously filling our white-board with obscure pictures of data structures. Another special thanks to Kina for many discussions regarding this thesis. I have also much enjoyed sharing offices with Magdalena Mikolajczyk, David Hagstrand, Britt Hartmann, Jörgen Aasa, Per Weijnitz and Joakim Lundborg, for varying lengths of time. I appriciate having had Jennifer Spenader, Sofia Gustafson-Capkova and Magnus Sahlgren as colleagues in the Stockholm CL-group.

Special thanks to all proof-readers: Kristina Nilsson, Britt Hartmann, Marina Santini and Yvonne Samuelsson. I really appreciate your help.

Tomas Einarsson has very patiently but enthusiastically explained various mathematical issues to me, I am very grateful for that. Per Sundqvist introduced me to MATLAB and talked to me at length about matrices. Christoph Schwarz, Mikael Parkvall and Eva Lindström have all kindly answered my questions on linguistic issues. Magnus Rosell taught me about clustering evaluation – much appreciated!

My two supervisors, Martin Volk and Joakim Nivre both deserve special mention. Martin has kept me on track during this process, making sure I focused on the right things, in the right order. I appreciate him giving me the chance to start working as a PhD student, even though I did not have official funding. Joakim has been invaluable to me in helping me sort out various sta-

tistical matters, among other things. Both have provided me with quick and precise feedback during the writing of my thesis. For this I am truly thankful.

Most importantly, I want to thank my family. You have been a constant moral support during these years, I am very happy to have you.

Contents

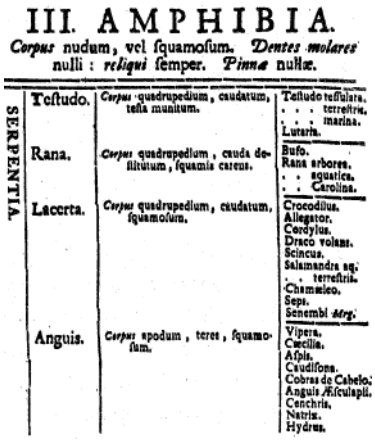
1. Introduction

> . . . a perfect notation would be a substitute for thought. (Russell, 1961, Introduction, p. xix)

Think of the frustrating experience of running into circular definitions when looking up a word in a dictionary. Yet – how do we convey the meaning of a word without referring to meanings of other words? This line of reasoning has been summarized concisely by linguist/computer scientist Yorick Wilks: "Meaning. . . is best thought of as other words. . . " (Wilks, 1999, p. 83) In ontologies – the object of study in this thesis – this is exactly what we find: words (or concepts) defined by their position in a hierarchy of other words and the relations that hold between them.

Dating back to Aristotle, through 18th century botanist Carl Linnaeus and to present day Internet pioneer Tim Berners-Lee (2005), scientists and philosophers have sought to organize their knowledge of the world in hierarchal structures (see Fig. 1.1). As can be seen in the success of, e.g., object-oriented programming languages and the Unified Modeling Language, where hierarchical structures play an important part, this way of organizing knowledge also lends itself for implementation in computer systems. Whereas the object hierarchy in a computer program models data relevant to the system internally, an ontology is typically a way of representing *real-world knowledge* in a formal manner (processable by humans and machines alike).

There is a consensus in the world of natural language processing (NLP) that we have reached a point where we need to incorporate meaning, *semantics*, in the NLP systems of today, in order to take the next big qualitative step, but also in order to increase, e.g., usability. But the acquisition of machine-readable semantic resources is expensive in terms of human effort and therefore also expensive financially. This thesis aims at exploring ways to improve the quality of one type of automated knowledge-acquisition systems, known as *ontology learning systems*, in an effort to tackle the task of computerizing the acquisition of semantic knowledge.



III. AMPHIBIA.

Corpus nudum, vel ſquamoſum. *Dentes molares* nulli : *reliqui* ſemper. *Pinnæ* nullæ.

SERPENTIA.

Teſtudo.	*Corpus* quadrupedium, caudatum, teſta munitum.	Teſtudo teſſulata. . . . terreſtris. . . . marina. Lutaria.
Rana.	*Corpus* quadrupedium, cauda deſtitutum, ſquamis carens.	Bufo. Rana arborea. . . aquatica. . . Carolina.
Lacerta.	*Corpus* quadrupedium, caudatum, ſquamoſum.	Crocodilus. Allegator. Corylus. Draco volans. Scincus. Salamandra aq. . . terreſtris. Chamæleo. Seps. Senembi *Amer.*
Anguis.	*Corpus* apodum, teres, ſquamoſum.	Vipera. Cæcilia. Aſpis. Caudiſona. Cobras de Cabelo. Anguis Æſculapii. Cenchris. Natrix. Hydrus.

Figure 1.1: Linnaeus hierarchical categorization of amphibians. Picture (out-take) from Wikimedia commons, taken from Linnaeus (1939).

1.1 Ontologies

> The ontology in ontological semantics is the next best thing to being able to refer to the outside world directly. (Nirenburg and Raskin, 2004, p. 88)

Ontology as a field of study within philosophy deals with questions about the basic entities of *existence*: what the nature of those entities is and how they can be grouped or subdivided based on their respective traits and qualities (see Quine, 1969). Though not completely unrelated, ontology within information science refers to a way of representing knowledge or structuring the terminology within a domain. Chandrasekaran et al. (1999) define ontologies as "content theories about the sorts of *objects*, *properties* of objects, and *relations* between objects that are possible in a specified domain of knowledge" (my emphasis).

Sowa (1999) distinguishes between three main types of ontologies: *formal ontologies*, *prototype-based ontologies* and *terminological ontologies*. Formal ontologies are widespread within the artificial intelligence and knowledge representation fields, where they are vehicles for reasoning about concepts and relations between concepts. An example of such an ontology is shown in Fig. 1.2. The prototype-based ontology typically appears as the result of applying a hierarchical clustering technique to distributional data, a process which we describe in Sect. 2.5.1. Its concepts are defined by enumerating the (prototypical) concept members (see an example in Fig. 1.3). Finally the terminological ontologies are structured like the formal ones, but they order terms, or groups of terms, rather than concepts (see Fig. 1.4). Terminological ontologies are often referred to as *taxonomies*. This thesis deals with terminological- and prototype-based ontologies; formal ontologies are dealt

with implicitly, because terminological ontologies often function as backbones when building formal ontologies.

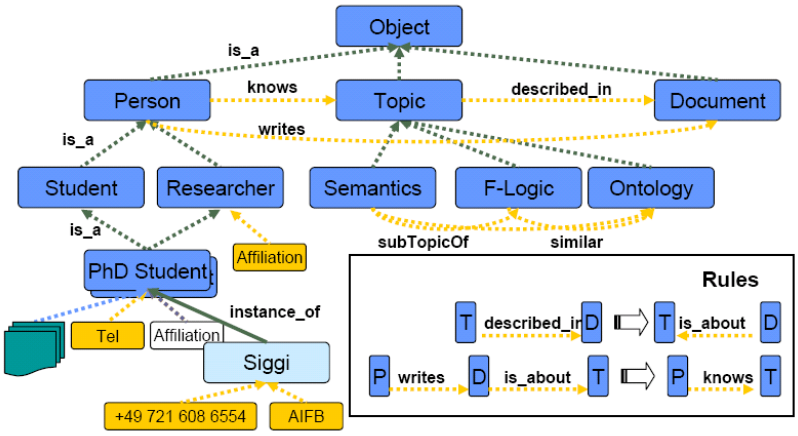


Figure 1.2: Formal ontology. Picture from tutorial *Ontology Learning from Text* at ECML/PKDD 2005, Porto, Portugal, by Paul Buitelaar, Philipp Cimiano, Marko Grobelnik and Michael Sintek.

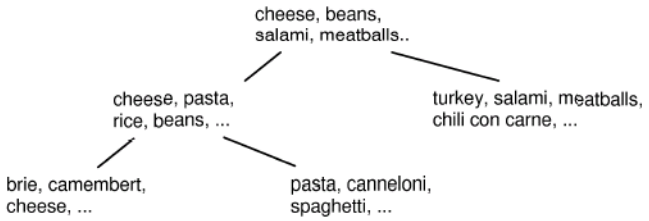


Figure 1.3: Prototype-based ontology, adapted from Biemann (2005).

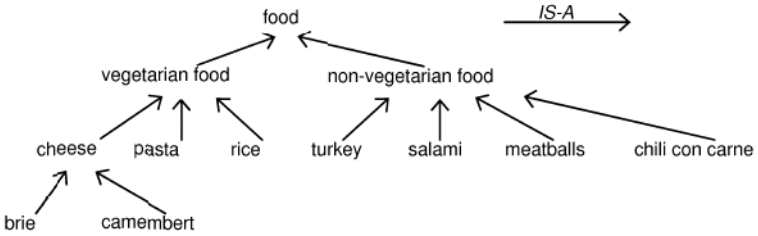


Figure 1.4: Terminological ontology, adapted from Biemann (2005).

Formal ontologies have much in common with schemas used in object-oriented database systems (see Connolly et al., 1998, for a description of such systems). Database schemas exist at different levels of abstraction. The *con-*

ceptual schemas define the objects important to a particular database application, along with the relevant relations that can hold between these objects. The conceptual schemas allow for inheritance, but this is not supported by all database management systems. A conceptual schema can, e.g., define a TRANSACTION object, which involves a PERSON as a seller and buyer, a PRICE to be paid, and a PRODUCT to switch OWNER. The objects listed here could be represented by concepts in the ontology, and the actions (broken down into actions involving pairs of objects) could be represented as relations in the ontology.

Another way of categorizing ontologies is given in Maedche (2002): *top-level*, *domain*, *task* and *application ontologies*. This is a grouping of ontologies where the focus lies on their degrees of generality. Top-level ontologies are the most general kind, containing concepts related to, e.g., time (day, hour, minute) or space (length, volume). A domain ontology refines the concepts in a top-level ontology by focusing on a particular domain, whereas a task ontology does the same but focuses on a given task. The application ontology is viewed as a further specialization of both the domain and the task ontology, with the aim of customizing the ontology for use in a particular application – close to the conceptual database schema just discussed. The relations are schematically displayed in Fig. 1.5. This thesis focuses on the learning of domain ontologies – top-level ontologies do not need to be learned because of their general nature, and task- and application ontologies lack the wider range of use of the domain ontology.

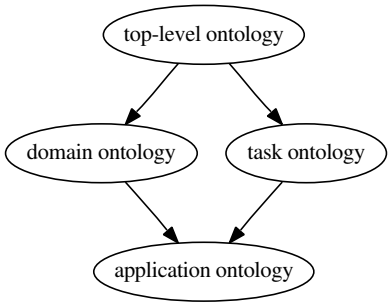


Figure 1.5: Relations between ontology types, adapted from Maedche (2002).

In formal ontologies, sharp distinctions are made between the ontology and the lexicon, whereas this is not true in terminological ontologies. The formal ontology contains concepts and relations encoding world knowledge, but by itself does not contain lexical knowledge (Nirenburg and Raskin, 2004). If we want to add lexical knowledge, this is done via a lexicon (more than one lexicon is needed if we want to deal with more than one language). The connections between ontology and lexicon are made via mappings between the two. In Nirenburg and Raskin (2004), the authors distinguish between ontology, lexicon, fact repository and onomasticon (a repository of names). Only the

lexicon and onomasticon are language specific. 'John Kennedy' is the name (stored in the onomasticon) of an instance (in the fact repository) of the concept labeled US-PRESIDENT, which is stored in the ontology and connected to 'US-president' and 'president of the USA' in the lexicon. The concepts in the formal ontology can be said to form the top of the semiotic triangle, connecting *symbols* (words or terms) with *referents* (real life objects, processes or phenomena) (Ogden and Richards, 1923).

This thesis limits its scope to studying *is-a* hierarchies. Other relations, such as the *part-of* relation, are also capable of forming hierarchies, but they do not ensure inheritance to the same degree as is-a relations. Cruse (1986) gives the example of a hand being a part of an arm and a finger being part of a hand, but that it is less clear if a finger can be said to be part of an arm. As discussed in the introductory section, the is-a relation is a widespread and fundamental relation, prevalent in society in general as well as in scientific contexts, which together motivates it being our principal relation of study throughout this thesis (discussed further in Sect. 2.1).

1.2 Ontology Learning

Ontology learning has been defined as "the acquisition of a domain model from data" (Cimiano, 2006, p. 19). This section describes the basic principles involved in ontology learning in accordance with this definition and motivates the automation of the ontology learning process. The ontology learning field is very heterogeneous, both in terms of the methods used and in terms of the results sought. An overview of the different approaches is given in Sect. 2.5.

Given a collection of texts, the ontology learning task typically consists of first recognizing the relevant objects (words or terms) in the text collection and secondly ordering these into a hierarchical inheritance structure. In Maedche et al. (2003), this is referred to as "learning the taxonomic backbone of ontologies", or "ontology learning part one". The result of such a learning process is thus a terminological or prototype-based ontology (see Sect. 1.1), which could then be used "as is" or be further embellished and transformed into a formal ontology.

What would make us want to learn an ontology automatically? Can't we get by with WordNet (Fellbaum, 1998) or the public domain version of Roget's Thesaurus?[1] There turns out to be two main disadvantages with using these kinds of handmade, general content resources. The first is their lack of coverage; when working with text from a particular domain, chances are that a large part of the vocabulary will not be covered by any handcrafted resource. Roark and Charniak (1998) report that when using their semantic class learner system to add domain-specific information to WordNet, over 60% of

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

the (valid) terms suggested by their system did not occur in WordNet (this is most likely even more pronounced for the previously mentioned version of Roget's Thesaurus, given its being published almost 100 years ago). Related to this is that we cannot be sure that words in WordNet will have the same meaning in the specialized domain as they have in general discourse.

The second major problem for handcrafted resources is their lack of adaptability. In everyday language, but also in technical domains, new words and terms are constantly being added to the existing vocabulary, and others shift meaning or fall out of use. Language, also when speaking of terminology and non-fiction texts, exhibits plasticity. This means that, for a handcrafted resource, changes need to be monitored and the taxonomical structure has to be updated on a regular basis – an effort which would be much reduced if automatic tools for updating or even rebuilding the resource were available. Learning the ontology from text data also has the added advantage of tackling the previously mentioned problem of coverage: the terms in the ontology will coincide with the ones actually appearing in the texts.

The task of an ontology learning system is to automate the process of ontology construction. Note that we do not require the process to be completely automated – a partial automation is better than none, provided that the quality of the system output is high enough.

A ubiquitous raw material for ontology learning is natural language text: in the form of web documents, reports, corporate policies, documentation, periodicals etc. If we can tap into this wealth of data and use it in the automation process, much will be won. This is a major reason why this thesis will place a large focus on unrestricted text and refrain from using hand-edited resources such as WordNet; resources where humans have already done part of the job.

One enticing aspect of using nothing but the text itself as input to our system is that our results can be seen as posts on the road to a much broader goal: the automated semantic interpretation of language.

1.3 Research Questions

This thesis will try to answer a set of questions regarding the automatic learning of ontologies and issues related to this topic. The questions deal with different aspects in the field:

1. **Sources of information**: What types of information are useful for solving the ontology learning task? How can they be combined in an effective way? This subject is dealt with mainly in Sects. 6.2 and 6.4.
2. **Cross-language aspects**: Can cross-language data teach us more than data from a single language, for the ontology learning task? If so, can cross-language equivalency relations be extracted automatically, with high enough accuracy to be of use, or do we need to rely on handcrafted

resources? We perform a variety of experiments in Chaps. 5–6 to address these questions.
3. **Evaluation**: How can we measure the quality of a learned ontology? Can we improve or complement the evaluation measures used today? We investigate this in Chap. 4.

We will be returning to the questions again in the concluding chapter, to discuss our results.

1.4 Thesis Outline

In this chapter we have tried to give an overview of the topic and to motivate to the reader why we consider ontology learning a problem worth solving. The rest of this thesis is structured as follows:

Chapter 2
Ontology Learning Perspectives

We take a closer look at the ontology learning problem: what it entails and how it has been approached in the past. We also look at cross-language issues as they relate to ontologies and questions regarding equivalence and translation.

Chapter 3
Resources

A number of language resources are needed in order for us to test the validity of our theories, from software to corpora and ontologies used as gold standards. We present the ones used in this thesis in this chapter.

Chapter 4
Theoretical and Experimental Investigations Regarding Evaluation

To be able to measure the effectiveness of our proposed methods, we need standardized ways of evaluating our results. This chapter reviews existing measures and also investigates the use of a proposed new evaluation measure.

Chapter 5
Experiments with Identifying Cross-language Term Equivalency

Towards the goal of incorporating cross-language lexical information in ontologies, we here investigate methods for automatically identifying equivalents (translations) among terms from different languages.

Chapter 6
Experiments in Ontology Learning

This chapter contains the bulk of our contribution to the field of ontology learning, with a focus on exploiting cross-language resources towards solving problems traditionally involving a single language.

Chapter 7
Conclusions

This chapter looks back to summarize what we have learned and also gives an outlook on the types of problems that could be tackled next.

2. Ontology Learning Perspectives

We start this chapter by giving a more in depth view of ontologies and how they are used by humans and machines. We discuss implications of including terms from more than one language in a single ontology and also take a closer look at what what constitutes a term. We introduce distributional similarity, a key concept for many approaches to ontology learning. These approaches are the main topic of the chapter; we give an extensive overview of how ontology learning has been tackled in the past. Finally, we discuss the identification of translational equivalents among terms, and how cross-language data has been exploited for solving a variety of NLP problems, in manners inaccessible to single language data.

The ontology learning task has a lot of closely related subfields in NLP: learning semantic networks, semantic clustering and automatic thesaurus extraction to name a few. Some of these fields are included in the following reviews, but we focus on an overview of various approaches to ontology learning itself in this chapter. Other survey literature for ontology learning can be found in Biemann (2005) and Cimiano (2006).

2.1 Primary Uses of Ontologies

Why do we need ontologies? There are two ways of motivating this: one is as a resource in their own right, used by humans, and the other is as a semantics-providing resource in an information system.

Examples of ontologies used by humans directly include the Common Procurement Vocabulary[1] (used for standardizing public procurement in the EU), Eurovoc (see Sect. 3.3.1) and, on a more general level, a great number of taxonomies and thesauri for looking up words in everyday life situations. Many companies also use taxonomies to structure their product lines, and web-shops typically group their articles hierarchically to make for easier navigation for the customers. This wide range of uses indicates that an ontology constitutes a valuable resource in and of itself. Additionally, as stated, they are commonly included in computer applications of various sorts in order to enhance the application quality; this is what we will look at in the rest of this section.

In NLP, we are often faced with problems involving synonymy or homonymy/polysemy. Consider the following expressions: ‘heart attack’,

[1] http://simap.europa.eu/codes-and-nomenclatures/codes-cpv_en.html

'myocardial infarction' and 'coronary thrombosis'. These are cases of (near) synonymy. A person submitting one of these expressions to a search engine would probably also be interested in documents containing any of the other expressions. Additionally, a person searching for information on 'heart disease' is likely to be interested in documents containing any of the three previously listed expressions. It is less clear that a person searching for 'arrhythmia' would be interested in documents dealing with 'heart attack' – here is where the hierarchical structure of an ontology can be put to use (see Fig. 2.1), separating hyponyms and synonyms from other notions of relatedness or similarity (cohyponymy in this case).

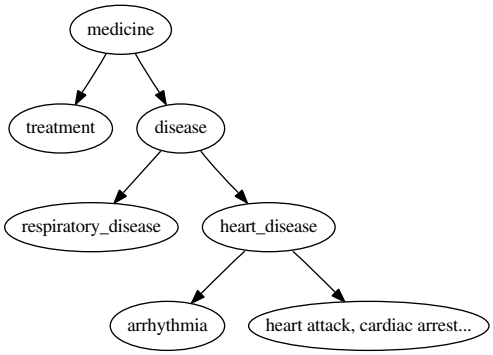


Figure 2.1: Toy medical ontology.

Homonymy and polysemy deal with the opposite problem, where the same expression can mean different things depending on the context where it is used. A 'bug' in a computer program is different from a bug in nature; this is an example of homonymy, modeled in Fig. 2.2. Using *concepts* (or nodes) in an ontology to refer to an object rather than using an expression like 'bug' lets us move away from the ambiguities of natural language and reduce the potential for misinterpretation.

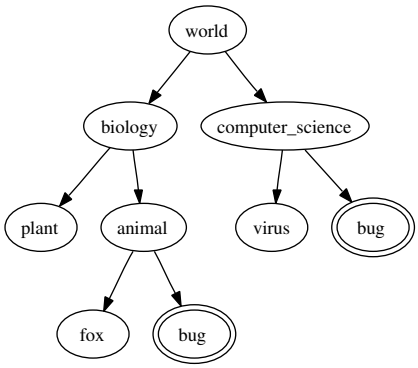


Figure 2.2: Toy general ontology.

We will now look at some NLP applications where the use of an ontology has proven beneficial. The examples all deal with *information access*, a subfield of NLP where semantics plays an important part.

Hotho et al. (2001) show positive results for including background knowledge from an ontology in a *text clustering* application. Apart from the improved results, the ontology can also be used as a means for explaining the clustering to the end user, via references to the ontology concepts and hierarchy.

Bloehdorn and Hotho (2004) demonstrate statistically significant improvements on a *text classification* task with the use of an ontology. They attribute the improved performance to the detection of multiword units in the text (a process also called *term spotting*) and to the fact that they include superconcepts in the document feature representation (e.g., if the word 'beef' occurs in the text, its hyperonym 'meat' is also added to the document representation). In Bloehdorn et al. (2006), the authors show the same thing but for an automatically learned prototype-based ontology. They also show improvements in some settings for the learned ontology over a domain-specific handcrafted ontology (the MeSH Tree Structures).[2] This may seem odd at first, but remember that the automatically learned ontology has the advantage of dealing with exactly the same terms that occur in the document collection, whereas this is not true for MeSH.

Makkonen et al. (2004) use a simple geographical ontology to see which events overlap geographically in a *topic detection and tracking* application. The hierarchy in a geographical ontology is typically not based on the is-a relation, which is our focus in this thesis, but on the part-of relation. Without the ontology, the application would have had to rely on string-matching to identify overlap, presumably resulting in higher precision but lower recall, although no such evaluation is carried out.

In Song et al. (2007), *query expansion* using WordNet (Fellbaum, 1998) is shown to have a beneficial effect on an information retrieval task, using TREC 5, 6 and 7 datasets.[3] This approach has been tried before without success; one of the differences here is that a word sense disambiguation technique is used to find the correct sense to use for expansion, which might explain why this set of experiments sees an improvement where previous approaches have not.

Liu and Chu (2007) also measure a positive effect from query expansion, using the OHSUMED testbed.[4] They start out by calculating a set of terms that are statistically related to the terms used in the query, using a *distributional similarity* measure (see Sect. 2.4) and *pseudo-relevance feedback* over a domain corpus. This term set is then filtered using the UMLS[5] ontology, so that only terms of the desired semantic types (i.e., children of a number

[2] http://www.nlm.nih.gov/mesh/intro_trees2005.html
[3] http://trec.nist.gov/
[4] http://ir.ohsu.edu/ohsumed/ohsumed.html
[5] http://www.nlm.nih.gov/research/umls/

of selected superconcepts) are added to the query. They show significant improvements on a number of different measures as compared to a system where no query expansion is performed.

The intention here is not to give a comprehensive list over all applications ever having profited from the use of an ontology; the mentioned articles are meant to give an idea of which types of problems can be addressed. For a complementary list of applications using ontologies, the reader is directed to Cimiano (2006, p. 281).

2.2 Cross-language Ontologies

> Their diversity [languages] is a diversity not of sounds and signs but of ways of looking at the world. (Kerényi, 1976, p. xxxi)

When we learn an ontology from a text, what is it that the final model captures? Is it "reality", a model of the world around us? Or maybe the representation of the world, as it appears in the writer's head? Perhaps an amalgamation of several representations, if the text has been written by several writers? If the text collection we are using as input to the system consists of texts in different languages, we might also ask if this fact has any further implications on the resulting model. On a related note, Goodman (1978) argues that just as we choose to acknowledge some patterns of stars as constellations, so other decisions are involved when we interpret some things as stars. In other words, the formation of concepts involves conscious decisions; the concepts are not given by nature once and for all. These decisions will not always look the same, for all languages and people (see Fig.2.3 below), which raises the question of how the differences will affect an ontology learning system. This section discusses what happens when we bring a cross-language perspective to ontologies, but we will not be able to answer the questions posed here until we have seen some experimental results (Chap. 6).

Since, as stated in Sect. 1.1, a formal ontology does not contain lexical knowledge, it is by nature language independent. We can add mappings between the ontology and arbitrarily many lexica in any number of languages. The result would be a *cross-language ontology*. Looking at terminological or prototype-based ontologies, the division between ontology and lexicon is weaker, since terms are used as labels and definitions simultaneously for the concepts. However, nothing stops us from using mappings from lexica in different languages to the hierarchical structure for these ontologies as well. The difference between the types of ontologies here lies on a theoretical level and has no practical consequences.

A cross-language ontology can be a useful resource when performing machine translation (Nirenburg and Raskin, 2004) or cross-language information

retrieval (Volk and Buitelaar, 2002). Apart from the usefulness of the resulting resource, we are here mainly interested in exploiting the cross-language information to *improve* on the results we get when learning an ontology from resources in a single language.

The field of *lexical typology* is concerned with the ways in which different languages "dissect" semantics, or meaning, and form it into words (see Koch, 2001). E.g., where English has 'sibling' as a unifying word for 'brother' and 'sister', French only has 'frère' and 'soeur' but no unifying word; how kinship relations are expressed varies greatly between the languages of the world (Koch, 2001). Another example is that English 'go' corresponds to both German 'gehen' (go by foot) and 'fahren' (go by some means of transportation) (Goddard, 2001). Fig. 2.3 also depicts how a particular *semantic field* is subject to different categorizations in different languages. We wish to investigate in this thesis whether this type of diversity will prove an asset in an ontology learning system or whether the different "views" will merely serve to clutter the meaning expressed through an isolated language. We will return to describe how such differences influence different features in our machine learning experiments in later parts of the thesis.

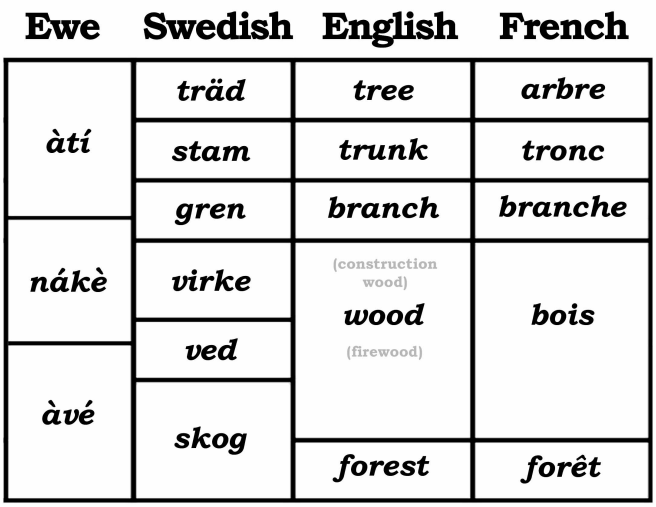


Figure 2.3: Words related to trees/wood in different languages. Ewe is a language in the Niger-Congo family and is spoken in Ghana, Togo and Benin. Note how English and French are closer to each other in this example than to Swedish, even though English and Swedish are both Germanic languages. (Figure from Mikael Parkvall, Stockholm University, personal communication.)

Other researchers have also considered the potential benefits from using more than one language when solving tasks traditionally involving a single language. Resnik (2004) notes that there is a potential for exploiting parallel texts for improving the accuracy in a number of NLP applications, including parsing and word sense disambiguation. According to Resnik, coupled with

each English sentence e, there is a meaning m_e, which is unobservable. Consider the French sentence f, a translation of e, with meaning m_f. Since the two sentences are translations of each other, we know that $m_e = m_f$ and we can consider the French sentence a semantic annotation of the English, and vice versa. This is akin to what we want to achieve in this thesis: exploiting parallelisms in order to learn a more accurate ontology.

According to Sager (1994), the notion of *equivalence* is central to the field of translation. Equivalence relations, in turn, have the properties of being reflexive (any word is a translation of itself), symmetrical (if A is a translation of B, then B is a translation of A) and transitive (if A is a translation of B, and B of C, then A is also a translation of C) (Boolos and Jeffrey, 1989). In practice, the notion of equivalence could be modified to a notion of relative equivalence (see Fig. 2.3), just as the synonymy relation is commonly relativized. Some philosophers have even argued that translation is in fact impossible between natural languages; see Quine (1960), where Quine gives his famous demonstration on what he calls the *indeterminacy of translation* (this applies to *radical translation*, where the translator learns a previously unknown language strictly from observing its use). It is not clear which, if any, implications such objections have on our current undertaking – we instead simply assume that translation is possible and that it works rather well in the vast majority of cases (especially in technical domains). The soundness of this assumption will partially be measured by the level of success of our experiments, as presented in later chapters.

In addition to lexico-typological aspects, there are two other obvious sources of discrepancy when dealing with translations that deserve to be mentioned. For a given word in the source language, it is not always possible to come up with a single translation to that word. This can have different causes, apart from the typological differences already discussed:

- The source word is polysemic or homonymic and the different senses of the word give rise to different translations in the target language.
- The target language has more than one word with more or less the same meaning, used interchangeably as translations of the source words (the target language words are synonyms).

Again, since we are dealing with terms, we expect situations as the ones mentioned to occur less frequently than they would in non-technical texts. Though problematic at times, we actually expect (in line with, e.g., Resnik mentioned above) that these discrepancies will be a source for added information when trying to learn an ontology over a domain, rather than just posing a problem.

Continuing with our example from Fig. 2.1, we could add some Swedish counterparts to the English synonyms listed for ‘heart attack’: ‘hjärtstillestånd’, ‘hjärtattack’ and ‘hjärtinfarkt’. We would then have taken a first step towards building a cross-language terminological ontology.

2.3 Terms – Building Blocks of the Ontology

We saw already in Sect. 2.1 that we sometimes have to look at more than one word at a time to get at the correct meaning of a textual unit. E.g., ‘arrest’ takes on a whole new meaning when seen in the context of ‘cardiac arrest’ than it does in ‘house arrest’. In these cases, it does not make sense to list the meanings of ‘cardiac’ and ‘arrest’ separately and rely on a combinatory process when the two words appear together; the joint meaning is *non-compositional*. This is the reason for us choosing to treat terms as singular units in this thesis, whether they consist of one word or many;[6] otherwise it would be impossible for us to model this type of non-compositionality directly.

Here is what the Swedish Centre for Terminology has to say about terms:

> We cannot tell if a word is a term just by looking at it; terms are basically just regular words. But for a word to be called a term, it should be well known – preferably acknowledged – among experts in a specialized field. Further, the experts should agree on what the term stands for.
>
> A term sometimes consists of more than one word, e.g., the energy term ‘air mass’ and the term ‘interrupting quenching’ in the heat treatment industry.[7] (Terminologicentrum TNC, p. 5)

Researchers in NLP have been aware of the necessity of handling textual units on the sub-word level for a long period of time, as witnessed by the amount of systems available dealing with morphological analysis and decompounding (e.g., Karlsson, 1992). Dealing with units on a super-word level, we have systems for *term spotting* and *term extraction* (Jacquemin, 2001). Cowie (1998) discusses this issue at length, but not from a computational perspective. Research regarding *multiword units* produces yearly conferences and workshops (MWE, 2008).[8] Danielsson (2003) describes an approach for identifying what she refers to as *units of meaning*. These units can consist of one or more words and are not necessarily continuous, but allow for interspersing words that are not part of the unit. We ignore this added layer of complexity in this thesis and only consider continuous strings of words, but we return to the issue again in Chap. 7.

[6] Not all terms consisting of more than one word are non-compositional.

[7] Translation from this Swedish original: “Egentligen går det inte att se på ett ord att det är en term; termer är ju i grunden vanliga ord. Men för att ett ord ska kallas term bör det vara allmänt känt – helst erkänt – bland fackmän inom ett fackområde. Dessutom bör fackmännen vara överens om vad termen står för. En term består ibland av mer än ett ord, till exempel energitermen *relativ luftmassa* och termen *avbruten kylning* inom värmebehandlingstekniken.”

[8] Terms consisting of more than one word are often referred to as multiword units, but not all multiword units are terms.

2.4 Distributional Similarity

Building a model for distributional similarity entails counting co-occurrence frequencies for the words or terms in the model. We speak of co-occurrences of the first or second degree (see Manning and Schütze, 1999). Two words co-occur (first degree) if they both show up in the same document or sentence (the scope of the context varies between different models). A co-occurrence of the second degree takes place when two words both co-occur with a third word. 'Astronaut' and 'cosmonaut' can be assumed to have a high degree of second order co-occurrence,[9] since they both tend to occur frequently along with words such as 'space', 'rocket' or 'Mars'.

Parallel to this, Sahlgren (2006) draws a major distinction between *syntagmatic* and *paradigmatic* relations. Words that stand in a syntagmatic relation to each other are words like 'cradle' – 'baby'; there is a thematic connection, but the two words do not necessarily share many semantic features. These words have a high first order co-occurrence frequency. Conversely, the words 'cradle' – 'bed' are paradigmatically related, and many more semantic features are shared. Typically such word pairs have a high level of second order co-occurrence. Lund and Burgess (1996) refer to these relations as *associative* (for syntagmatic) and *semantic* (for paradigmatic). This distinction is interesting from an ontology learning perspective, since words that stand in a cohyponymy relation to each other (sharing the same superordinate word) are also typically paradigmatically related. We can therefore predict that similarity as measured by second order co-occurrences will be a particularly useful measure for ontology learning, where recognizing cohyponyms is an important subtask.

But how can distributional models catch anything of the semantics of a particular word? Consider the following passage, taken from Nida (1975, p. 167):

> There is some tezgüino. A jar of tezgüino is on the table. You need a lot of tezgüino to get your land cleared. Everyone likes tezgüino. Tezgüino makes you drunk. We make tezgüino out of corn.

Although it is at no point in the text explicitly stated what 'tezgüino' means, after reading the passage, we feel pretty certain that it is some sort of alcoholic beverage. So, given the validity of distributional similarity models, we are no longer forced to search the text for dictionary-like definitions of tezgüino (e.g., "Tezgüino means *alcoholic beverage made from corn*.") but can exploit the implicit information in the text. This is fortunate in the context of ontology learning, since we are often dealing with technical texts, which are not prone to express basic facts explicitly – the reader is assumed to be sufficiently familiar with the topic to get by without them.

[9] This obviously depends on the text collection used for building the model.

The most common theoretical motivation for why the distributional models can be said to capture meaning is given by referring to the work of Zellig Harris (1954). From this work, the *distributional hypothesis* has been (more or less directly) derived: Words that occur in similar contexts tend to have similar meanings.

Another angle of this idea is found in Alston (1964, p. 38), where it is defined when two words (W_1 and W_2) can be said to mean the same thing:

> In most sentences in which W_2 occurs, W_1 can be substituted for it without changing the illocutionary-act potential of the sentence.

Again we see the *context* playing a decisive role in determining word meaning. Of course, when we are just scanning a text for co-occurrences, we lose Alston's criterion for keeping the illocutionary-act potential constant. This is why we move away from Alston's specific synonymy relation to the vaguer notion of similarity or relatedness in the distributional hypothesis. Grönqvist (2006) notes this as well, stating that synonymy implies strong similarity, but that the reverse is often not the case.

We find similar ideas expressed by Sinclair (1996), who is advocating the *empty lexicon*. He argues that the definition of a word in a lexicon should be empty in an initial phase. The definition should then be built up by making observations of the word in different (textual or spoken) contexts. The only (lexical) meaning that exists is accordingly the one given to a word by its co-occurrences with other words.

One matter of discussion when collecting co-occurrences is whether to consider all words within a fixed distance from the *focus word*[10] as co-occurring, or merely those words that stand in a particular syntactic dependency with the focus word. Grefenstette (1994) made some comparative experiments with the two alternative models, but the results were non-conclusive: for high frequency focus words, the "syntactic model" performed better, for lower frequency focus words the reverse was true. Schütze (1998) attributes this to the fact that a syntactic model is typically sparser – has fewer co-occurrences – than a strictly word-based model.

More recent experiments by Padó and Lapata (2007) compare a syntactic model, or a combination of a syntactic and a non-syntactic model (no longer suffering in the same way from the previously mentioned sparsity) with a baseline, word-based model, to the advantage of the combined model. However, their results on the synonym recognition task are not on par with top-performing word-based models from other experiments, such as, e.g., Sahlgren et al. (2008). The word-based model has other advantages, such as having one less processing step, and not being reliant on the existence of a high quality syntactic parser for the language in question. It can be said to be

[10] In corpus linguistics this would be called the *node word*.

the more language-neutral approach of the two, since it applies readily to all languages where tokenization is not a major problem.

When having performed a co-occurrence analysis over a large text collection, we are left with a sparse co-occurrence matrix with a dimensionality of typically tens or hundreds of thousands. Much of this data is noise: spurious co-occurrences between otherwise unrelated words – data which could safely be disregarded without loss of predicative power for the model. Also, the large dimensionality can become a problem in systems where performance is an issue. Such considerations led researchers to investigate dimensionality reduction techniques, such as singular value decomposition (SVD) (Deerwester et al., 1990; Schütze et al., 1995; Landauer and Dumais, 1997; Grönqvist, 2006) – the mathematics are described in Golub and van Loan (1996) – or random indexing (Kanerva et al., 2000; Sahlgren, 2006), both of which perform dimensionality reduction and implicitly abstract away from the noise in the data. When applied to textual data, the SVD approach is commonly referred to as latent semantic indexing (LSI), because of its claims of being able to uncover indirect, or *latent*, relations between words in the data (see Landauer and Dumais, 1997, for a theoretical motivation of these claims). Because of these claims of a potential for discovering latent information, we also include SVD-reduced data in our experiments presented in Chap. 6, along with the non-reduced data.

2.4.1 Distributional similarity model parameters

We have already noted that there is a fundamental choice to be made between using first or second order co-occurrences. In addition to this, there are a number of other parameters which can be varied to achieve different effects in the distributional models. We next discuss the parameters that we examine in our experiments in Chaps. 5 and 6 – these are not the only parameters one could vary, but they are some of the most important.

Size of context window: When using second order co-occurrences to build the model, one typically makes use of a fixed-size sliding window, which is moved over the text, to determine what should be considered to be in the context of the focus word and what not. Varying the size of this window affects the type of information captured by the model (Sahlgren, 2006).

Left/right distinction: Keeping track of whether a context word appears to the left or to the right of a focus word is a simplistic way of adding word order information to a distributional model. If we want to make this distinction, we simply introduce separate features for each context word: one for the left and one for the right context. The disadvantage of doing this is of course that we have to double the size of the co-occurrence matrix, something

which could be costly when working with large document collections.

Distance weighting: Intuitively, words appearing closer to the focus word should be given more weight than context words appearing further away, when building a co-occurrence model. We consider three different distance weighting schemes (d stands for distance measured in number of words from the focus word in the following):

1. Flat: no weighting scheme is applied; all words are given the same weight, regardless of distance from the focus word.
2. Inverse distance: the context word co-occurrence value is weighted by $\frac{1}{d}$.
3. Logarithmic distance: the context word co-occurrence value is weighted by 2^{1-d} (weights decrease faster than for inverse distance). Has been used in, e.g., Sahlgren (2006).

Feature weighting: We can hypothesize that a very frequent context word (appearing as a context word for many different focus words) would contribute less to defining the "co-occurrence profile" of a focus word, than a less frequent context word would. We consider three feature weighting schemes in order to test this hypothesis (the choice of weighting schemes is inspired by Cimiano, 2006):

1. Flat: no feature weighting is applied; all features are given a weight which is based directly on frequency.
2. Conditional probability: if the term under consideration is t, the current feature is f and $freq$ stands for the frequency of a particular term or term-feature pair, then we get:

$$weight(t,f) = p(t|f) \approx \frac{freq(t,f)}{freq(f)}$$

3. Mutual information: measures the mutual dependence between a term and a feature. We use the following formulation of mutual information:

$$\sum_{t_x,f_y} p(t_x,f_y) log \frac{p(t_x,f_y)}{p(t_x)p(f_y)}$$

where $x,y \in \{0,1\}$, indicating the presence or absence of t and f (again, probabilities are estimated using relative frequencies).

2.4.2 Is there Structure in Word-space?

Distributional similarity models are used to find words that are related or somehow similar to each other. When given a geometrical interpretation, these

models are often referred to as *word-spaces* (Sahlgren, 2006), because the dimensions in such a geometrical space are made up of words. This raises the question: Is it possible to make more precise judgments of relations between words in word-space than merely stating that they are similar (or dissimilar)? Are there structures in word-space which can be exploited in order to make more precise statements of the nature of the similarities?

The question has been answered in the affirmative in Caraballo and Charniak (1999), Sanderson and Croft (1999) and Ryu and Choi (2006), although they do not use this terminology, but rather speak in terms of probabilities. Most importantly to us, these authors have shown that it is possible to determine (or at least make a qualified guess at) which of two words is the more specific – a useful piece of information when it comes to constructing a hierarchy. We return to discuss their work in more detail in the following section. Others have identified structures that separate synonyms (Lindén and Piitulainen, 2004; van der Plas and Tiedemann, 2006) or meronyms (Girju et al., 2006) from other relations, though the latter authors mainly work with lexico-syntactic patterns rather than strict word-space models.

2.5 Ontology Learning Approaches

We start with a short overview of four of the most important developments in the field from the early 90's through to today. After this, we present different approaches to ontology learning and also look at attempts to merge various information sources in integrated systems.

Schütze's articles on word-space (e.g., Schütze, 1992, 1998) have had a major influence on all work in the ontology learning field. Though they do not explicitly deal with ontology learning as such, the ideas presented regarding distributional similarity (see Sect. 2.4), or variants thereof, have been incorporated in most ontology learning systems to this date. This is not to imply that Schütze was the "inventor" of the distributional similarity models, but rather that his ideas have been influential on the field.

One of the earlier attempts at something similar to ontology learning was presented by Grefenstette (1994). His aim was to present a system for learning a thesaurus from free text and he was able to show some impressive results, but because no standard ontology learning evaluation measures were available at that time, it is hard to compare his results with more recent work. His work is, like Schütze's, based on distributional similarity.

Spurred on by the accumulating interest in the semantic web, Maedche (2002) was (one of) the first to start using the term ontology learning for this line of work. He also was among the first to suggest a measure for evaluating the results from an ontology learning system (see Sect. 4.3 for more on evaluating ontology learning).

Cimiano (2006) presents new approaches for ontology learning, including *formal concept analysis* (see Sect. 2.5.4) and combining several different knowledge sources in a classification system. He also presents a variation of Maedche's evaluation measure (also discussed in Sect. 4.3).

A growing consensus among ontology learning researchers indicates a need for exploiting many different sources of information simultaneously, where the weaknesses of one source can be remedied by the strengths of another (see Sect. 2.5.6). Typically we will have rule-based approaches (*intra-term information* and *lexico-syntactic patterns* in the list below) providing high precision and low recall, whereas the statistic approaches provide the opposite. We will refer to such integrated systems as *hybrid systems* and this is the main approach followed in this thesis, interpreted in a probabilistic framework. Examples of such systems are given in Sect. 2.5.6 and we present our own experiments in Sects. 6.4–6.5.

The most prolific approaches to ontology learning are:

- Term clustering based on distributional similarity
- Intra-term information
- Lexico-syntactic pattern analysis
- Formal concept analysis
- Statistical measures of term specificity
- Hybrid systems
- Probabilistic approaches

The following sections present an overview of the ideas behind each approach, along with references to the most influential articles and pointers to how the different approaches have been incorporated into the work presented in this thesis.

2.5.1 Term clustering based on distributional similarity

Ruge and Schwarz (1991) present some of the very first work in this area. They use a combination of first and second order co-occurrence to find semantically similar terms. They work with dependency-parsed text and restrict the syntactic relations for both types of co-occurrence. For first order co-occurrence, they look at conjunctive relations and for second order they look at the overlap of heads and modifiers for two terms. As discussed above in Sect. 2.4, the approach of working with dependency-parsed text is again receiving more attention.

An early attempt of learning something similar to what we here call a prototype-based ontology is presented in Pereira et al. (1993). The distributional similarity of nouns with respect to their appearing as objects of transitive verbs is used to form a hierarchical clustering. The article deals with general-purpose words rather than terminology. Another early approach is de-

scribed in Hearst and Schütze (1993), where distributional similarity is used to extend clusters derived from WordNet.

Lin (1998) uses parsed text (i.e., triples of <W_1, dependency-relation, W_2>) to calculate distributional similarities between words. Instead of using the standard cosine measure he introduces a measure that uses the amount of shared information between two words in contrast with the information of the words by themselves. Information here is defined in terms of the probabilities of observing the triples associated with a word with their associated frequencies, as opposed to observing randomly generated triples from the corpus. This measure is shown to perform better than cosine, at least on parsed text, for the purposes of automatic thesaurus construction (for the type of parsed triples used in this experiment).

A procedure for word sense discrimination and word sense clustering is described in Pantel and Lin (2002). A soft clustering is performed (meaning that the same word can belong to more than one cluster) using distributional similarity techniques. Each cluster of which a word W is a member is considered a sense of W. We do not consider word senses in our experiments in Chap. 6, but assume a one-to-one correspondence between terms and concepts, wherefore this method is not used.

Li (2002) uses a distributional similarity technique coupled with *minimal description length* for clustering nouns and verbs. The results are used and evaluated in a PP-attachment resolution scenario, where it achieves a higher precision and lower recall than a WordNet-based approach. This is unexpected because, as mentioned in Sect. 1.2, the main argument against using handcrafted resources such as WordNet is their lack of recall, not their lack of precision. However, the balance between recall and precision is often a matter of parameterization in the system, which means that the presumed higher precision, resulting from using a handcrafted resource, should rather be seen as a rule-of-thumb rather than an anything else.

Widdows (2003) presents a method that is meant for extending an existing ontology (he uses WordNet) with new terms/concepts. First, a distributional similarity model is built over a corpus, where part-of-speech information is taken into account, as a way of performing low-level word sense disambiguation. When adding a term to the ontology, the first step consists in calculating the n closest neighbors of the term, according to the distributional similarity model. For all terms/concepts which subsume at least one of the n neighbors, an *affinity* score is calculated with the neighbor set, which trades off coverage (the concept should subsume as many of the neighbors as possible) with informativity (the concept should be as precise as possible, not subsuming too many non-neighbors). This method does not lend itself to our purposes, since we aim at building an ontology from scratch, rather than extending an existing ontology.

In Weeds et al. (2005), an approach very similar to that of Widdows, mentioned above, is used to place new terms under one of the 36 basic subdivision

classes in the GENIA ontology. Different types of distributional similarity measures are used to extract the n nearest neighbors of a term and then a majority vote amongst these neighbors is used to place the term under the right superconcept. The authors restrict their experiments to dealing strictly with terms, which is the same restriction we impose, discussed in Sects. 2.3 and 5.2.1.

An interesting recast of distributional similarity in terms of recall and precision is presented in Weeds and Weir (2005). Given a focus word W_1 and a candidate related word W_2, recall is measured by the number of contextual features (typically words) occurring together with W_2 that also occur in the context of W_1, mutatis mutandis for precision. Subsumption measures, discussed in Sect. 2.5.5, make implicit use of this notion, since they strive for high recall but *low* precision, to a certain degree.

Wong and Liu (2007) use co-occurrence as measured by the Google search engine to cluster terms by semantic similarity. The first clustering, produced by the Google-similarity measure, is refined in a second step by exploiting Wikipedia's hierarchical category information. Note that there has been some controversy over using Google for research purposes (see Kilgarriff, 2007).

2.5.2 Intra-term information

One basic intuition that is exploited by the methods presented in this section, is that the head of a complex term (noun phrase) tends to be a hyperonym to the whole noun phrase. A 'nuclear power plant' is a 'power plant' which in turn is a 'plant' (in its 'factory' sense). This is referred to as the *head-matching heuristic* in Cimiano (2006), where it is shown to be an effective way of detecting hyperonymy.[11] We include this information as a feature in our experiments in Sect. 6.4.

Oster (2006) uses the terms *determinatum* for the head and *determinant* for the modifier, but we will keep the simpler terms. Oster also makes an analysis of the different kinds of relations that can exist between head and modifier, such as 'wine' (CONTENT) – 'bottle' (CONTAINER), but regardless of the nature of this relation, the is-a relation is implied simultaneously in the vast majority of cases (a 'wine bottle' is-a 'bottle').

In Bodenreider et al. (2001), the authors examine the UMLS for how terms in this type of head-modifier relation are modeled. They find that out of 28,851 such pairs (they require the modifier to be adjectival, which we do not), 43% stand in a hyperonym-hyponym relation in the UMLS, possibly with other terms in between in the hierarchy. They see this partly as an idiosyncrasy of UMLS, which is not strictly *hierarchical*, but also as an indication that some of these relations have simply been forgotten by the designers of the

[11] There are numerous exceptions to this principle, e.g., a 'seahorse' is not really a 'horse' and a 'guinea pig' is not a 'pig'. When dealing strictly with terminology, exceptions are less frequent but they still occur.

UMLS. Their method is accordingly both a quality check and a simple way of extending an existing ontology.

In Navigli and Velardi (2004), a system based mainly on the head matching heuristic and the external knowledge sources WordNet and SemCor is presented. Additionally, word sense disambiguation is performed and an analysis along the lines of Oster is suggested. For the term 'bus service', it is specified that 'bus' is the INSTRUMENT of 'service' and also which senses of 'bus' and 'service' are involved.

SanJuan et al. (2005) use intra-term information when constructing a prototype-based domain ontology. Their model identifies different combinations of heads and modifiers and looks for term variation in a manner inspired by Jacquemin (2001). Combined with a filter based on relations in WordNet, they perform hierarchical clustering based on the intra-term information extracted.

2.5.3 Lexico-syntactic patterns

The seminal paper for this approach is Hearst (1992); in fact, the type of lexico-syntactic patterns she makes use of in her paper have since been referred to as *Hearst-patterns*. The basic intuition is that if we find a passage of text on the following form: "NP1, such as NP2", this indicates that NP1 is a hyperonym of NP2. Hearst gives the pattern a more general form, to allow for lists and co-ordination (e.g., "NP1, such as NP2, NP3 or NP4"). She presents a total of six different patterns in the paper, all of which (or translations of which) are made use of in our experiments in Chap. 6. This approach typically results in high precision and low recall when used for ontology learning, which is why it is a popular part in many hybrid systems.

For the purpose of "fully automatic knowledge acquisition from large corpora of unseen text", Iwanska et al. (2000) also present a list of lexico-syntactic patterns for the extraction of related term pairs, partly overlapping with Hearst's patterns. The patterns presented by Iwanska et al. are not limited to the is-a relation, but also include definitions and cohyponymy (referred to as *related types*).

Rydin (2002) uses this same approach to learn a taxonomy over general Swedish vocabulary from newspaper text. Since this is general vocabulary, she takes some preventive steps to avoid problems of homonymy/polysemy. If word A appears as a hyperonym in a Hearst-pattern for words B and C and later again for C and D, it is assumed that we are dealing with the same sense of A, since the intersection of B, C and C, D is non-empty. Had A instead later appeared with words D and E, the intersection with B and C would have been empty and we would have assumed that there were two different senses of A in play. Because we are dealing with terminology rather than general vocabulary in our experiments in Chap. 6, we do not make use of any word sense disambiguation techniques.

Pantel and Ravichandran (2004) present a method for naming automatically constructed semantic clusters. For a cluster consisting of words like 'pink', 'red' and 'turquoise', they would like to be able to name this cluster 'color' (which is the same as looking for a common hyperonym for the cluster). They first select a number of prototypical instances from each cluster and then calculate a mutual information-based measure for how strong the association is with different nouns in four different Hearst-patterns. This is different from the task in this thesis, in that we assume that the hyperonym of a term will be one of the other terms given by the term extraction process (or the root of the ontology).

Malaisé et al. (2005) search texts for *defining contexts*, in essence sentences where a term is given a definition, implicitly or explicitly. Defining contexts are triggered by the occurrence of particular words like "define" or "designate" or they are triggered by more complex, Hearst-like patterns. By collecting all attributes given to a term in a defining context, terms can be clustered in a hierarchy, reminiscent of formal concept analysis (see Sect. 2.5.4).

Snow et al. (2005) build on Hearst's insight and construct a system for learning lexico-syntactic patterns indicative of hyperonymy. Their patterns consist of paths in dependency-parsed sentences and they use WordNet as a source of known hyperonym/hyponym pairs for training their system. Their system drastically outperforms a system based on Hearst's original patterns in a classification task, where the goal is to recognize hyperonym/hyponym pairs.

Rinaldi et al. (2006) start by parsing text with a dependency parser and then look for word pairs that stand in particular, domain-specific relations to each other. They then examine the contexts and dependency-relations for each relation type and select sets of lexico-syntactic patterns, meant for extraction, manually.

2.5.4 Formal concept analysis

According to Cimiano (2006), formal concept analysis (FCA) can be seen as a clustering technique. Concepts in the hierarchy are clustered on the basis of *formal contexts*, a description of objects and their attributes. A hierarchy is formed by applying set theory to the formal contexts: a concept whose set of attributes subsumes those of another concept is placed further down in the taxonomy; it is considered more specific than the concept whose attributes it subsumes. The attributes used are the set of verbs which have taken the concept under consideration as an argument. FCA results in a structure similar to the one you get from hierarchical clustering: all terms are leaf nodes, internal nodes are abstract, or labeled by their corresponding formal contexts. An example ontology resulting from FCA is shown in Fig. 2.4.

In Cimiano et al. (2004) and Cimiano et al. (2005a), an FCA-based clustering is compared to traditional clusterings based on distributional similarity.

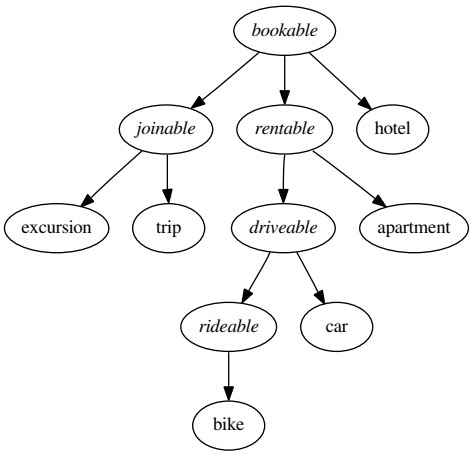


Figure 2.4: FCA example, reproduced from Cimiano (2006). Concepts with labels written in italics are attributes (formal contexts), used to produce the hierarchical structure; the others are concepts in the traditional sense.

The FCA clustering is shown to produce a structure more similar to the gold standard used (a handcrafted ontology from the tourism domain), using the taxonomic overlap evaluation measure (see Sect. 4.3). The FCA clustering has a higher taxonomic recall, since some of its formal context-concepts also occur in the gold standard. This means that we cannot be sure whether we would see the same improvement in taxonomic overlap in an experiment such as presented in Sect. 6.1, where we have a fixed set of terms to be clustered.

There is an interesting theoretical discrepancy between FCA and the term specificity measures discussed in Sect. 2.5.5, directly following this section. FCA places a term (concept) that occurs in the widest set of formal contexts at the bottom of the hierarchy, whereas the term specificity theories claim that more specific terms typically occur in more homogeneous contexts than do general terms. Some support for the latter is given in an experiment in Sect. 6.2.5.

We are not using FCA as part of the ontology learning system presented in Chap. 6, mainly for the reasons just mentioned and for its dependence on a parser for constructing the formal contexts (which might not be available for all languages).

2.5.5 Statistical measures of term specificity

In Spärck-Jones (1972), we find an early attempt at giving term specificity a statistical interpretation. Spärck-Jones there presents a measure for term specificity, or term weighting, which is meant to boost the importance of low-frequency terms in an IR setting. Note that this measure is meant to function as part of an IR system and is not designed to order arbitrary terms according to

their specificity – the connection between low frequency and high specificity is strong but does not always hold.

Some researchers have made use of measures from information theory in the domain of ontology learning. Most notably, Caraballo and Charniak (1999) introduce a method for ordering a set of nouns according to their specificity. The intuition behind their measure is that more specific terms are modified by a fixed set of words, whereas more general terms can be modified by any number of other words. They build a context vector for the words occurring in a window around the nouns and calculate the entropy of this vector. The higher the entropy, the more different words are used to modify the term, and the less specific the term is, so their reasoning. This is evaluated on three sets of related terms, with the hierarchical ordering taken from WordNet. For over 80% of the noun pairs, their measure is able to identify which of the two nouns is more specific (the baseline accuracy of taking a random guess is of course 50%). Note that this problem is much simpler than separating related term pairs from non-related term pairs; here we already know that the two terms are related, just not which term is more general and which more specific. We use this measure in our experiments, as discussed in Sect. 6.2.

Sanderson and Croft (1999) use the concept of *subsumption*, where one term is said to subsume another if the first term occurs in all the documents where the second term occurs, but not vice versa. In their experiments, it turned out that the requirement that *all* occurrences of the second term should be "covered" by the first term was too strong; instead a threshold is introduced, saying that it is enough if 80% of the occurrences are covered. We use a variant of this measure, explained in Sect. 6.2.1.

Ryu and Choi (2004) introduce a measure called *Information Quantity*, which also makes use of entropy and frequency of term distributions in determining the specificity of a term. In Ryu and Choi (2006), the authors combine this measure with other well-known methods for ontology learning and the results are evaluated by comparing with a gold standard. Their work can be regarded as a continuation of Caraballo's methods, discussed previously. The results are somewhat hard to interpret, but the entropy-based measures give a high recall whereas the other measures give a higher precision on the ontology learning task. They start with an empty taxonomy and iteratively add one term at a time, using their measures for similarity and specificity to place the new term in the hierarchy. We stick with Caraballo and Charniak's original measure in our experiments in Sect. 6 because of its simplicity and intuitiveness.

2.5.6 Hybrid ontology learning approaches

In Caraballo (2001), three different sources of information are combined for learning a hierarchical structuring of words: distributional similarity (using

selected grammatical contexts from parsed text), Hearst-patterns and the term-specificity measure described in Sect. 2.5.5. Caraballo first builds a baseline system using the distributional similarity information and the Hearst-patterns and then demonstrates that including the specificity information results in a more accurate hierarchy.

Ogata and Collier (2004) use versions of Hearst-patterns and intra-term information in combination to build a terminological ontology in the molecular biology domain. Due to the characteristics of the terms in this domain, the terms do not lend themselves well to classical morphological analysis. Instead, terms are searched for common endings, where endings are simply character strings. These common endings in combination with the information from the Hearst-pattern analysis are used to build up a hierarchical term structure.

Nenadic et al. (2004) calculate term similarities using a combination of three different approaches. First they make use of intra-term information, by counting shared heads and modifiers among term variants. Next they use Hearst-patterns for identifying co-ordinated terms (cohyponyms). Finally they calculate something which resembles a distributional similarity model, but they restrict the contexts to certain lexico-syntactic environments. The results from these three approaches are combined using a linear combination, where the weight for each approach is set by comparing results to a small example ontology constructed by a domain expert.

Mani et al. (2004) use a number of different knowledge sources for constructing an ontology. They look at Hearst-patterns, a subsumption measure reminiscent of Sanderson and Croft (1999), the head matching heuristic mentioned above and finally relations from existing ontologies such as the Gene Ontology or WordNet. We choose to include all these knowledge sources in the experiments in Chap. 6, except for the external ontologies, which we exclude for reasons given in Sect. 1.2.

A number of different information sources are incorporated in Cimiano et al. (2005b), in order to train a classifier to recognize term pairs in the hyponymy relation. They make use of Hearst-patterns, WordNet relations, the head matching heuristic and versions of Sanderson and Croft's subsumption measure to train a one-class support vector machine with good results. We follow this approach in Sect. 6.4, though we use a dual-class support vector machine and have a different set of features.

2.5.7 Probabilistic approaches

Some research has been directed towards reinterpreting (or recalculating) the output of the previously described approaches as probabilities. The main idea is that a higher score from one of the previously described measures corresponds to a higher probability that the predicted relation holds. Exactly how this transformation is done differs from case to case and is not always described explicitly.

In an experiment in Caraballo (2001), the assumption is made that there exists a "true" hierarchy from which the observed co-occurrence data was generated; the task then is to find the hierarchy which gives the highest probability to the observed data. Accordingly they assume a process which generates a noun with probability $P(n)$ and then a context word with probability $P(w|n)$. Caraballo then suggests a backing-off version of this, where the superordinate (hyperonym) node c of n is taken into account: $P(w|n) = \lambda P(w|n) + (1-\lambda)P(w|c)$, thus incorporating the hierarchical structure in the calculations. She recasts the experiment described earlier in Sect. 2.5.6 in this setting, but does not measure any improvements in accuracy over the non-probabilistic version.

Snow et al. (2006) describe a probabilistic approach to constructing an ontology based on two classifiers trained to recognize hyperonym/hyponym pairs and cohyponym pairs, respectively. The ontology is built up stepwise, by at each step adding the relation that maximizes the probability of the ontology *as a whole*. This is done by calculating all new relations that will be created by adding a particular relation to the ontology, using the transitive closures of the hyponymy and cohyponymy relations. This is also the approach we follow in our experiments in Sect. 6.5.

2.6 Translational Equivalence for Terms

Because we wish to model terms from different languages in our learned ontologies (see Sect. 2.2), it becomes important to know which terms mean the same thing, which terms are *translational equivalents* of each other. This section introduces the main approaches for extracting equivalents automatically, given a parallel or comparable corpus.

Tiedemann (2003, p. 12) gives the following definition for the task of bilingual lexicon/dictionary extraction: "*Bilingual lexicon extraction aims at the identification of lexical word type links in parallel corpora*".[12] The focus lies on *word types*, in contrast with the task of *word alignment*, where the focus is on *word tokens*. The bilingual dictionary extraction task is thus not concerned with which word(s) in a particular source language sentence correspond to which word(s) in a particular target language sentence; instead, the entire corpus is taken into consideration and a most probable translation of the source language word is sought in this global perspective. The two contrasting views give rise to different evaluation schemes and we indicate for all articles below which of the two tasks is being evaluated.

The cross-language approach to ontology learning advocated in this thesis depends on the existence of a bilingual dictionary, or a parallel or comparable corpus from which to extract such a dictionary. Resnik and Smith (2003) pro-

[12] We choose also to include comparable corpora in the definition.

pose a way of remedying the absence of these resources, by describing how the web can be mined for parallel texts. The parallel texts would then have to be filtered to keep only domain-specific texts, which might make the availability of large enough text quantities an issue, depending on the language pair and domain. Somers (1999) presents a general approach for all the steps from extracting a parallel corpus via sentence alignment to bilingual dictionary extraction techniques. Merkel (1999) likewise gives a thorough overview of the translation process, discussing term extraction and bilingual dictionary extraction but also how the results can be incorporated in the workflow for human professional translators.

2.6.1 Cross-language distributional similarity

For a recapitulation of distributional similarity, see Sect. 2.4. How to exploit distributional models when working in a cross-language setting is explained more closely in Sect. 5.2.1. Here we give an overview of previous work using distributional similarity in a cross-language setting.

Each word or term (row in a co-occurrence matrix) can be compared to each other word or term, using similarity measures defined for vectors, as discussed in Sect. 2.4. There is a plethora of such measures, many of which have been evaluated on the dictionary extraction task, or one similar to it. In Ribeiro et al. (2000), a total of 28 different similarity measures are evaluated on extracting equivalents from aligned parallel texts. They use one language pair (Spanish – Portuguese) for testing on a parallel corpus containing about 18,000 words, which is rather small. Two of the highest ranking measures in that evaluation, the cosine measure and the mutual information measure,[13] are compared in our experiments in Sect. 5.2.1.

A few years earlier, Smadja et al. (1996) showed differing results, indicating that the Dice coefficient was in fact more effective than mutual information for measuring the association between two terms (or collocations in Smadja et al.'s case). The reasons for this discrepancy are unclear.

To bypass the potential problems connected with the large dimensionality of the co-occurrence matrices (see Sect. 2.4), Sahlgren and Karlgren (2005) experiment with random indexing for the dictionary extraction problem. The results are evaluated in a type-based evaluation and are shown to give higher accuracy than GIZA++[14] (see Sect.2.6.2 below). In Hjelm (2007), we come to the opposite conclusion: that GIZA++ outperforms the distributional similarity measures, with or without random indexing. The difference in outcome is most likely due to the different ways of training GIZA++ in the two experiments; Sahlgren and Karlgren use parallel documents whereas we use parallel sentences. The SVD method for dimensionality reduction has yet to be evaluated on the dictionary extraction task.

[13] Referred to as *average mutual information* in Ribeiro's evaluation.
[14] http://www-i6.informatik.rwth-aachen.de/web/Software/

2.6.2 Statistical machine translation

The seminal paper in this area is Brown et al. (1993), where a series of five models for calculating word-alignments are presented. A word-alignment links each word in a source language sentence to one or more words in the corresponding target language sentence. The five models are designed to be applied in a chain-like fashion, where each later model adds a level of complexity to the previous one.

GIZA++, a piece of software which implements Brown et al.'s models, produces a bilingual dictionary file, where each source language word or term is listed with its possible translations and associated probabilities. The most probable translation of a particular source term can thus be found by sorting the possible translations in descending order based on their associated probabilities. We make use of GIZA++ output in our experiments in Sect. 5.2.

Melamed (2000) describes three statistically based approaches, all using co-occurrence information coupled with, e.g., a noise model or statistical smoothing. Melamed's models deal with words and he makes a fundamental one-to-one assumption for the word alignment task: that one word in a particular source language sentence corresponds to one word in the parallel target language sentence. Depending on the language pair involved, this assumption will be more or less fortuitous; Melamed performs both a type- and a token-based evaluation on an English-French corpus, a language pair where the assumption will work better than for many others. He demonstrates improvements over Brown et al.'s previously mentioned first model. Note that the one-to-one assumption makes more sense in our case, since we are using texts where term spotting has been performed (see Sect. 2.3), enabling us to treat multiword terms as single units.

Och and Ney (2003) propose extensions to Brown et al.'s translation models and show improvements over the non-extended system on a token-based evaluation scheme. They present a variety of added information sources, including part-of-speech information and a bilingual dictionary, all handled in a probabilistic framework. We chose to work with the non-extended GIZA++ model in our experiments in Sect. 5.2 mainly for its ready availability and widespread application in the statistical machine translation research community.

2.6.3 Hybrid approaches for translation

Tiedemann (2003, 2005) proposes a method for word alignment which uses both distributional similarity measures[15] and the dictionary files produced by GIZA++ mentioned above. Tiedemann also uses other information, such as string matching (edit distance) and part of speech, and so is able to boost the performance of GIZA++ by weighting the scores of the different sources to-

[15] He refers to these measures as *co-occurrence measures*.

gether. The weights in Tiedemann (2005) are learned using a genetic optimization algorithm and the system clearly outperforms GIZA++ on a token-based word alignment evaluation task.

Kraif (2003) combines distributional models with a model for *cognateness*, similar to Tiedemann (2003). He examines four different similarity measures and gets the best result with a measure he calls "P0" or "the log-probability of the null hypothesis". The method is evaluated in a token-based word alignment setting, but no comparison with, e.g., GIZA++ is performed, which makes it hard to judge the effectiveness of the approach.

Volk et al. (2002) look at different measures for term association. They get good results for comparing the lengths of the terms extracted as translations from a sentence-aligned parallel corpus and filtering out term pairs where the difference is too large. The length feature could be seen as a subset of the differences measured by the edit distance, though perhaps applicable to less related language pairs. The idea would be that a long, complex term in one language would correspond to a long complex term in the other, even if they, e.g., use different alphabets, but this last point remains to be investigated.

We present results in Hjelm (2007), showing that combining GIZA++ output with the output of different distributional similarity measures gives increased accuracy, compared with any single system in isolation. The experiments are described more closely in Sect. 5.2.

Including features like string similarity or edit distance gives a slight boost in performance in the models described in this section. String similarity features have the disadvantage of making the process less language independent and in the experiments presented in Sect. 5.2, no such information has been used.

2.6.4 Working with comparable corpora

In cases where no appropriate parallel corpus exists, it is usually possible to at least construct a *comparable* corpus, i.e., a corpus in two or more languages composed of L1 texts that have been collected using the same sampling techniques (McEnery et al., 2006). Experiments with such texts are presented in Sect. 5.3.

In Fung and McKeown (1997) and Rapp (1999), the authors describe a method for dealing with comparable corpora in the context of the dictionary extraction task. The main assumption they make is that if word *A* is a translation of word *B*, the words frequently co-occurring with word *A* also tend to be translations of words frequently co-occurring with word *B*. This can be thought of as the correlating formulation of the distributional hypothesis (Sect. 2.4) in the cross-language setting. This means that, if we build a distributional similarity model using second order co-occurrences for one language, we can take a bilingual, general purpose, dictionary and translate the words co-occurring with the source language word. This gives us a mapping from

the dimensions of the model of the source language to the dimensions of the model of the target language. The idea is that general purpose dictionaries may well contain general words occurring in the context of the term we wish to translate, but the probability that it will contain the (mostly more specific) term itself is much lower. In Sect. 1.2, we list reasons for learning domain ontologies even though handcrafted resources such as WordNet already exist. Here, we see many of the same reasons coming into play when we consider learning domain-specific bilingual dictionaries, even though manually constructed general dictionaries already exist for many language pairs.

Before this, Rapp (1995) had presented an interesting approach for solving the same problem, but without the usage of a bilingual dictionary. His method involves permuting the columns of one of the co-occurrence matrices and searching for a maximum similarity between the two matrices. Finding the maximum similarity permutation would also provide a mapping between the two matrices, specifying which columns correspond to each other. Since each column corresponds to a word (in a second order co-occurrence matrix), this would also provide the translation. Unfortunately, Rapp shows that without providing a set of seed translation pairs, similar to the method using the bilingual dictionary just discussed, this computation is intractable.

Déjean et al. (2002) present a method building on Fung and McKeown (1997) and Rapp (1999), additionally making use of a multilingual thesaurus. Even if the word we want to translate is present in the multilingual thesaurus, we typically have a problem with polysemy/homonymy and therefore have to perform some type of word sense disambiguation to arrive at the correct translation. Adding a multilingual thesaurus of course gives better results than using the purely corpus-based approach, but the decision for this thesis was not to assume the existence of such a thesaurus, for reasons given in Sect. 1.2. In the same spirit, Déjean et al. make *direct* use of a general bilingual dictionary, assuming that some of the specialized terminology will also be translated there. The results from these three information sources are then combined in a supervised manner, where weights are set according to the results of a manually constructed set of term translation pairs. The authors also describe how their method could be used to add translations in a new language to concepts in an ontology. In Déjean et al. (2005), the authors show how the results from this method can be of use in a cross-language information retrieval setting.

Holmlund et al. (2005) use similarities of a higher order for performing the dictionary extraction task. Similar to the approaches discussed above, a set of reference translations is used to establish a common ground for comparing distributional similarity measures across the language borders. For each word in a language, a similarity value using second order co-occurrence is calculated with each of the k reference words. This results in a k-dimensional similarity matrix for both languages involved, where the dimensions have the same meaning in the two matrices. Vectors in both matrices can now be com-

pared with each other, resulting in *third* order co-occurrence similarities. It is not demonstrated that this approach gives any improvements in accuracy over the previous measures, using the more well-established second order co-occurrences.

In Wu and Fung (2005), an attempt is made to sidestep some of the problems with comparable corpora, by first mining them for parallel sentence pairs and then treating the set of mined sentence pairs as a parallel corpus. First, the documents in one of the languages are glossed into the other language via a bilingual dictionary, then similar documents are identified via a similarity measure (*cosine*). Next, sentences within a similar document pair are parsed and sentence pairs across documents are scored for their likeliness of being translations, using constraints from *inversion transduction grammars*. The method is said to identify parallel sentences at an uninterpolated average precision of 64.7%, but no results are given for the dictionary extraction task.

2.7 Exploiting Cross-language Data

One of the major hypotheses in this thesis is that adding information from other languages will improve the results of a task that traditionally is approached in a single language framework (we discussed this in Sect. 2.2). This section looks at other attempts of profiting from cross-language information in a variety of NLP settings.

Aizawa and Kageura (2001) look at co-occurrences of English and Japanese keywords in scientific articles. The co-occurrences are used to form a graph where the edges are weighted by the number of co-occurrences. They perform a clustering by splitting up the graph using the *minimum edge cut*, a graph-theoretically motivated approach. They do not compare their cross-language results with single language results, so we do not know if the cross-language data gives a better clustering than the single language data. On the other hand, the resulting resource (cross-language term clusters) has added value, because it can be used in, e.g., a cross-language information retrieval setting.

Simard (1999) and Borin (2000) look at the effects of adding a third language for solving problems normally involving two languages: sentence and word alignment in parallel corpora. They both conclude that information from the third language increases the accuracy of the system, evaluated on cross-language alignment tasks. Simard also concludes that "the more languages, the merrier", implying that adding a fourth language would improve the results further. Borin also notes that adding a language from the same language family is more helpful than adding a language from a different language family.

Somewhat related is work such as presented in Yarowsky et al. (2001), where annotations from well established English NLP tools (taggers and

parsers) are projected via word alignment to texts in other languages. These annotations, though noisier than their English counterparts, can in turn be used to train NLP tools for use in the new language.

In Carpuat et al. (2002), the authors present a method of creating a bilingual ontology by merging two monolingual ones; they use the English WordNet and the Chinese HowNet. This means dealing with the idiosyncrasies of the two ontologies, such as their different scopes and granularities. In the absence of a bilingual dictionary (or where the coverage of such a dictionary is insufficient), their method relies on calculating the correspondences between concepts using automatic translation methods such as those described in Sect. 5.3. Carpuat et al.'s method differs from ours, in that we do not start out with pre-existing ontologies, but rather create a cross-language structure from scratch.

Dyvik (2005) proposes to use *semantic mirrors* to perform both word sense discrimination and a hierarchical ordering of word senses using a word-aligned parallel corpus. A key concept here is the so called *t-image* of a word W, which consist of all the words in the target language which have been word-aligned with W. The linking process is then reversed by taking the t-image of all the words in the first t-image, which gives us a set of words in the source language, called the *inversed t-image* of W. By going back and forth between languages (Dyvik also forms what he calls a *second t-image* by going back to the target language from the inversed t-image), one is able to form sets of words in either language that share at least one member (apart from W). These sets correspond to senses of W. Further, these sets can be used to produce a hierarchy by using a set inclusion analysis to form an upper semilattice. A disadvantage is that the semantic mirroring process depends on high quality (i.e., manually produced) word alignments, something which is hard to come by on a larger scale. Dyvik also reports that the method works better for adjectives than for nouns, which also makes it less useful for our purposes (most terms are nouns or noun phrases).

van der Plas and Tiedemann (2006) use a cross-language distributional similarity model for targeting the extraction of synonyms rather than generally similar words. The idea, not unlike Dyvik's, is that synonyms will "co-occur" with the same translations in a parallel corpus, whereas this would be less true for other lexical semantic relations such as antonymy or hyperonymy. Their system outperforms a traditional, single language, distributional similarity model on the task of identifying synonyms in the Dutch EuroWordNet. We are not modeling synonymy in our experiments (we assume a one-to-one mapping between concepts and terms), so this method is not applicable in our case.

Lindén and Piitulainen (2004) use distributional similarity (from parsed text) to learn clusters of synonymous words. To evaluate these synonym clusters, they use a technique similar to Dyvik's. They look up a word W in a bilingual dictionary and get a group of words as possible translations (Dyvik's first t-image). These translations are then translated back to the

source language, generating a set of source language words (the inversed t-image). This process is repeated with at least one more bilingual dictionary, translating into a different target language, and the intersection of the reversed t-images is used as a set of synonyms for W. More than one other language is used to avoid incorrect results if one or more words in the t-image should be homonymous.

2.8 Summary

This chapter has provided an overview of different approaches for solving problems involved in ontology learning, or, specifically, cross-language ontology learning. Not all approaches were developed with this application in mind, but most of them have proven useful in various types of ontology learning systems.

A common trait of all the methods discussed in this chapter is that they all deal with meaning in one way or another, whether they aim at capturing similarity (e.g., statistical machine translation and distributional similarity) or at exploiting asymmetries (e.g., subsumption and Hearst-patterns). A large part of our task during our experiments will be to find strategies for combining this wealth of information in ways that take advantage of the strengths and weaknesses of the different approaches. We present experiments to this end in Chaps. 5–6.

3. Resources

A few initiatives, some of them ongoing, have been taken in the ontology learning community to establish a standard document collection along with a corresponding domain ontology to be used for evaluation. The existence of such a standard would facilitate automated qualitative comparisons among ontology learning systems and approaches – we return to this point in Chap. 4. To date, the most ambitious initiative was taken for the *2nd Workshop on Ontology Learning and Population*, held in Sydney, Australia, 2006. Participants were encouraged to perform experiments on the *OLP2 dataset*,[1] a corpus, ontology and knowledge base in the soccer domain. This dataset is still freely available for research purposes, but it has not yet had the needed homogenizing impact on the community. Of course, an exaggerated homogeneity can also be harmful to a field, such as is arguably the case with the use of the Wall Street Journal part of the Penn Treebank (Marcus et al., 1993) in the *parsing* community. For the ontology learning field, the danger lies rather in slipping into the opposite ditch, where the lack of standards is preventing competitive development.

We choose to work with a different setup for this thesis: a corpus and a terminological ontology dealing with European Union (EU) related issues. There are several reasons for this, the main reason being the massively parallel nature of the corpus (described in Sect. 3.2.1). Another reason is the availability of the rather large terminological ontology, where all terms have been translated into most of the EU languages (see Sect. 3.3.1). This combination of a parallel corpus and a cross-language terminological ontology is crucial to several of the experiments we describe in Chaps. 5–6.

On a smaller scale, we also work with a corpus and an ontology from the domain of human anatomy. This corpus is *comparable* rather than parallel (meaning that the texts in the different languages are not translations of each other but still deal with the same topics). This has some disadvantages but also poses some interesting challenges; experiments on this material are also described in Chaps. 5–6.

[1] Available at http://www.dfki.de/sw-lt/olp2_dataset/

3.1 Pre-processing

We apply certain low-level linguistic pre-processing to all corpora and ontologies, and we describe the different steps in the following sections.

3.1.1 Morphological analysis

In order to lessen some of the detrimental effects caused by data sparseness, we lemmatize the corpora to get more occurrences of each base form term. We use Intrafind's[2] LiSa system for morphological analysis (Hjelm and Schwarz, 2006) for all languages except Swedish, where we use the system described in Carlberger and Kann (1999). We also lemmatize the terms in the ontologies, in order to allow for direct matching between the lemmatized corpus texts and the terms in the ontologies. E.g., the term 'state trading' will be turned into 'state trade' by the lemmatizer in the running text. So, if we want to be able to map between terms in the texts and in the ontology, we need to make sure that we have the term as 'state trade' in the ontology as well.

We also analyze each word in all terms with a compound splitter. This step provides the information that is needed in the experiments described in Sect. 6.4. For all languages except Swedish we again use the LiSa system and for Swedish we use a system described in Sjöbergh and Kann (2004).

The increase in recall (i.e., finding more occurrences of each term) we get from this pre-processing will partly have to be paid for with loss in precision. In Hjelm and Schwarz (2006) we report a cumulative error rate of 1.3% for the lemmatization and compound splitting. The majority of the errors are of the type that no analysis is given. In such cases, the precision is not affected, because the system then simply leaves the word as it stands in the text.

3.1.2 Term spotting

Since the concepts in the ontologies are associated with terms rather than words, we need a way of letting the system treat multiword terms in the corpus as single units, as well as being able to distinguish single word terms from "mere" words. We therefore use a simple term spotting technique (see Schwarz, 1990, Jacquemin, 2001, for more on term spotting), marking the longest consecutive string of words that also appears in the ontology, as a term.

A complete pre-processing of the data works like this:

```
The zygomatic bone (malar bone) is a pair bone of the
human skull. ->

the TERM_zygomatic_bone_55158 ( malar TERM_bone_34122
```

[2] http://www.intrafind.de

```
) be a pair TERM_bone_34122 of the human
TERM_skull_49338 .
```

This makes the terms recognizable to the system and, as mentioned, also allows the system to treat multiword terms as single textual units. Note that the terms 'malar bone' and 'pair bone' are not marked as terms in this example because they are not listed as terms in the ontology.

3.2 Corpora

As discussed in Sect. 1.2, this thesis aims to investigate different aspects of learning ontologies from text. In order to do so, we naturally need large text collections on which to test our theories. This section presents the text collections (corpora) used in our experiments.

3.2.1 JRC-ACQUIS Multilingual Parallel Corpus

This corpus consists of legal texts concerning matters involving the EU. The number of words per language varies between 6.5 million (Swedish) and 7.8 million (French) among the languages used in the experiments: German, French, English and Swedish.[3] This choice of languages is to a certain extent arbitrary; it is based on the existence of readily available, high quality preprocessing software, such as lemmatizers and compound splitters. The corpus is parallel and contains over 20 European languages in total (Steinberger et al., 2006). Note that there is a version 3.0 released of this corpus, which is almost three times bigger than the one used in this thesis (version 2.2).

The corpus is distributed in a format where it has been aligned automatically on the paragraph level. The paragraphs are very short and usually only contain one sentence or even one part of a sentence. There are two alignment versions available for download;[4] we have opted for the version produced by the Vanilla aligner.[5] Since the alignment process is automatic, we are of course introducing an error source here; unfortunately we are unaware of any figures concerning alignment accuracy for this corpus and the languages involved. To put it informally, we consider the effects of this error source small but non-negligible.

Below is a short exemplifying passage from one of the documents from the corpus (excerpt from document "jrc32005R0123-en.xml").

> Commission Regulation (EC) No 466/2001 [2], sets maximum levels for certain contaminants in foodstuffs. (2) According to Regulation (EC) No

[3] The differences are due to the idiosyncratic ways of the different languages of, e.g., forming compounds.

[4] http://wt.jrc.it/lt/Acquis/

[5] http://nl.ijs.si/telri/Vanilla/doc/ljubljana/

> 466/2001, the Commission shall review the provisions as regards ochratoxin A (OTA) in dried vine fruit and with a view to including a maximum level for OTA in green and roasted coffee and coffee products, wine, beer, grape juice, cocoa and cocoa products and spices taking into account the investigations undertaken and the prevention measures applied to reduce the presence of OTA in these products.

3.2.2 Wikipedia anatomy corpus

We have downloaded the Wikipedia[6] pages filed under the 'Anatomy' category for English, French, German and Spanish.[7] This resulted in about 7,300 pages for English, 2,600 for French, 2,400 for German and 1,000 for Spanish. The corresponding number of words is about 4.4 million for English, 1.1 million for French, 890,000 for German and 400,000 for Spanish. The choice of domain and languages is partly influenced by a project, briefly described in Sect. 3.3.2, which provided a context for some of our experiments, even though our experiments do not constitute an integral part of that project. We stripped the texts of HTML and other markup or scripts, as well as Wikipedia-related text. It should be noted that Wikipedia is constantly changing and growing and that these numbers reflect the status as of February 2007.

We again display a short passage from one of the texts, to give an idea of the type of content featured in this corpus (excerpt taken from the document "Index_finger").

> It [the index finger] is usually the most dextrous and sensitive finger of the hand, though not the longest. It may be used to point to things, for hunt and peck typing, to press an elevator button, or to tap on a window. A lone index finger often is used to represent the number 1, or when held up or moved side to side (finger-wagging), it can be an admonitory gesture.

We can imagine a continuum, going from parallel corpora on one extreme, to comparable corpora on the other. Suppose we have a corpus with documents in two different languages, where we have document pairs with newspaper articles by different writers relating the same events in two languages. The documents are not translations of each other, which they would be in a parallel corpus. Still we have a distinct sense of which document in the one language corresponds to which in the other, which we would not have in a typical comparable corpus. Our Wikipedia corpus is similar in structure to the corpus in the example, in that we know for some documents, via the Wikipedia links, which documents correspond to each other. For other documents, there are no correspondences, because not all articles exist in all languages. This means

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

[7] For English: http://en.wikipedia.org/wiki/Category:Anatomy. This page links to the corresponding pages in the other languages.

that there are traits of parallelism in our Wikipedia corpus, though it definitley is closer to the ‘comparable’ end of the scale.

3.3 Gold standard terminological ontologies

Parallel to the two corpora described above, we are using two terminological ontologies in our experiments.

3.3.1 Eurovoc

Eurovoc V4.2[8] is a freely available multilingual thesaurus with entries in more than 20 languages, and it covers topics where the EU is active, e.g., law, politics, economics and science. The thesaurus contains 6,645 concepts, each of which is given a *descriptor*, or recommended term, in each language. Only the descriptors are taken into consideration throughout all our experiments, which means that we make a simplifying assumption that there is a one-to-one relationship between terms and concepts. The average depth of the Eurovoc hierarchy is 4.32 and the maximum depth is 8. An example of a small is-a hierarchy is displayed in Fig. 3.1.

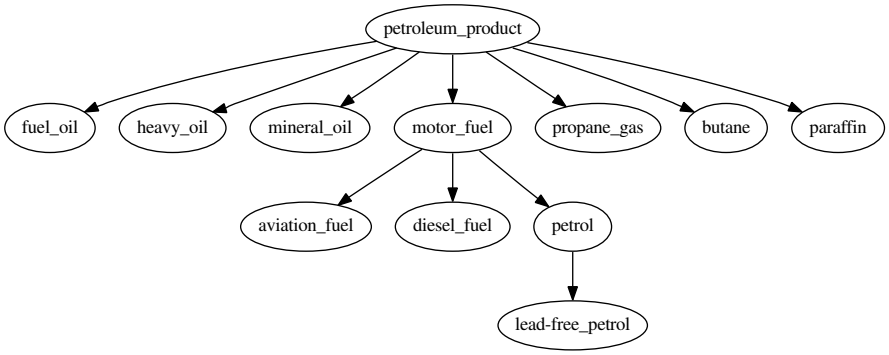


Figure 3.1: Excerpt from Eurovoc, terms involving petroleum products.

Apart from hierarchical (is-a) relations, also equivalence and associative relations are listed in Eurovoc, but only the hierarchical relations are considered in our experiments. Eurovoc is divided into 21 *fields*, each representing a domain of interest in the EU. E.g., we have the fields ‘politics’, ‘finance’ and ‘education and communications’ and examples of terms from these respective fields are ‘composition of parliament’, ‘financial accounting’ and ‘educational administration’.

A minority of the terms in the hierarchy have more than one super-ordinate term; this concerns mainly geographical entities, which have both part-of and

[8] http://europa.eu/eurovoc/

is-a relations marked (though the marking itself does not discriminate between the two relation types). E.g., 'Sweden' has 'Northern Europe' and 'EU member state' as super-ordinate terms, but only with the latter does it enter into an is-a relation. In such cases, we have processed the data to remove this type of polyhierarchy and only keep the is-a information (this is generally discernible using the ID numbers of the concepts, though this is only a heuristic). Some concepts have *only* part-of information listed, e.g., 'Skåne county' only has 'South Sweden' as parent in the hierarchy. These cases are not marked in the thesaurus, which means that we are not able to filter them out automatically, and in turn, that we end up with a small number of part-of relations in our gold standard. We estimate the number of such cases to make out no more than 2–3% of the total, which means that they will have a negligible effect on our evaluations.

One source of controversy, which has also affected other thesauri or taxonomies such as WordNet, is what to do with instances (see Miller and Hristea, 2006, for a discussion). An is-a hierarchy imposes inheritance on the vertical axis, but the inheritance chain is broken when an instance is encountered. E.g., 'sea captain' is a 'profession' and 'Ahab' is a 'sea captain', but 'Ahab' is hardly a 'profession'. In version 2.1 of WordNet, instances have been marked as such, but this is not the case with the current version of Eurovoc – this is an imperfection in our gold standard we will have to live with throughout this thesis.

Eurovoc partitions

For some of our experiments it is necessary to separate the Eurovoc data in training and test sets. To be able to perform cross-validation in our machine learning experiments in Sect. 6.4, we split the Eurovoc taxonomy into ten parts, approximately equal in size. We do this by making use of the fact that Eurovoc is already segmented into 21 fields, as mentioned. We thus have nine partitions containing two fields each and a tenth partition containing three fields and we number the partitions 0–9. The partitions contain the following fields:

0: *Business and Competition* and *International Organizations*
1: *Economics* and *Energy*
2: *Education and Communications* and *Transport*
3: *Environment* and *Agri-foodstuffs*
4: *Finance* and *Industry*
5: *International Relations* and *European Communities*
6: *Law* and *Employment and Working Conditions*
7: *Politics*, *Trade* and *Production, Technology and Research*
8: *Science* and *Agriculture, Forestry and Fisheries*
9: *Social Questions* and *Geography*

3.3.2 FMA ontology

As part of the recently started THESEUS MEDICO project,[9] funded by the German government, a system for querying and analyzing medical information (medical records, x-rays, etc.) is currently under construction. Certain parts of such a system would arguably benefit from a domain ontology as a source of background knowledge during, e.g., information retrieval or image recognition tasks. In the domain of (human) anatomy, there exists such an ontology: the Foundational Model of Anatomy (FMA) ontology. It is developed by the Structural Informatics Group at the University of Washington, and it is open source.[10] It contains about 100,000 English terms, 8,000 Latin, 4,000 French, 500 Spanish and 300 German terms. There are also a few terms in other languages such as Italian and Filipino, but we disregard these. The ontology models mainly the hierarchical is-a and part-of relations, but we only consider the is-a structure. Figure 3.2 shows an example excerpt from the FMA ontology.

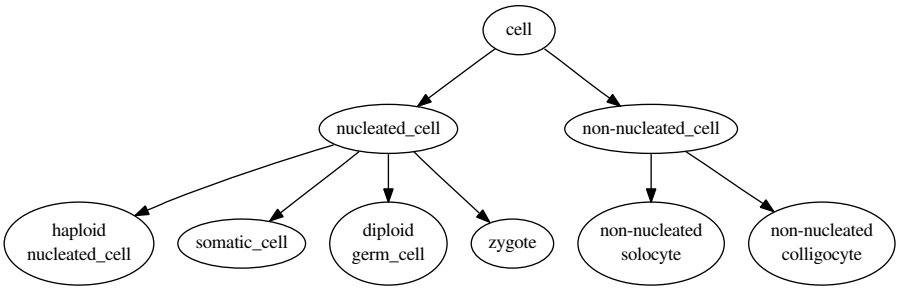


Figure 3.2: Excerpt from the FMA ontology, terms involving cells.

3.4 Bilingual dictionary: Wiktionary

For the experiments described in Sect. 5.3, we also need bilingual dictionaries for English-German, English-French and English-Spanish. We decided to make use of the open source dictionaries from Wiktionary.[11] They are neither of the highest quality nor do they have the highest coverage of the existing machine-readable dictionaries, but they are available in a large number of language pairs and have no (or very few) restrictions on their use. The status as of February 2007, when we downloaded the material, was that the dictionaries contained about 9,200 English words translated into German, 10,600 English words translated into French and 7,600 English words translated into Span-

[9] http://theseus-programm.de/scenarios/en/medico
[10] http://sig.biostr.washington.edu/projects/fm/index.html
[11] http://www.wiktionary.org

ish. As a comparison, the German-English dictionary from LEO[12] contains about 460,000 entries, with the disadvantage that it is not freely accessible. Table 3.1 shows some examples from the English-French part of Wiktionary that we use.

English	French
Aachen	Aix-la-Chapelle
aardvark	oryctérope
...	...
abacus	abaque
abaft	en arrière de, sur l'arrière de, sur larrière
abandon	abandonner
abandoned	abandonné
abandonment	abandon
...	...
abhorrent	répugnant
abide by	se soumettre

Table 3.1: *Examples from the English-French part of Wiktionary.*

[12] http://dict.leo.org

4. Theoretical and Experimental Investigations Regarding Evaluation

Dellschaft and Staab (2006) point out that the existence of methodologies and datasets for evaluation in "batch mode" has played an important role in the success of fields like information retrieval and speech recognition (and this also holds for, e.g., text categorization and statistical machine translation). Also Maedche (2002) expresses the need for standardized datasets, especially cross-language ones, and evaluation measures to go with these datasets. Though there is ongoing work towards establishing benchmark datasets (see Chap. 3) and evaluation metrics in ontology learning, there is still a great need for further understanding and refinement of the measures used today. The current chapter, especially Sect. 4.3, is meant to add some of that sought-after comprehension and clarification to the ontology learning evaluation problem.

4.1 Evaluation Paradigms

We can divide the evaluation approaches for the ontology learning task into two major groups: those measuring the quality of the learned ontology by its conformance to some standard and those measuring the increase in performance of an application which uses the learned ontology as a knowledge resource. In conformance with the mentioned needs for batch mode evaluation, we focus our attention on the former in this thesis and restrict ourselves to giving a brief overview of the latter in Sect. 4.1.2.

4.1.1 Gold standards and human judgments

When evaluating the ontology against a formal standard, we are again faced with a choice between two alternatives. Either we measure the similarity of the learned ontology to a predefined gold standard, or we let human experts look at the learned ontology and decide whether the suggested structure is reasonable or not. The second is typically a much more forgiving task, since people are generally able to interpret the results in a more flexible manner than an evaluation script is.

Using a gold standard to evaluate an ontology learning system brings some inherent problems. A gold standard provides one particular conceptualization

and structuring of a domain. If the learned ontology differs from the gold standard, it could still be a valid model for the domain – just not the same model as suggested in the gold standard. The experiments presented in this thesis are contrastive in nature; they compare the effects of using different learning strategies or different feature sets. This means that we are able to measure an *increase* in performance as the model improves, even though the *absolute* correspondence values may be low for all evaluated models. Again, even though there generally does not exist just *one* way of modeling the domain, the models used as gold standards in this thesis should be sufficiently different from *unstructured* data, that a more effective learning system will capture more of the structure in the gold standards than a less effective system will. The great advantage of the gold standard-based approach is of course that it allows for evaluation on a massive scale; we can produce countless variants of learned ontologies and evaluate them all via the push of a button.

Involving humans directly in the evaluation process introduces two main problems. The first is that we are then moving away from the goal of a batch-like processing of the results, making the evaluation process costly in terms of time and resources. The second is the issue of reproducibility; ask any two human experts to make hundreds of qualitative judgments on a task as complex as the current, and we are unlikely to find two identical series of decisions (this also holds for the same person performing the same evaluatory process at different points in time). These issues become even more problematic when we have a large number of models that we would like to compare against each other (as is the case with the experiments performed in this thesis) – it often is unfeasible to have humans perform evaluations on this scale. The problems of involving humans in the evaluation process for the ontology learning task have also been discussed by Faatz and Steinmetz (2004).

4.1.2 Application-based evaluation

Using a learned ontology in an application and measuring the increase in performance of that application has the advantage of demonstrating the practical benefit of the ontology; a task of some general interest can now be performed better than it could without the learned ontology (or with a learned ontology built with another method). The downside is that we cannot make any *generalizing* statements of the quality of the ontology; all we can say is which ontology is better for performing a particular task. In Sect. 2.1 we gave an overview of the types of applications where ontologies have been put to use; here we look at the same application types from the perspective of how suitable they would be in an evaluation scenario.

Query processing in an information retrieval system

A learned ontology can be used to perform some degree of query modification and the change in performance of an underlying information retrieval system

can be used as a quality measurement. This presupposes the existence of texts, queries and relevance judgments from the relevant domain – something which severely limits the applicability of this evaluation approach. A further complication is that this kind of query processing far from always has proven helpful (see Sect. 2.1), which makes this application less attractive from an evaluation standpoint.

Text categorization

We reported on the successful application of ontologies to the text categorization problem in Sect. 2.1. It has not been shown, however, that the *hierarchical* structure of the learned ontology is what brings the advantage, as opposed to the information available from a simple flat clustering of the terms. This casts some doubts on this evaluation scenario, along with its being dependent on the existence of a domain-specific document collection categorized by hand.

Question answering

In Bloehdorn et al. (2007) and Buitelaar et al. (2006), the authors demonstrate the use of an ontology inside a question answering system. For question answering to be used as an evaluation scheme, we would again need a set of documents, questions and correct answers from the appropriate domain. The fact that the mentioned question answering systems make explicit use of the hierarchical structure of the ontology when answering questions, makes this one of the more promising approaches for evaluation, given the needed resources.

4.2 Evaluating Term Clustering

Term clustering does not form a necessary part of a generic ontology learning system. However, if we consider learning a prototype-based ontology (see Sect. 1.1), using a hierarchical clustering method based on distributional similarity, an approach for evaluating different settings in the clustering environment would be of great use. Further, it will not always be the case that we want to form a complete hierarchy when clustering terms. Sometimes the flat clusters are all that is needed, and for these cases it would be useful to have an evaluation procedure.

To evaluate the quality of the clusters produced, the question is what constitutes a good cluster? The goal of the clustering is to capture groups of semantically related terms, but what should be considered related and what not?

There are a number of *intrinsic* cluster evaluation methods; methods which do not compare the resulting clusters to any outside source, but instead try to quantify clustering-internal attributes. But do these intrinsic measures agree with human judgments on what is considered a good clustering of terms? This is the main question we will examine in this section.

We propose to compare the results from the intrinsic methods to those of an *extrinsic* evaluation method, which compares the results of the automatic clustering to the implicit clustering provided by an ontology. This allows us to apply well established evaluation techniques from *document* clustering to the *term* clustering evaluation problem. We perform a series of experiments on the Wikipedia anatomy corpus (see Sect. 3.2.2), using the FMA ontology (Sect. 3.3.2) as our reference ontology.

4.2.1 Forming term clusters from ontologies

Our extrinsic evaluation approach is based on work in evaluation of document clustering; Rosell (2005) provides a detailed description of how the measure is used in that setting. The method relies on the existence of a gold standard, and we form our gold standard clusters by placing a divisive cut at some level in an ontology (see Fig. 4.1). Each node at the level of the cut forms the basis of a separate cluster. All nodes (with their associated terms) that are dominated by a particular base node are assigned to the cluster of that base node.

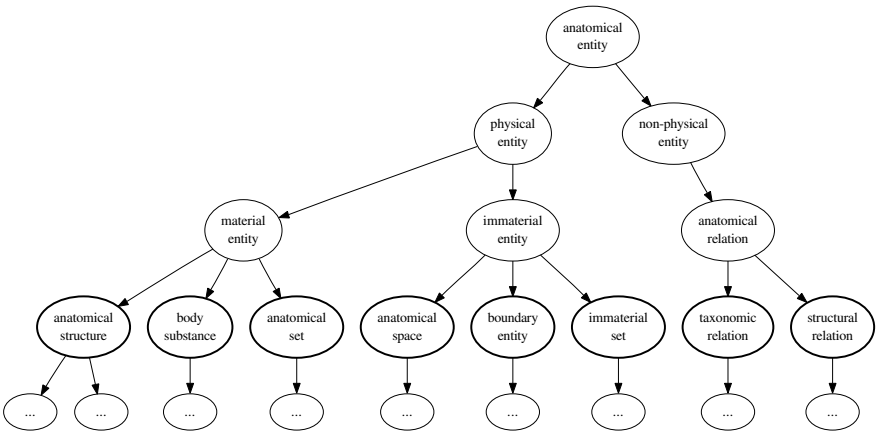


Figure 4.1: Ontology split into clusters: a small, slightly modified excerpt of the FMA ontology, where only the first four levels are shown. Concepts marked in bold (second level from the bottom) constitute the basis of the clusters. Concepts higher up in the hierarchy than this base level are disregarded.

4.2.2 Evaluation measures

We introduce three different intrinsic evaluation measures and compare their results on a set of evaluation tasks to the results of the previously mentioned extrinsic measure.

Cohesion: an intrinsic measure, which gives a value for how "tight" a cluster is, i.e., how closely together the cluster members are in vector space. Each

cluster is represented by a centroid, a vector with the averaged feature values from all cluster members. For each cluster, we calculate an average similarity value (cohesion) of all the cluster members with this centroid – the higher the value, the "tighter" the cluster. To evaluate the whole clustering, we calculate the average cohesion value of all clusters:

$$\frac{\sum_{c_i \in C} \sum_{t \in c_i} sim(t, \bar{c_i})}{|t|}$$

C is the set of all clusters, t is a term, $\bar{c_i}$ is a cluster centroid and *sim* is a similarity function (we use *cosine*).

Separation: also an intrinsic measure, which gives a value for how far apart from each other the clusters are in vector space. We calculate the average distance between all cluster centroids:

$$\frac{2 \cdot \sum_{1 \leq i < j \leq |C|} 1 - sim(\bar{c_i}, \bar{c_j})}{|C|^2 - |C|}$$

Cohesion and separation both give values in the range $[0,1]$. Both measures are discussed further in Rosell (2005).

F-measure: in an effort to balance cohesion and separation against each other, we calculate the harmonic mean between these two measures:

$$\frac{2 \cdot cohesion(C) \cdot separation(C)}{cohesion(C) + separation(C)}$$

We name the measure in analogy with the F-score used to balance recall and precision in information retrieval.

Mutual information (MI): this is the only extrinsic evaluation measure we investigate. We use the clustering results from the splitting operation (see Fig. 4.1) as our gold standard clustering. Obviously, we cannot expect our cluster c_i to correspond to ontology cluster oc_i (*oc* for "ontology cluster") other than by pure chance, since the numerical ordering of the clusters is arbitrarily made. What we *can* hope for is that as many terms as possible that are clustered together in the ontology also will be clustered together in our automatic clustering. We now form a matrix M, where the rows correspond to the clusters in our automatic clustering and the columns correspond to the ontology clusters. We next number the rows $1 \ldots \gamma$ and the columns $1 \ldots \kappa$. We let n_i denote the number of terms in cluster c_i and n^j the number of terms in cluster oc^j. Finally, we let m_i^j mean the number of terms that are shared by clusters c_i and oc^j. We can now calculate:

$$I(C;OC) = \sum_{i,j} \frac{m_i^j}{n} \log \frac{m_i^j n}{n_i n^j}$$

where n stands for the total number of terms clustered. We normalize this measure by dividing it by $\frac{\log(\gamma\kappa)}{2}$. This evaluation method was introduced for document clustering in Strehl et al. (2000).

4.2.3 Comparing the results

We split the FMA ontology at a level that gives us 63 separate clusters. We limit the comparison to terms that occur at least 50 times in the corpus, meaning a total of 1,164 English terms – this threshold is set in order to ensure that the clustering is performed on sufficiently high quality data. We then cluster the term vectors using k-means clustering, setting k to 63, to match the number of clusters in the gold standard.

In Sect. 2.4.1 we list five different parameters that can be varied easily when working with distributional similarity models (the list is non-exhaustive but representative). We run a series of experiments where each of these parameters is varied while the others are kept constant, resulting in a total of 39 experiments. We evaluate the results of each experiment using each of our four evaluation methods, and we plot the results of the three intrinsic methods against the extrinsic method in Figs. 4.2–4.4. The correlation coefficient for each intrinsic method compared with the extrinsic method is given in Table 4.1.

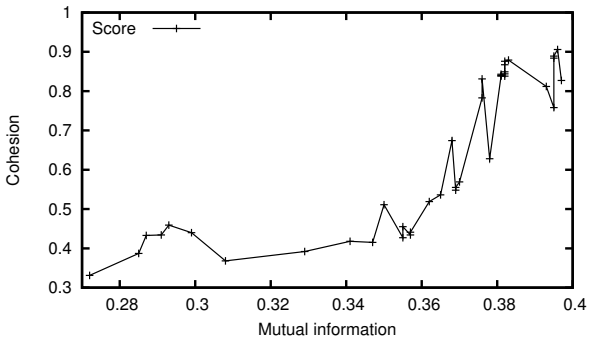


Figure 4.2: The cohesion values plotted against mutual information.

	Cohesion	Separation	F-value
Correlation with MI	0.811	-0.629	0.815

Table 4.1: *Correlation coefficients for the three intrinsic measures compared with the extrinsic MI measure.*

We see that there is a strong positive correlation for both the cohesion and the F-value measures with the MI measure, and a negative correlation for the separation measure. However, it seems the separation measure is able to can-

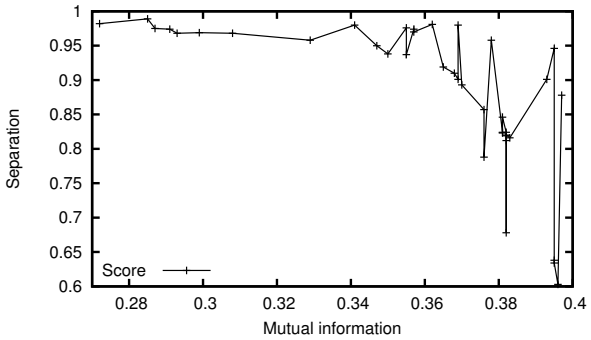


Figure 4.3: The separation values plotted against mutual information.

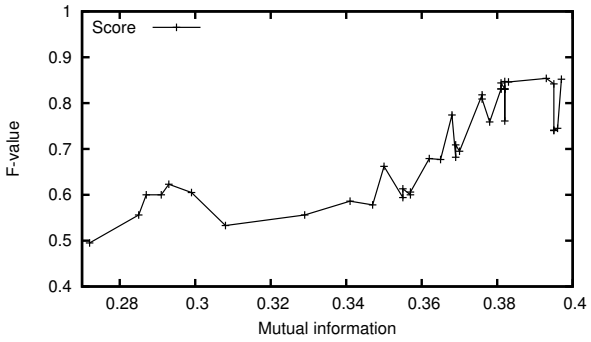


Figure 4.4: The F-values plotted against mutual information.

cel out some outliers in the cohesion measure and that the F-value measure is able to profit from this; hence the slightly higher correlation for the F-value measure than for the cohesion measure. Overall, though, the difference in correlation between the cohesion and the F-value measures is not big enough for us to draw any definite conclusions as to which is more strongly correlated with the extrinsic method, and therefore also in extension with the human-created clustering.

4.2.4 Interpreting the results

All terms in an ontology cluster are related by a hyperonymy or a hyponymy relation, or by the repeated application of these two relations (which then also covers cohyponyms). These lexico-semantic relations show different facets of what people in general mean when they say that two terms are similar. The clustering provided by the ontology thus seems to be a sensible reference. Granted, most of the words in a cluster will be related via a topical connection rather than directly through one of the previously mentioned lexico-semantic relations. Still, words in the same ontological cluster will typically have a

shorter "semantic distance" to the words in its own cluster than to words in other clusters, and this is what we are after here.

The negative correlation for the separation measure may seem surprising. Remember though, that one parameter that was kept constant throughout these experiments was the number of clusters produced by the k-means clustering algorithm. Should we additionally have been interested in trying to find an appropriate k for our data, it seems more natural that the separation measure would prove useful. Consider the extreme case, where each term is its own cluster – here we would expect a rather low average separation but a very high cohesion (since the only cluster member would in fact be identical to the cluster centroid). Gradually decreasing the number of clusters would then give us an evening out of the two measures against each other. We suspect that the F-measure would be of most value also when trying to find an appropriate k for a particular set of terms, since the F-measure does exactly this; it balances the cohesion and separation values against each other. Though this remains to be verified experimentally, we use this intuition in our experiments in Sect. 6.1.

Our stated purpose for performing these experiments was to see whether the readily available intrinsic cluster evaluation measures coincide with human similarity judgments. The strong correlation demonstrated here between a human source for relevance judgments, provided via the ontology, and the clustering-internal, intrinsic, evaluation measures demonstrate that this is the case, to a large degree. In most cases where we wish to perform term clustering, the intrinsic measures will in fact be our only option, since the existence of a domain ontology would reduce the need for performing an automated clustering. Our results thus point towards the intrinsic measures providing good indications as to which clustering is to be preferred, in a real world scenario.

4.3 Evaluating Ontology Learning

The previos section showed different possibilities of evaluating term clusterings. If we wish to include the hierarchical structure of an ontology in our evaluation, we need another approach. We are thus interested in finding a metric for measuring the similarity between two ontologies. A first suggestion towards this goal within the ontology learning community was given in Maedche (2002). The measures *taxonomic precision* (TP), *taxonomic recall* (TR) and TF (the harmonic mean of TP and TR) were introduced there. These measures make use of what Maedche calls *conceptual cotopy* or *semantic cotopy* (SC). The SC of a concept in a hierarchy consists of all its superconcepts, its subconcepts and the concept itself. To separate the evaluation of the overlap of the lexica of the two ontologies from the evaluation of the taxonomic structures, Dellschaft and Staab (2006) and Cimiano (2006) use *common semantic cotopy* (CSC), which disregards all concepts that are not found in *both* ontologies.

These measures have been applied in a number of different settings and are starting to establish themselves as the standard measures for evaluating an ontology learning system. We argue here that they are not applicable to all settings where such an evaluation measure is needed and also that they do not always behave in a predictable manner (see Figs. 4.7–4.9). We also introduce a new evaluation measure, based on Pearson's product-moment correlation coefficient (PMCC).

We follow Dellschaft and Staab (2006) in treating concepts and the terms used to represent them as one and the same, assuming a one-to-one relationship between the two. This is not crucial to the arguments presented, but is meant to simplify the discussion.

4.3.1 The PMCC evaluation measure

Oxhammar (2007) uses Spearman's rank correlation to evaluate how well an ontology enrichment algorithm manages to add terms to an existing ontology, when comparing to a gold standard. We propose to use the related PMCC measure for the problem of calculating a similarity score between two ontologies. In Sect. 4.3.3, we show why using PMCC is more fitting for the current task, than using Spearman's rank correlation.

The idea behind the PMCC measure is that we can characterize an ontology by listing all pairs of concepts that it contains, along with the distance between each concept pair, measured in the number of edges between the two concepts. E.g., for the ontology in Fig. 4.5, we get the following distances:

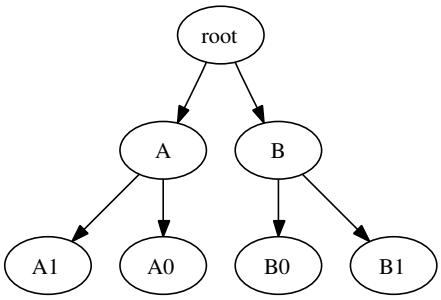


Figure 4.5: Small example ontology.

```
A0 -> A:    1
A0 -> A1:   2
A0 -> root: 2
A0 -> B:    3
A0 -> B0:   4
A0 -> B1:   4
A1 -> A:    1
```

```
A1 -> root: 2
A1 -> B:    3
A1 -> B0:   4
A1 -> B1:   4
A  -> root: 1
...
```

Note that if we have listed the distance $A0 \rightarrow A1$, we do not need to list the distance $A1 \rightarrow A0$, since it will be identical.[1] Neither are we interested in the distance between any concept and itself, and also not in distances between concepts that do not occur in *both* the reference ontology and the learned ontology (analogous to the CSC measure). This means that if the two ontologies share n concepts between them, we need to calculate a total number of $\frac{n^2-n}{2}$ distances per ontology. Once we have calculated these series, we can use them to calculate the PMCC measure:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{4.1}$$

where X is the series from the learned ontology and Y the series from the reference ontology, *cov* stands for covariance and σ is the standard deviation. The measure is symmetrical, so we could swap X and Y and get the same result. The measure returns a value between -1 and 1, where 1 means perfect correlation, 0 means no correlation and -1 means perfect negative correlation, somewhat simplified. A negative value thus means that long distances in one ontology correspond to short distances in the other, and vice versa. Note that a high negative value also would be of interest, since we then would have a method which consistently makes wrong decisions, and to get a functioning system we should do the opposite of what the original system suggests.

PMCC assumes that the series being compared are normally distributed. We plot the distribution of distances in the Eurovoc thesaurus, according to the method described for Fig. 4.5 above, in Fig. 4.6. The plot fits a normally distributed curve nicely, and we take this as an indication that assuming a normal distribution is reasonable, in the general case.

4.3.2 Evaluating the measures

Just from reading the descriptions of these measures, it is not necessarily clear how the different measures react to different types of input. To remedy this, we have constructed a number of tests where we aim to make the differences in behavior evident.

In Dellschaft and Staab (2006), three main criteria are listed for determining what is a good evaluation measure:

[1] We treat the ontology as a simple directed acyclic graph.

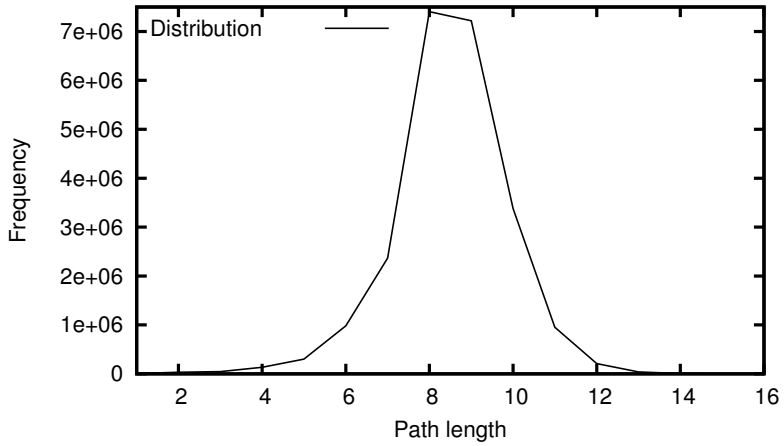


Figure 4.6: Distribution of path lengths in Eurovoc.

- **Criterion 1 – Independent dimensions of evaluation**: The measure should allow the term extraction part of the ontology learning system to be evaluated independently from the part that learns the hierarchical structure.
- **Criterion 2 – Severe errors should have a higher impact on the measure than less severe ones**: E.g., an error near the root of the ontology should have a bigger effect than an error further down in the tree.
- **Criterion 3 – A gradual decrease in correctness should result in a gradual decrease in the value of the evaluation measure**: This relates to the previous point, but emphasizes a linear, predictable behavior of the measure.

To clarify Criterion 1, consider that in a complex, multilayered system like the ones used in ontology learning, one has to take into account the effects of error propagation, where errors in a previous module will cause later modules to function worse than if they had been fed correct input. To eliminate the effects of error propagation between modules during evaluation, one should consider each subtask in separation of the others and this is what Criterion 1 stresses. This criterion is trivial for the PMCC measure: since we only consider concepts shared by both ontologies, the quality of the term extraction (often the first module in an ontology learning system) does not directly influence the result. To evaluate the term extraction component, any version of precision and recall, such as described in, e.g., Maedche (2002), could be used. The following pages present a series of experiments designed to test how far the different measures meet the other criteria.

In addition to the criteria from Dellschaft and Staab (2006) listed above, we would like to add the following:

- **Criterion 4 – Scaling**: Often the learned ontology will only contain a subset of the concepts in the gold standard. This can in turn lead to large differences in average path lengths between concepts in the two ontologies. A good measure should be able to abstract away from such differences in scale.
- **Criterion 5 – Vertical and horizontal perspective**: The vertical dimension, typically representing an is-a inheritance structure, is of course a key characteristic of an ontology. However, much information is also coded in the horizontal plane, where the sibling information (often representing cohyponymy when dealing with terms) can be read. A good measure should consider both dimensions.

We present experiments for testing these additional criteria, and we finally also look at the behavior of the measures in a typical machine learning scenario.

Experiments with the Eurovoc thesaurus

For the first series of tests, we are using the Eurovoc thesaurus (see Sect. 3.3.1) as the reference ontology. We perform randomized distortions on this structure and see how the different measures respond to the changes. Counting the root node, domain and micro-thesaurus information, the Eurovoc taxonomic structure has 6794 concepts. Because our first three experiments involve a random element, they were all repeated ten times, using different random initializations.

Experiment 1 – Randomized scrambling: Starting off with two identical ontologies as the learned and the reference ontology, we gradually introduce a randomization factor into the "learned" ontology. We start by randomly selecting 10 percent of the concepts in the ontology and have them switch places, also in a random fashion. We then increase the degree of randomized concepts stepwise, by 10 percent for each step, up to 100 percent.

Fig. 4.7 shows a linear decrease for the PMCC measure, which is in accordance with Criterion 3, listed previously. The SC measure starts flattening out at about 30 percent randomization and the CSC at perhaps 40 percent, although it starts out with a steeper descending curve than the other two. This flattening out of the curves is unfortunate, especially if you have a learned ontology which is noisy, since differences may be hard to detect in this flattened, low accuracy area of the curve.

Experiment 2 – Randomized scrambling, neutral root: In the previous experiment, the root node was kept the same for both ontologies and excluded from the randomization. In this experiment, the root node in the learned ontology was switched to a neutral concept, which does not occur anywhere in the reference ontology (this new root was kept constant throughout this set of experiments). Fig. 4.8 shows that the PMCC measure is unaffected by this change. The SC measure has a sharp drop in accuracy when no

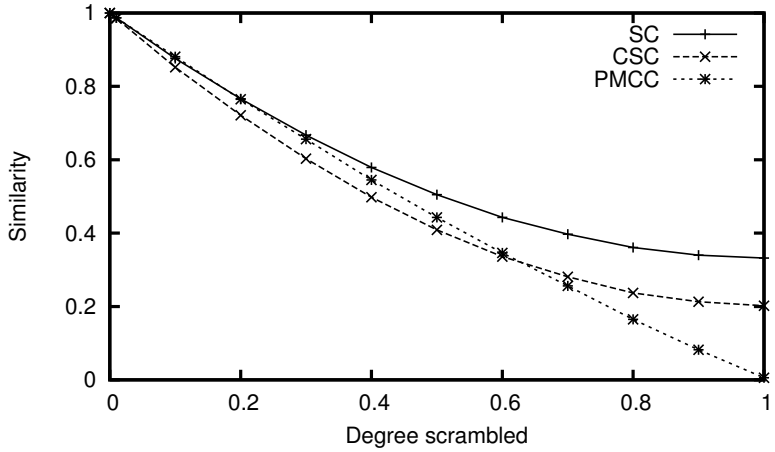


Figure 4.7: Increasing level of randomization.

randomization has been performed. The curve for the CSC measure drops even more steeply than in Fig. 4.7, before starting to flatten out at around 60 percent randomization.

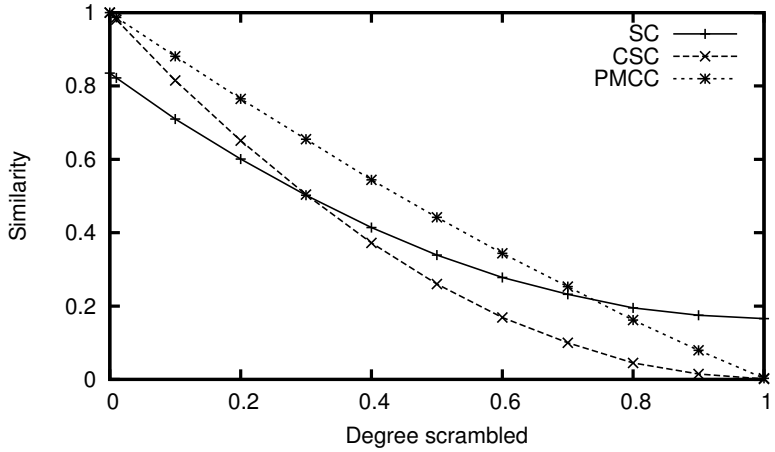


Figure 4.8: Increasing level of randomization, neutral root.

Experiment 3 – Randomized scrambling, switched root: This experiment is similar to the previous two, but here we let the root and a (randomly picked) concept occurring somewhere in the reference ontology switch places. (The same randomly picked concept was kept as root throughout Experiment 3.) Figure 4.9 shows that the PMCC measure again is unaffected. The SC measure has a similar behavior as it had in Fig. 4.8, but the CSC measure now has the same sharp drop in accuracy where no randomization has taken place,

as the SC measure had in the previous experiment. Experiments 2–3 show an unwanted sensitivity to changes in the root node for the SC and CSC measures, which is not shared by the PMCC measure.

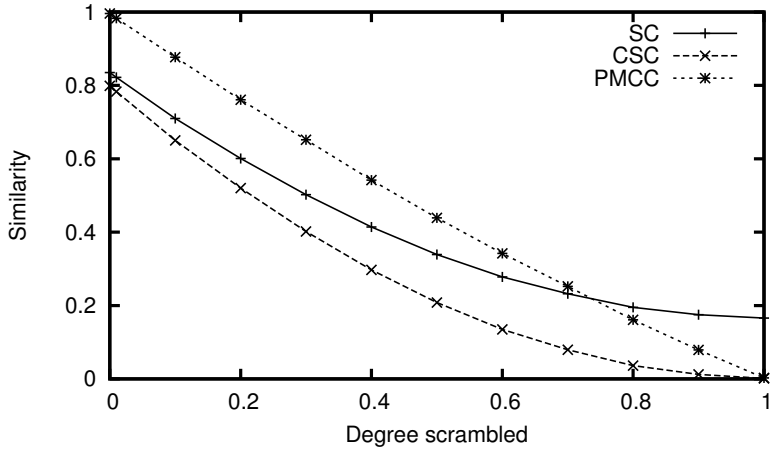


Figure 4.9: Increasing level of randomization, switched root.

Experiments with ontologies designed for evaluation

We have constructed an artificial reference ontology, shown in Fig. 4.10, for checking the behavior of the different measures in a number of situations, simulating different types of learning errors and learner behavior.

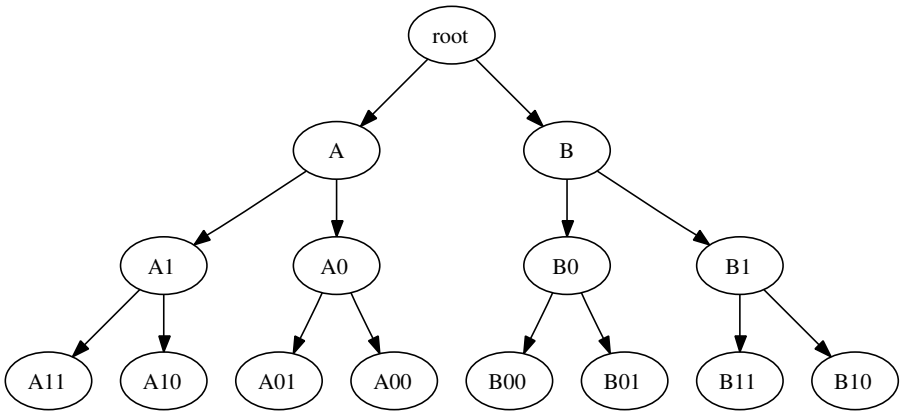


Figure 4.10: Reference ontology.

Experiment 4 – Concept displacement on high or low level in the ontology: Misplacing concepts that belong high up in the ontology is intuitively more serious than misplacing concepts belonging further down in the ontology. We have made two different alterations to the reference

ontology: the first has two misplaced concepts at the lowest level (Fig. 4.11) and the second has two misplaced concepts at the top level, below the root (Fig. 4.12). This experiment corresponds to the previously listed Criterion 2, that is, we would like the misplacement closer to the root to cause a bigger drop in similarity than the misplacement at the lowest level. Table 4.2 indicates that this is captured by all three measures.

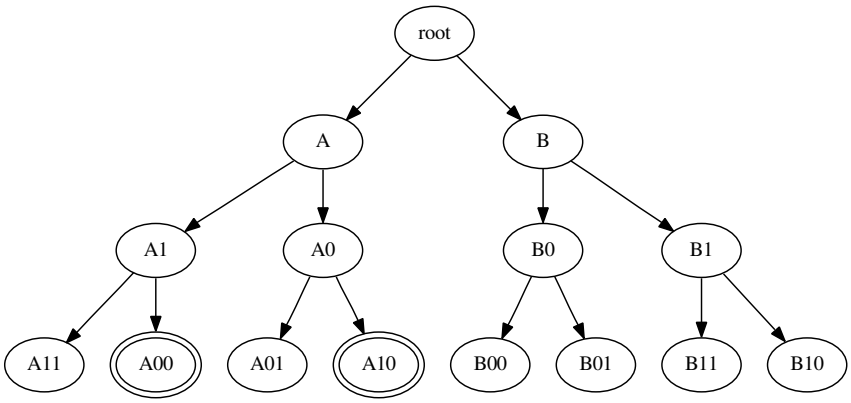


Figure 4.11: Ontology with concepts misplaced on the lowest level.

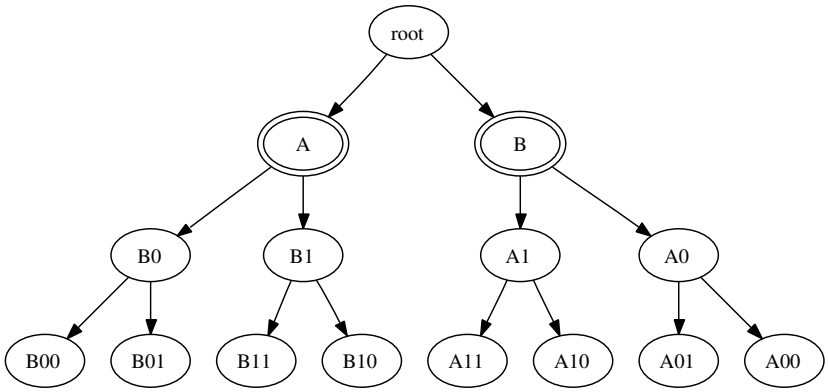


Figure 4.12: Ontology with concepts misplaced at the top.

Experiment 5 – Scaling: This experiment means to test our additional Criterion 4, which deals with scaling. One can easily imagine a situation where, in addition to the concepts on which we wish to focus our evaluation, the reference ontology contains layers of nodes which are abstract in nature and do not show up in the learned ontology. In these cases, we still want to be able to use this reference ontology for evaluation and get a high score if the same basic structures are captured. The elongated reference ontology used for this evaluation is depicted in Fig. 4.13, while our previous reference ontology

	SC	CSC	PMCC
Bottom displacement	0.940	0.922	0.942
Top displacement	0.713	0.641	0.826

Table 4.2: *Comparing scores for misplacement at the top or at the bottom of the ontology.*

from Fig. 4.10 serves as the learned ontology for this experiment. The results are presented in Table 4.3. The SC measure does not handle this situation well, it gives a low similarity score to the two ontologies, whereas the CSC measure gives a perfect score, which conforms better with Criterion 4. The PMCC measure copes well with the additional layers; the score indicates close to perfect similarity.

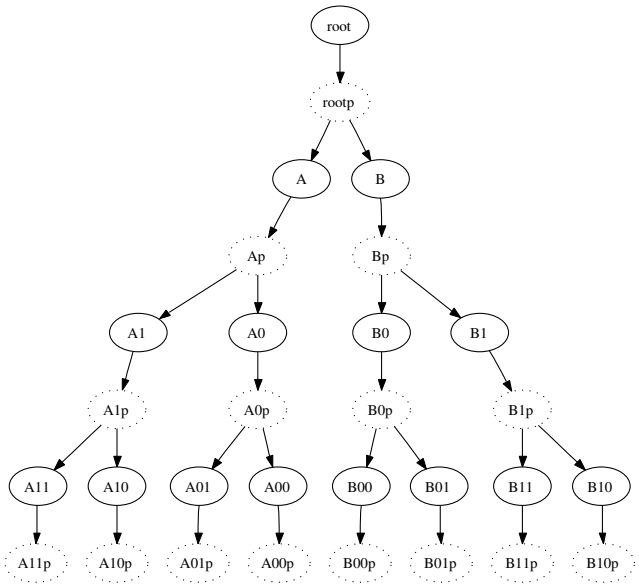


Figure 4.13: Ontology with additional layers of abstract concepts.

	SC	CSC	PMCC
Elongated ontology	0.400	1	0.969

Table 4.3: *Similarity score when using a reference ontology with interspersed abstract concepts.*

Experiment 6 – Horizontal relations: In this case we use the ontology in Fig. 4.14 as our reference. Figure 4.15 does a good job of grouping siblings together, but assigns the wrong "top" node to each subtree. The ontology in Fig. 4.16 preserves some of the hierarchical relations, but this is merely by chance; the leaf nodes are really just distributed in an alternating pattern. For a human correcting the output of an ontology learning system, the first structure would be much preferable to the second, since it involves correcting just two nodes, whereas six nodes need correction in the second structure. Cimiano (2006) suggests a measure he calls *sibling overlap*, designed to measure this type of correlation between structures. We believe a measure which considers both the vertical *and* the horizontal relations in an ontology would be even more useful. Table 4.4 shows that the PMCC measure prefers the ontology with a higher degree of preserved horizontal relationships, whereas the other two measures prefer the ontology with the alternating leaf nodes.

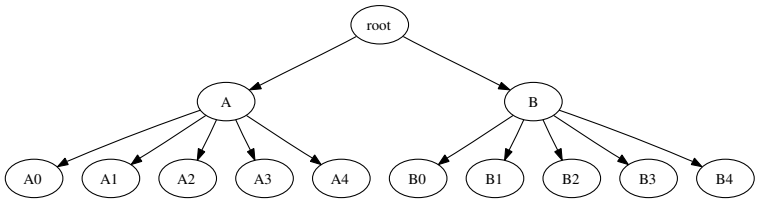


Figure 4.14: Reference ontology for horizontal relations.

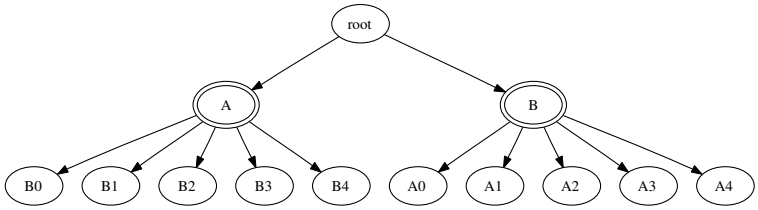


Figure 4.15: Ontology with "switched" parent nodes.

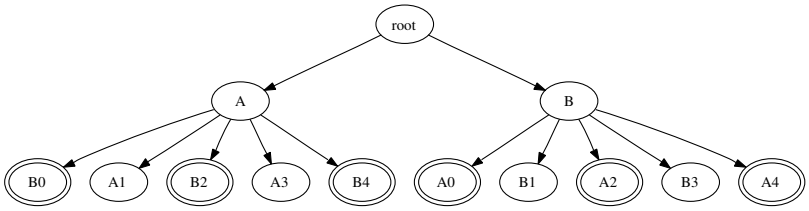


Figure 4.16: Ontology with alternating leaves.

	SC	CSC	PMCC
Switched parents	0.634	0.487	0.567
Alternating leaves	0.780	0.692	0.221

Table 4.4: *Comparing scores for preserving horizontal relationships.*

Experiment 7 – Hierarchical clustering: Fig. 4.17 is typical of the structure one gets when some kind of hierarchical clustering is used. The non-leaves in such a structure are abstract; they have no single label, but are usually considered as a set containing all subnodes. E.g., node '3' in Fig. 4.17 would consist of the set {*B00, B01*}.

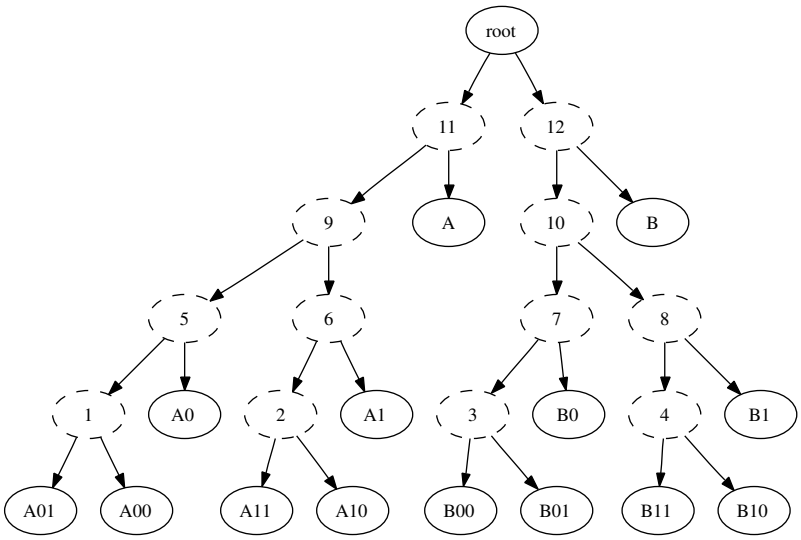


Figure 4.17: Ontology resulting from hierarchical clustering.

We see now that when we take the SC of any concept in Fig. 4.17 and compare it with its *corresponding* concept in Fig. 4.10, we will always get a complete mismatch; no two concepts in the SCs of the concepts under evaluation will be the same (except for the concept itself and the root, which is trivial). E.g., the SC of 'B00' is {*B00, B0, B, root*} in the reference ontology and {*B00, 3, 7, 10, 12, root*} in the learned ontology.[2] If we take the CSC of the same concept, we get {*B00, root*} for the learned ontology and again only the concept itself and the root are shared with the reference, which also is trivial. In fact, Cimiano (2006) notes that because the concept itself is included in the cotopy, a trivial ontology where every concept is placed directly

[2]Instead of using numbers, we could of course use the set of terms as labels, e.g., instead of '3' we could use {*B00, B01*}. This does not make any difference to our discussion, since sets do not occur in the reference ontology any more than do numbers.

under the root would score very high using the CSC measure (see Cimiano, 2006, p. 105). Cimiano then suggests to exclude the concept itself from the cotopy. This means that all leaf nodes in Fig. 4.17 will have empty CSCs (except for the root); they will not contribute anything to the evaluation. Cimiano concludes that these concepts should be excluded; only the concepts *not* occurring in both ontologies should be used in the evaluation (here meaning {*1–12*}). For the remaining concepts, an optimistic estimation is made, where the concept in the reference ontology with the greatest overlap in CSC is chosen for comparison. The new formula for Taxonomic Overlap (TO') looks like this:

$$TO'(O_1,O_2) = \frac{1}{|C_1 \setminus C_2|} \sum_{c \in C_1 \setminus C_2} max_{c' \in C_2} TO'(c,c',O_1,O_2) \tag{4.2}$$

$$TO'(c,c',O_1,O_2) = \frac{|SC'(c,O_1,O_2) \cap SC'(c',O_2,O_1)|}{|SC'(c,O_1,O_2) \cup SC'(c',O_2,O_1)|} \tag{4.3}$$

$$P_{TO}(O_1,O_2) = TO'(O_1,O_2) \tag{4.4}$$

$$R_{TO}(O_1,O_2) = TO'(O_2,O_1) \tag{4.5}$$

where SC' stands for the new version of SC, where the concept itself is excluded from its cotopy. One intuitive objection to this is that we now force a comparison between "abstract" nodes in the learned ontology and "concrete" nodes (represented by a single term) in the reference ontology, even though we "know" that these are not the nodes corresponding to each other. E.g., we see that the concept *B00* in the reference ontology corresponds to the concept *B00* in the learned ontology, but we are not allowed to make this mapping.

A more serious problem arises when we consider the $R_{TO}(O_1,O_2)$ measure above. If we take $\{C_{ref} \setminus C_{learned}\}$, we get the empty set – we have no concepts on which to calculate the recall part of the TO'. This is not apparent from the description in Cimiano (2006), since the author there assumes that the reference ontology will contain abstract nodes as well. This we consider to be the most serious problem with the TO' measure – that it is not applicable in these cases.

The results for evaluating the hierarchical clustering are displayed in Table 4.5. This is about as close to the reference ontology as we can get, using hierarchical clustering (in that concepts that are close in the reference are also close in the clustering), which is captured by the PMCC measure but not by the other two. (The CSC measure tested is the one proposed in Dellschaft and Staab, 2006, not Cimiano's altered version discussed above.)

	SC	CSC	PMCC
Hierarchical clustering	0.308	0.496	0.956

Table 4.5: *Similarity scores for hierarchical clustering.*

4.3.3 Alternative approaches and implications for ontology learning evaluation

An alternative to using PMCC would have been to use Spearman's rank correlation. That would mean that, instead of entering distances between nodes in the table below Fig. 4.5, one would enter the ranking for each term pair. All term pairs with distance 1 would get ranking 1 and all pairs with the same ranking form equivalence groups. Looking again at the example ontology in Fig. 4.5, we see that there would be six pairs with distance 1, meaning that pairs with distance 2 will get rank 7 and so forth.[3] If we apply this to an ontology such as the one used in the Eurovoc experiments, we get very big equivalence groups, containing millions of members, as can be seen in the distribution in Fig. 4.6. This means that a length difference of 1 could result in a rank difference of several million and this is not what we want. In statistics literature, the recommendation is to *not* use Spearman's rank correlation when the data contains many ties (Bassett, 2000), which is in line with our reasoning here.

The PMCC measure we suggest bears some resemblance to the method proposed in Brank et al. (2006). Their *OntoRand index* measure calculates average distances between *instances* in the ontology rather than considering the concepts themselves. They require the sets of instances in the gold standard and in the learned ontology to be identical.

Resnik (1998) and Budanitsky and Hirst (2006) discuss a number of alternatives to the absolute path length for measuring the distance between two concepts in a taxonomy. One intuitive notion these measures aim to capture is that two concepts at the leaf level sharing the same parent are more similar than two concepts sharing the same parent higher up in the taxonomy. This certainly makes sense if we wish to model similarities between concepts in an ontology. Here we want to do something different, namely measure the similarity between two ontologies. As shown in Experiment 4, the PMCC measure reacts stronger to displacements at the top of the ontology, using the standard path length measure – switching to a depth-sensitive measure therefore adds unneeded complexity to the method while introducing a degree of unpredictability by trying to solve the same problem at two different places at once.

[3] Rankings within groups are typically averaged, meaning that the group with distance 1 would be ranked 3.5. This is not crucial to the discussion here though.

Wishing to measure the similarity of ontologies/tree structures is not something indigenous to the ontology learning community. Schilder et al. (2005) calculate similarities between parse trees and Jin et al. (2005) compare RNA structures. Common to many of these approaches is their indebtedness to the tree-to-tree correction problem (Tai, 1979), see Barnard et al. (1995) for a survey. In tree-to-tree correction, the similarity measure is based on the number of edit operations necessary to get from one tree to the other. The allowed operations are *insert*, *delete* and sometimes also *switch*.

One issue with applying this approach to our particular problem is that we no longer make distinctions between large displacements and small. We would expect an ontology where a concept has been moved a small distance from its original place in the gold standard to get a higher similarity score than an ontology where the same concept has been moved a larger distance. This is not the case using the principles of the tree-to-tree correction problem. Further, it seems intuitive to give a mistake in the lower, most specific, levels of an ontology less importance than a mistake at the top of the ontology (Criterion 2). This is also not captured by the tree-to-tree correction approach, whereas the three approaches compared in this paper all make this distinction.

Another aspect that sets the PMCC measure apart from the tree-to-tree correction approach is its ability to handle differences in scale – Criterion 4. The CSC-based measures share this ability, the SC-measures do not, as can be seen from the numbers in Table 4.3. Imagine that we are working with an extensive gold standard, containing thousands or tens of thousands of concepts, but our learning algorithm only handles a fifth of these. If this subset is more or less evenly spread out over the gold standard, the PMCC measure can abstract away from this difference of scales, simply by assigning the proper k-value in the $y = kx + m$ linear transformation. The tree-to-tree correction approach falls short of detecting this relation. Similarly, our PMCC measure handles constant length differences between the ontology through the value of m. The tree-to-tree correction approach would also fall short of detecting the similarities between the gold standard and the learned ontology shown in Fig. 4.17, where the learned ontology stems from a hierarchical clustering system.

Cimiano (2006) suggests another approach that makes use of graph similarity measures, such as presented in Chartrand et al. (1998). Chartrand et al.'s formula for calculating a distance between two graphs $G1$ and $G2$ looks like this:

$$d(G1, G2) = \sum |d_{G1}(u, v) - d_{G2}(u, v)|$$

where the sum is taken over all unordered pairs of vertices (u, v). This seems more promising (and in fact very similar to the approach proposed here) than using the tree-to-tree correction approach. One problem with this is that calculating a maximum graph distance can be a computationally very costly procedure. Without a maximum, it is not possible to normalize the measure, which

in turn would make it hard to use for comparing different ontology learning approaches in different settings.

It is true that the PMCC measure does not distinguish between distances going up or down in the ontology. It is even possible to construct different ontologies which are indistinguishable to the PMCC measure.[4] However, this is only possible as long as all concepts in the ontology (apart from the root) have no more than one subconcept, which is a type of ontology one would expect to see rarely, if ever. In all other cases, even though the distance between any two concepts that change places in the same "is-a chain" (where one concept dominates the other) remains the same, their respective distances to other concepts in the ontology do not, meaning that any such alteration still would be detected in the overall similarity score.

Returning to the criteria for what constitutes a good evaluation metric presented in Sect. 4.3.2, we now have a better idea of how the different measures stand up. The independence criterion (Criterion 1) is handled by the PMCC and the CSC measure, but not by the SC measure (Dellschaft and Staab, 2006). All three measures show a bigger reaction to concepts swapped at the top of the hierarchy than to concepts swapped on leaf level (Criterion 2). Only the PMCC measure displays a linear *decrease* in its value for a linear *increase* in noise, or randomization (Criterion 3). For the two additional criteria suggested in this paper, we have seen that the CSC and PMCC measures handle scaling well, whereas this is not the case for SC. The PMCC measure is the only measure of the three explicitly taking horizontal relations into account in addition to the vertical relations. Only the PMCC measure detects the similarity between the simulated hierarchical clustering result and the gold standard in Figs. 4.17 and 4.10.

The PMCC measure is the only measure tested here that meet all five listed criteria. We have also shown that it can be used in situations where currently available measures cannot. It therefore constitutes a valuable alternative and complement to the already established SC and CSC measures, for the evaluation of ontology learning systems.

[4] Note that the same is true also for measures based on the SC and CSC measures.

5. Identifying Cross-language Term Equivalence

The experiments presented in this chapter deal with the identification of cross-language term equivalents, a topic interesting for its applicability in a number of language technology fields. The most obvious application is the automatic construction of domain-specific bilingual dictionaries. Such dictionaries are used in many different settings, including rule-based machine translation and cross-language information retrieval, where some approaches rely on the existence of bilingual dictionaries for translating queries. The main reason for looking at this problem in this thesis is, as mentioned in Sect. 2.2, to be able to automatically create a cross-language terminological ontology.

If we want to automate the ontology learning process completely, we need to be able to extract the terms relevant to a particular domain in an automatic manner. Given a collection of domain-specific documents, we want to identify the textual units that constitute the terms of the domain (document collection). This process is referred to as *term extraction* (Castellví et al., 2001; Jacquemin, 2001). Of course, for term extraction to constitute a viable first step in the ontology learning process, we need the recall and precision of the term extraction system to be rather high – otherwise all further processing will be contaminated by error propagation. We therefore start by performing a term extraction experiment, to see where we stand in this matter.

5.1 Term Extraction

In order not to reinvent the wheel, we use tools already available for term extraction. For English, we use a publicly available tool named TermExtractor (Sclano and Velardi, 2007), developed at the University of Roma "La Sapienza". For other languages, a comparable, publicly available tool does not exist. We instead take a simplistic approach for the non-English languages; we use the mutual information measure for deciding what is to be considered a term (see Castellví et al., 2001, for a list of term extraction systems using mutual information). The scores of the mutual information measure are based on term frequencies in the Wikipedia anatomy corpus (see Sect. 3.2.2) and a contrasting reference corpus consisting of newspaper texts. The contrasting corpus consists of excerpts from the Reuters Corpus, volumes 1 and 2 (Lewis et al., 2004), and is about ten times the size of the

	FMA terms	Extracted terms	In common	Precision	Recall
English	6047	4075	506	0.124	0.084
German	88	2657	68	0.026	0.773
French	754	2498	114	0.046	0.151
Spanish	149	2888	77	0.027	0.517

Table 5.1: *Term extraction evaluation, Latin terms not included. The first column states how many of the terms in the FMA ontology occur at least once in the text, the second column the number of terms extracted by the different tools and the third column how many terms are in the intersection between the previous two sets. Columns four and five state the precision and recall for each language.*

	FMA terms	Extracted terms	In common	Precision	Recall
English	7703	4075	537	0.132	0.070
German	1817	2657	126	0.047	0.067
French	1174	2498	136	0.054	0.119
Spanish	298	2888	95	0.039	0.319

Table 5.2: *Term extraction evaluation, Latin terms included.*

domain corpus for each language. We also specify patterns of parts of speech for each language and only allow terms that comply with these patterns. These part-of-speech patterns are slightly different for each language, and are overgenerating rather than undergenerating in their basic aim to capture noun phrases.

We carried out an evaluation of recall and precision for the two approaches and for the four different languages in the FMA ontology (see Sect. 3.3.2). We calculate precision as how many of the terms suggested by the term extraction system are actually terms in the FMA ontology. Recall, accordingly, is calculated as how many of the terms in the FMA ontology, that also occur at least once in the domain corpus, are identified as terms by the system. Results are presented in Table 5.1.

Considering the domain of the corpus, we expect a large number of Latin terms to be identified by the term extraction tools. In a sense, identifying these as terms in the language in question is not necessarily incorrect, since Latin functions as a lingua franca in the anatomy domain (see Sect. 5.3). So, what if we include Latin terms in our evaluation? The results can be seen in Table 5.2.

The biggest difference between the two tables is that the recall drops enormously for German in the second table. A lot of Latin terms occur in the German part of the corpus and the term extraction system is not able to cap-

ture these well. Also, only 88 German FMA-terms occur in the corpus, most of which are rather frequent, which makes it easy to get a high recall on these terms.

It is possible, at least in our mutual information-based system, to balance recall and precision against each other by varying the threshold for the mutual information value. We refrain from doing that here, since the purpose of this experiment is not to create a state-of-the-art term extraction system; much more we want to perform a feasibility check, to see whether the results are good enough to form the basis for our further experiments. Unfortunately, even the more advanced TermExtractor system only reaches 0.124 precision and 0.084 recall. These numbers are too low for us to be able to use the system output as input to our ontology learning system. The errors from this preliminary step would prevent us from performing relevant evaluations of later steps in the ontology learning chain. We therefore make the following assumption for all subsequent experiments: we assume that we have access to the output from an ideal term extraction system, which outputs all the terms that are in the gold standard ontology and occur in the corpora – and nothing else. This assumption allows us to focus on evaluating other parts of the ontology learning process, without contamination from error propagation.

5.2 Term equivalence in parallel corpora

Given that we now have a set of terms in each language (as discussed in the previous section), the next step towards learning a cross-language terminological ontology is to find out which terms in the different languages mean the same thing.[1]

Various kinds of distributional similarity measures have been tried on the task of extracting cross-language term equivalents in past research. Another approach aims at solving the task by using methods from statistical machine translation, though the focus there has often been word alignment rather than extraction of equivalents – a distinction between looking at word *types* and word *tokens*, as discussed in Sect. 2.6.

We perform a systematic comparison of these two main approaches on a variety of language pairs, using the JRC-ACQUIS multilingual parallel corpus (see Sect. 3.2.1) to train the models, and Eurovoc V4.2 (see Sect. 3.3.1) to evaluate the results. The methods we investigate rely on sentence or document aligned texts, which is why we cannot use the Wikipedia anatomy corpus (studied in the previous section) for these experiments.

[1] Part of the research described in Sect. 5.2 has been published as Hjelm (2007).

5.2.1 Experimental setup and results

We use a group of distributional similarity models in this evaluation. We start off by comparing them to each other; we look at how they differ and at which model is the most effective for solving the translation task. The following are the three main characteristics where our distributional models differ:

1. Whether the co-occurrence matrix is built using first or second order co-occurrence data.
2. Whether random indexing or no dimensionality reduction is used.
3. Whether cosine or mutual information is used as the similarity measure.

We describe each alternative in the following.

The experiment involves all pairwise combinations of the following languages: German, English, French and Swedish. This means that six language pairs have been evaluated and thus twelve directions of translation.

Translating terms

When learning a cross-language ontology from text, we are interested in finding equivalence relations between *terms* in the source and target languages – relations between terms and non-terms ("regular words") or relations purely between non-terms are only of secondary interest (see Sect. 2.6 for a discussion on term equivalence).

In our experiments, we assume that the term extraction has already been carried out correctly. This means two things:

1. The task for the systems consists in translating the Eurovoc terms.
2. The translation candidates are limited to the target language terms – no non-terms are allowed as translation candidates.

These restrictions may seem rigid. However, if we assume that the term extraction process has been carried out correctly and we also assume that a *term* in the source language is always translated with a *term* in the target language, the restrictions are needed for the sake of consistency.

Data and gold standard

We use the JRC-ACQUIS corpus for building the distributional models and for training the GIZA++ system. GIZA++ is the word alignment part of a statistical machine translation system, introduced in Sect. 2.6.2.

The descriptors in Eurovoc constitute our gold standard; when the system translates the descriptor for a concept in the source language with the descriptor for the same concept in the target language, the translation is counted as correct, otherwise as incorrect.

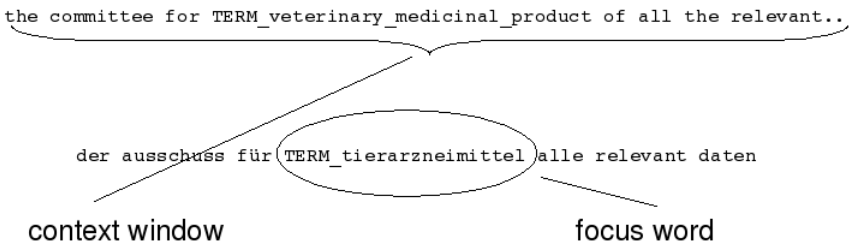


Figure 5.1: Constructing a distributional model for translating from German to English (the texts have been pre-processed). The entire English alignment unit is used as a context window when building the second order co-occurrence model for German.

Comparing the distributional models

We start off by comparing the different distributional similarity models, before continuing to include statistical machine translation in the comparison.

First vs. second order co-occurrence models: When using first order co-occurrences, rows in the co-occurrence matrix represent terms and columns represent documents, or in this case paragraphs. One model per language and language pair is needed, since the paragraph alignment is unique to each language pair.

When using second order co-occurrences, one makes use of a fixed-size sliding window to determine which words are to be considered neighbors of the focus word. Building a model for the target language, we proceed in the standard fashion, just as if we were building a single language model. For the source language, we instead use the target language part of the alignment unit as the window, as illustrated in Fig. 5.1. However, nothing actually forces us to use the *target* language words as features, we might as well use the *source* language words as features, or we could use both. We will return to this point further down. We make no adjustment for the proximity of the words, since we do not wish to make any assumptions about the similarity of word order between the languages involved.

Random indexing vs. full matrix: As mentioned previously, we want to compare the effects of not using dimensionality reduction with that of using random indexing. Of course, using a reduced matrix can give computational benefits, especially when working with larger text collections. Here, we are mainly interested in the effects a reduced matrix might have on the *accuracy* of the system.

Cosine vs. mutual information: It would be methodologically pleasing to try these two different similarity measures for all combinations of first and second order co-occurrence models, paired with the dimensionality reduction option discussed previously. However, applying the mutual information measure

does not make sense after the dimensionality reduction has been performed, since most or all vectors will be dense by then, containing few or no zeros. We use the following formula to calculate mutual information:

$$\sum_{x,y} p(x,y) log \frac{p(x,y)}{p(x)p(y)}$$

This typically presupposes a binary representation, meaning that if the number of zero-entries in all vectors is very low or zero, the measure will judge most or all vectors to be equally similar to each other. We therefore refrain from evaluating the mutual information measure on models where dimensionality reduction has been performed. To calculate the cosine measure, we normalize the vectors to unit length, and then take the dot product of the two vectors.

Given the great number of similarity measures available, it would have been possible to include many more in the evaluation. The cosine measure was chosen because of its widespread application in information retrieval, and the mutual information measure because of its broad use in the information theory community, along with its giving the best results in the comparison in Ribeiro et al. (2000), where the task is similar to ours.

Results for the comparison of the distributional models: For each combination of settings, we evaluate each of the twelve translation directions. Again, as mentioned previously, we only consider the descriptors of the target language as translation candidates. As input to the system, we use all source language descriptors that occur at least once in the source language text of the parallel corpus at hand. We also split the descriptors into eleven frequency classes (counted separately for each of the twelve directions of translation): 1, 2–5, 6–10, 11–50, 51–100, 101–500, 501–1000, 1001–5000, 5001–10000, 10001–50000 and $\geq$50001. We calculate the average accuracy for all twelve directions of translation, for each frequency class as well as the overall accuracy, regardless of frequency (displayed later in Table 5.3). Fig. 5.2 shows a comparison of all applicable combinations of settings when working with first order co-occurrence models and Fig. 5.3 shows the same comparison for second order co-occurrence models. Both figures show that we get the best results when using the cosine measure and the non-reduced matrices.

As mentioned, there is no inherent reason to choose the target language words as features when building a second order co-occurrence model. In fact, since four languages are involved in these experiments, we made an experiment where words from all four languages are used as features. As can be seen in Table 5.3 (where this method is labeled "2ndOrder-Full-Cosine-CL"), this brought a very moderate increase in performance, but still this is the most effective second order model.

Throughout all experiments, we use the log_2 of the frequencies in the models rather than the raw frequencies. The intuition behind this is that a feature (word or document) co-occurring twice with the focus word should be

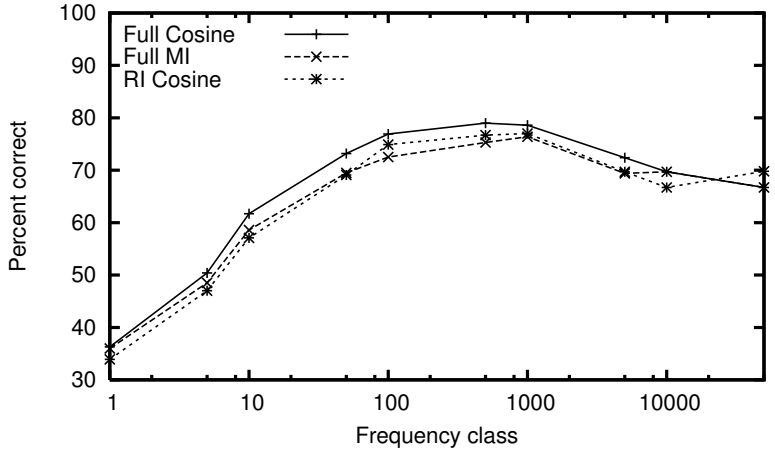


Figure 5.2: First order co-occurrence models. “Full” stands for no dimensionality reduction, “MI” for mutual information and “RI” for random indexing.

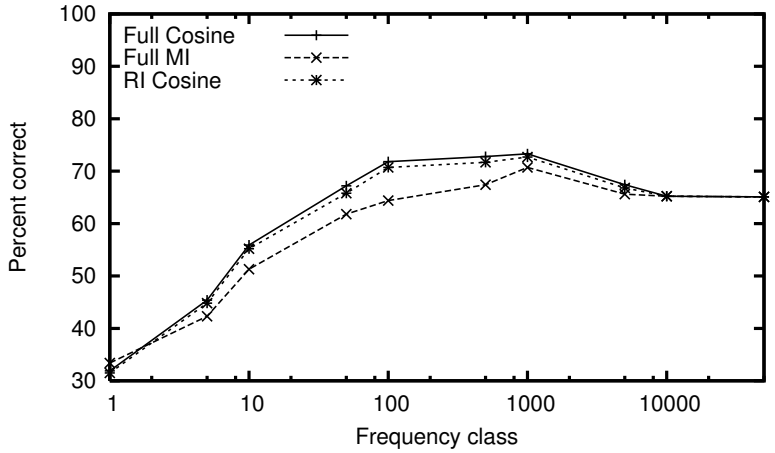


Figure 5.3: Second order co-occurrence models.

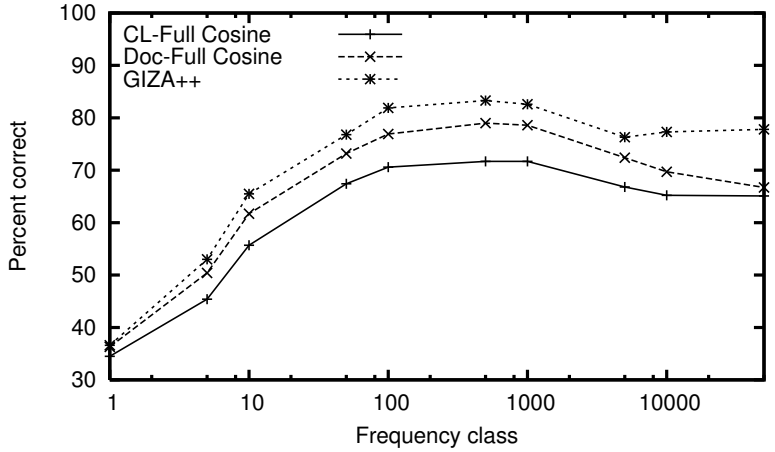


Figure 5.4: Top performing models compared: first order (labeled Doc-Full Cosine), second order (labeled CL-Full Cosine) and statistical (labeled GIZA++).

weighted higher than a word that co-occurs only once – but probably not *twice* as high. In information retrieval, using log frequencies, or the *logarithmic term frequency*, is a standard technique. It has also been applied successfully to the closely related problems of automatic thesaurus discovery (Grefenstette, 1994), latent semantic indexing (Landauer and Dumais, 1997) and text clustering (Hotho et al., 2001).

Statistical machine translation vs. the distributional models

We use the GIZA++ system with the standard settings provided in the publicly available distribution.[2] All terms are treated as single words by the system after the term spotting has been applied during the pre-processing. Because of this, we can ignore the fact that GIZA++ lacks the possibility to capture many-to-many relations. Figure 5.4 displays a comparison between the best performing first and second order models with the results from GIZA++. "CL" in the figure stands for *cross-language*, and refers to the fact that words from all four languages involved were used as features when training that model.

Ensemble method

We combine the results of the top performing models, shown in Fig. 5.4, in an ensemble method. The idea is that, even though the statistical model outperforms the other two, they may still contain useful information that the statistical model is missing. There are at least two factors one would like to consider when combining the results of the different systems: how confident each system is of its decision (modeled in the S' function below) and how accurate the system has been in the past (modeled in the S'' function below). For each source language term, we

[2] We used the version of GIZA++ released in 2003.

look at the top ten translation candidates for each of the three models. The scores for each model are rescaled, so that the scores for the top ten translation candidates for a particular source term sum to one, or, equivalently:

$$S'(x,y) = \frac{S(x,y)}{\sum_{y_i} S(x,y_i)}$$

where x is the source term, y a translation candidate, S the scoring function and S' the rescaled scoring function. We then weight the scores from each model according to how accurately it performs on one direction of translation for one language pair,[3] which we set aside for testing during this particular experiment. The scoring function which is finally used to re-rank the top ten suggestions from the three models looks like this:

$$S''(x,y) = \alpha \cdot S'_a(x,y) + \beta \cdot S'_b(x,y) + \gamma \cdot S'_c(x,y)$$

where α, β and γ are the accuracies of the respective models, normalized so that $\alpha + \beta + \gamma = 1$.[4] Basically, this amounts to the *average combination rule*, which is a standard way of combining multiple classifiers (Tax et al., 2000). The results, displayed in Fig. 5.5, show a slight improvement when compared to using the statistical model alone. Finally, Table 5.3 shows the percent correct for each method, regardless of frequency class.

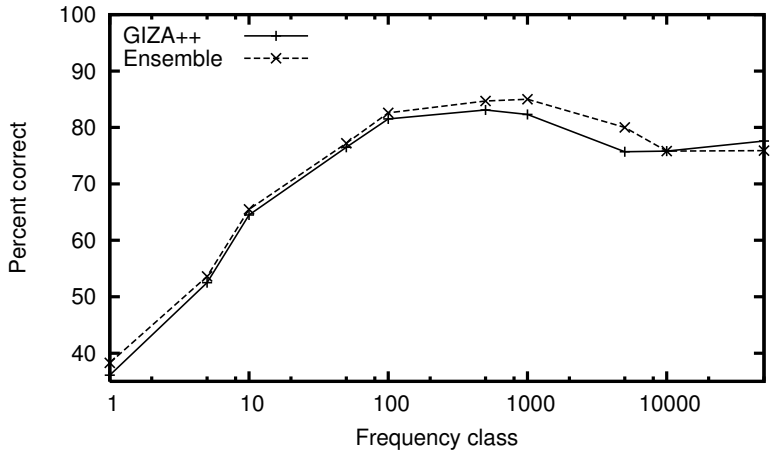


Figure 5.5: Comparing GIZA++ to the ensemble method.

[3] We used German to French, to have one Germanic and one Romance language.

[4] This resulted in the following parameters, for the first order, second order and statistical models, respectively: $\alpha = 0.334$ $\beta = 0.313$ $\gamma = 0.353$.

	Percent correct
1stOrder-Full-Cosine	**61.4**
1stOrder-Full-MI	58.7
1stOrder-RI-Cosine	58.1
2ndOrder-Full-Cosine	56.0
2ndOrder-Full-MI	52.4
2ndOrder-RI-Cosine	55.1
2ndOrder-Full-Cosine-CL	**56.2**
GIZA++	64.4 (64.0)
Ensemble	**65.8 (65.3)**

Table 5.3: *Percent correct over all frequency classes, totally 37,316 translations evaluated for each model. Numbers in parenthesis show results when German-French is not included (this direction of translation was used for parameter tuning in the ensemble method).*

5.2.2 Increasing translation precision

As we gather from Table 5.3, the very best method for finding term equivalents produces exactly correct results in 65.3% of the cases (for the given corpus and term set). In Sect. 2.2 we were discussing the possibility of using term equivalents as "bridges" between data from different languages, enabling us to fully exploit the added information of the cross-language data. If these bridges are faulty in too many cases, we run the risk of having the cross-language data confusing our ontology learning system rather than helping it. We would therefore be willing to trade recall for precision in this case; if we can have a subset of term equivalents where we can be relatively sure that the translation is correct, we will simply ignore the (possibly incorrect) information provided by the other translations.

Since we see from Fig. 5.4 that there is a strong connection between frequency and accuracy of translation, our first intuition may be to only use translations where the terms lie in a certain frequency range. We could, for example, decide not to use translations where the terms occur ten times or less in the corpus – this would give us an average accuracy of 80.4%, but we then only give translations in 48.6% of the cases. To try to increase accuracy even further, we can disregard all terms occurring 50 times or less. This would bring us to an accuracy of 83.8% and we would give translations in only 22.6% of the cases.

We instead decided to test an approach where we use the fact that we are working with more than two languages in our experiments. When deciding which translations to include in the final result set, we go through the translations in all language directions and look for groups of four terms (one from

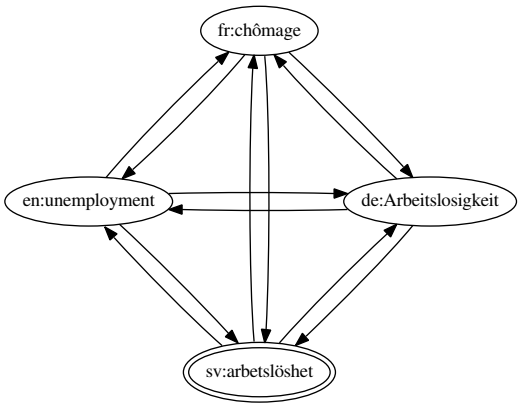


Figure 5.6: Example where all four languages form a complete directed graph (every node is directly connected to every other node), signifying that the translation probably is correct.

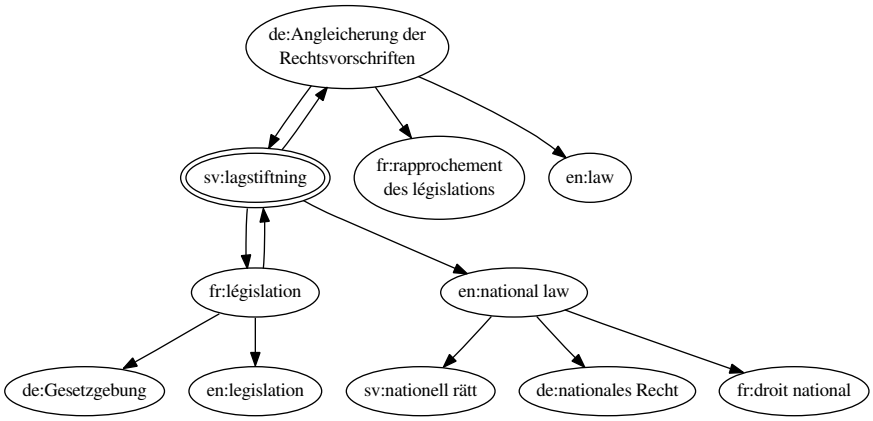


Figure 5.7: Example where the graph is not complete, indicating a probable error in the translation.

each language) where all terms are translations of each other. This is the same as saying that the four terms form a complete directed graph; an example of this is given in Fig. 5.6. We also show an example (Fig. 5.7) of translations that do not form a complete graph and that are therefore discarded. Using this approach, we suggest a translation in 36.5% of the cases (somewhere between the two frequency groups examined previously) but we reach an accuracy of 98.4%, which is a considerable boost when compared with the frequency threshold strategy. We are now at a level of accuracy where the results can be used as the previously discussed bridges across the data in different languages, without the risk of confusing our ontology learning system with incorrect information. Results of experiments using this data are presented in Sect. 6.4.

By allowing an increasing number of missing links in the structure in Fig. 5.6, one could increase recall at the cost of precision. Another way of achieving the same effect would be to only require translations from three or even two languages to form complete graphs during the filtering process. We return to discuss the effects on an ontology learning system of lowering the precision threshold in Chap. 7.

The idea of using cross-language data from more than two languages in the translation process is reminiscent of Borin (2000), where alignments from a third language are incorporated in the alignment data used for training a statistical machine translation system. The major difference is that where Borin's approach mainly contributes to increasing the *recall* during the alignment process, our approach targets *precision*.

5.2.3 Statistical machine translation, distributional similarity or ensembles?

Using the non-reduced matrix gives the highest correctness figures, both for first and second order co-occurrence models, though the reduced version is following closely for the second order model, as seen in Fig. 5.3. There are possible computational benefits of using a reduced representation. However, since both data structures and algorithms designed for working with sparse matrices and vectors exist, one would have to investigate just where the breaking point lies. For the current experiments, using the non-reduced, sparse matrix proved more efficient both in terms of time and in terms of memory usage, since the reduced matrices have to work with dense representations.

It should be noted that when using random indexing, the results will vary with the dimensionality of the matrix and the number of non-zero elements used in the random vectors. We used a dimensionality of 1800 and an average of eight non-zero elements (positive and negative), which lies in the range of what is suggested in Sahlgren (2006). We note a larger gap in accuracy between the reduced and the full matrix for the first order models than for the second order models (0.9% absolute vs. 3.3% absolute). From this we can hypothesize that the reduced first order model would have performed better using a higher dimensionality, considering that the non-reduced first order models have a higher dimensionality than the non-reduced second order models. This is left for future experiments to confirm.

The first order models consistently outperform the second order models in these experiments. Further, the cosine measure outperforms the mutual information measure in the cases where a direct comparison can be made. This is contrary to what Ribeiro et al. (2000) reported, but the experiments described here have been conducted on a much larger corpus with a larger variability of languages – perhaps this explains the differences in the results.

The statistical approach clearly outperforms both the first and the second order models. This is again contrary to what Sahlgren and Karlgren (2005)

report. However, they claim an accuracy of "something less then 1/3" for the GIZA++ system, which lies far below the 64.4% measured here. The two evaluations cannot be directly compared, due to several differences in the methodology of the experiments. The most important difference, which probably by itself explains the vast discrepancy when measuring the performance of GIZA++, is that our study uses texts aligned on a *paragraph* level, whereas Sahlgren and Karlgren used texts aligned on a *document* level. Sahlgren and Karlgren are also studying *words*, not *terms*, which makes their task harder, since they have to pick the correct word out of 40,000 to 70,000 translation candidates, whereas our study typically only has about 3,500 terms as translation candidates. On the other hand, the evaluation applied here is stricter, since only the descriptor in the target language is counted as correct, where Sahlgren and Karlgren also count partial matches in the target language part of a bilingual dictionary as correct.

The correctness for terms occurring only once seems low, at slightly below 40%. Consider, though, that there is no guarantee that the corresponding target language descriptor co-occurs *even once* with these terms. Such cases can arise from, e.g., faulty paragraph alignment or from the (human) translator choosing to use a different term than the descriptor in the target language translation. Accuracy also drops for the very frequent terms, which at first may seem surprising. Melamed (2000) notes, though, that frequent words tend to be translated with less consistency than less frequent words. We believe this decrease in consistency is what lies behind the drop in accuracy (also observed in Sahlgren, 2004) – remember that the evaluation only accepts *one* correct translation for each term: its target language descriptor.

Using the ensemble method, the results are boosted by 1.3% absolute. Though the increase is relatively small, the difference is statistically significant beyond the 0.001 level according to McNemar's test (McNemar, 1947). If we use a more lenient evaluation method, counting each result as correct if the corresponding descriptor occurs among the top three translation candidates, GIZA++ achieves 66.9% correct translations on average and the ensemble method reaches 68.6%. Extending this to the top ten candidates, we get 67.2% for GIZA++ and 70.3% for the ensemble method – a difference of 3.1% absolute. The rather small increases in correctness for GIZA++ using the lenient evaluation methods can most likely be explained by the internal thresholds in the system. Due to these thresholds, GIZA++ most often returns *less* than ten translation candidates for any given source term, which means that the system will not profit as much from using these lenient evaluation schemes.

In an application scenario where the target language translation will be used for cross-language information retrieval, it is not necessary for the target language term to be a precise translation of the source language term. The results are also of value if the suggested translation belongs to the same semantic field as the source language term. E.g., one of the suggested translations from

our experiments of German *Wohnungskauf* (buying of an apartment) is 'mortgage', which is likely, if posed as a query, to result in a set of relevant documents for someone interested in buying an apartment. This is another area which would be interesting to evaluate formally (such cases were obviously counted as errors in the evaluations presented here).

5.3 Term equivalence in comparable corpora

The main difference between the experiments presented in this section and the study in Sect. 5.2 is that the corpus used here, the Wikipedia anatomy corpus (see Sect. 3.2.2), is *comparable* (deals with the same *topic* in different languages) rather than *parallel* (the same *content* in different languages).

We follow the basic procedures developed by Fung and McKeown (1997) and Rapp (1999), which we outlined in Sect. 2.6.4. To recapitulate, we build second order co-occurrence models for the languages that we wish to translate between. We then translate the features (words) in the non-English model into English using general bilingual dictionaries that we download from Wiktionary (described in Sect. 3.4). This gives us a cross-language mapping between the features of the distributional models and we can proceed to calculate distributional similarity across languages in much the same way as we do for parallel corpora in Sect. 5.2.1.

Since we do not have a parallel corpus in this case, we do not have access to correspondences on the paragraph or document level[5] and are thus forced to resort to using second order co-occurrences, even though we showed in Sect. 5.2.1 that first order co-occurrences are more effective for solving the translation problem.

We introduce a new heuristic in these experiments. In addition to the mappings provided by the dictionary, we also make the assumption that if we find the exact same string of characters used as a feature in the two distributional models, these two strings (words) mean the same thing. Although this is far from certain, we are guessing that it will be correct in the vast majority of cases, due to a lot of Latin terms being used as a lingua franca and also to some proper names which keep their form after translation. Note that we are *not* using this heuristic as a way of directly translating our terms, we are only using it to provide us with additional mappings between our feature sets, for cases that are not covered by the bilingual dictionaries.

At first sight, working with an anatomy corpus might seem to be a special case, where our heuristic would work exceptionally well, since Latin is so closely associated with the anatomy domain in all languages. But almost any specialized domain makes use of a lingua franca to a greater or lesser degree, e.g., psychology (German), computer science (English) and philosophy

[5] For certain documents, we know their corresponding documents in the other languages, for others not. See discussion in Sect. 3.2.2.

(a mixture of Greek and German, among others). We therefore feel that this is not a "hack" that happens to work for our particular corpus, but something which is also applicable within other domains.

5.3.1 Experimental setup and results

When building the distributional models for each language, there are a number of parameters that can be varied and that have an impact on the quality of the results of the translations. We explain these parameters in Sect. 2.4.1 and we will be referring to them at various points throughout the rest of this chapter.

To evaluate the system, we use the terms that have a translation in the FMA ontology as a gold standard. However, to get more reliable statistics for the method, we limit the evaluation to English terms occurring more than 50 times in the corpus. Further, if the translation stipulated by the gold standard does not occur with some frequency in the target language part of the corpus, we cannot really expect the system to find this translation. We therefore introduce a further restriction on our test set: the target language translation must appear 15 times or more in the corpus.[6]

We evaluate three directions of translation: English-German, English-French and English-Spanish (English was chosen as source language, since we envision the scenario of extending the FMA ontology with translations of English terms). The restrictions on source and target word frequency unfortunately leave us with a very small test set. The English-German gold standard consists of 63 word pairs, the English-French of 105 and the English-Spanish of 71 word pairs. This means that we cannot take the accuracy figures presented in Table 5.4 at face value. They should be taken as indications on what to expect from a system such as the one described here and as pointing towards the increased difficulty of the problem as compared with working with a *parallel* corpus.

The results in the following evaluation are measured in *percent correct*. We apply the same criteria for correctness here as for the experiment in Sect. 5.2: a translation is considered correct if it is identical to the one given in the FMA ontology, otherwise it is considered incorrect. We measure correctness at three levels:

1. The highest ranking translation candidate is the correct translation.
2. The correct translation appears among the top *three* translation candidates.
3. The correct translation appears among the top *ten* translation candidates.

[6] We use a lower frequency limit for the non-English words, because the non-English document collections are smaller than the English.

Language pair	Top 1	Top 3	Top 10
English-German	33.3	44.4	65.1
English-French	19.0	33.3	54.3
English-Spanish	26.8	40.8	67.6

Table 5.4: *Evaluation of term translations using comparable corpora; results given in percent correct translations.*

The results are displayed in Table 5.4. If we compare these results with the ones in Table 5.3, we see a drastic drop in accuracy. This is to be expected, given that we are dealing with a harder problem.[7]

5.3.2 Parameter optimization and conclusions

We want to point out that the figures given in Table 5.4, low as they may seem, still in a sense are overly optimistic. Since the size of the gold standard is so small for this experiment, we could not split it in separate test and training partitions. The parameter optimization therefore has not been performed on a separate training set, but on the test set itself. The purpose of presenting our results here is thus not to try to convince anyone of the superiority of our system, but rather, again, to indicate the challenging nature of the task.

We give a brief overview of the parameter settings that proved the most effective when building the distributional models. There is little or no fluctuation in this matter between the different language pairs. We introduced the different parameters in Sect. 2.4.1.

- The larger the window size, the better the results. We experimented with window sizes ranging between 5–500 words. Of course, when the window size reaches 500 words (in each direction), we are using the entire document in many cases. This setting turned out to be the most effective. In a way, this can be seen as maximizing the amount of information taken into consideration, something which makes sense if data or other resources are scarce, which is the case here. For one, the corpus size for all involved languages is on the lower end of the spectrum (see Sect. 3.2.2). The same is also true for the size of the bilingual dictionaries (see Sect. 3.4). Going from a window size of 50 to one of 500 gave an increase in accuracy of 4% absolute, averaged over the three languages (we will only consider the accuracy of the highest ranking translation in this discussion). Using

[7] The two experiments were performed on different datasets, which of course inserts an uncertainty factor as to what is an effect of using the different datasets and what is an effect of using the different methods.

a window size below 50 decreases the accuracy further by another 1–2% absolute.

- The inverse distance weight outperforms the other two distance weighting schemes, but the difference to the logarithmic weighting scheme is very small, on the order of 1% absolute. This confirms our intuition that words close to the focus word should be given more weight, but it is not possible to say which of the logarithmic and the inverse distance scheme is more apt, due to the small difference in accuracy and the small test set.
- Some initial experiments using no feature weighting were performed, with bad results. Using mutual information for the feature weighting is in turn more effective than using the conditional probability. The difference in accuracy between the two feature weighting schemes is on the order of 8% absolute, averaged across the languages, making this the parameter with the largest impact on the system. The difference between the two can be explained by the fact that mutual information also takes negative evidence into account, whereas this is not the case for the simpler conditional probability measure.
- Distinguishing between the left and right contexts consistently gives a better result, although only on the order of 2% absolute on average. This is a cheap way of adding low-level word order information to the model, and the fact that making this distinction improves the results indicates that word order is a factor to be considered.
- Using the lingua franca-heuristic gives a moderate increase in accuracy, on the order of 3% absolute. Our assumption that these terms actually tend to mean the same thing thus pans out – at least for the current languages and domain.

Just as in the results of the term extraction experiments in Table 5.1, where we had far lower recall for French than for English and German, the results in Table 5.4 also show the lowest accuracy for translating into French. We believe the reason for this is simply that the German and Spanish words included in the gold standard are more common, with higher frequencies in the corpus, which makes them easier for the system to translate.

In Sect. 6.3, we present methods for using output from machine translated terms to merge information across languages when training an ontology learning system. The results from Table 5.4 show that the error rate is too high for the system output to be used for that purpose, using comparable corpora and the method presented here. There is thus plenty of room left for new ideas and methods to improve on the results.

6. Experiments in Ontology Learning

This chapter looks at the ontology learning process, going from a domain-specific parallel or comparable corpus and language-specific terms to a hierarchical is-a ordering of cross-language term sets. We first look at learning a prototype-based ontology and then move on to learning a terminological ontology, both with a special focus on cross-language aspects. The experiments in this chapter are carried out on the JRC-ACQUIS parallel corpus and the accompanying Eurovoc thesaurus, except for Sect. 6.1, where we use the Wikipedia anatomy corpus and the accompanying FMA ontology.

6.1 Learning a Prototype-based Ontology from Cross-language Data

As our baseline, we follow the work of several previous researchers (see Sect. 2.5.1) by extracting a prototype-based ontology, using English language texts, and compare the result to a gold standard ontology.[1] We then repeat the procedure, this time building a prototype-based ontology from resources in four different languages (adding German, French and Spanish to the English), using a comparable corpus. We show that the cross-language version gives an improvement in accuracy and stability over the single language version, when compared to the gold standard.

We also use a hierarchical k-means clustering technique and show that we are able to reproduce the original ontology with greater fidelity than when using a bottom-up agglomerative clustering approach. We use the FMA ontology and the Wikipedia-anatomy corpus for this series of experiments (see Chap. 3).

6.1.1 Hierarchical term clustering

We examine two kinds of hierarchical clustering: bottom-up agglomerative clustering and hierarchical k-means. Neither method produces a hierarchy in the traditional sense, but rather a structure like the one depicted in Fig. 6.1, which we refer to as a prototype-based ontology, when the objects we are

[1] Part of the research described in Sect. 6.1 has been published as Hjelm and Buitelaar (2008). The work was done by the author with support and guidance provided by the co-author, Paul Buitelaar.

clustering are terms. The bottom-up approach builds this structure starting with each term in its own cluster, whereas hierarchical k-means starts with all terms in the same cluster and recursively splits each (sub)cluster.

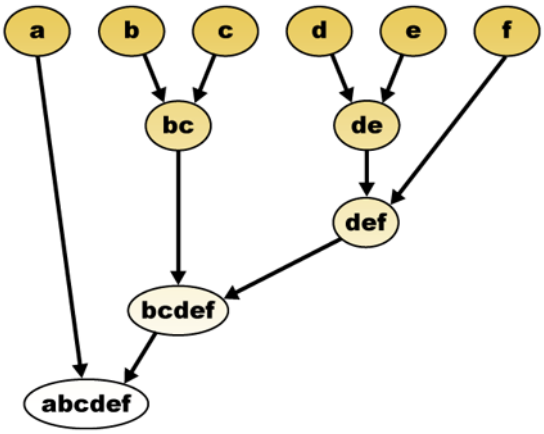


Figure 6.1: Structure produced by hierarchical clustering methods. Picture taken from Wikimedia Commons (http://commons.wikimedia.org), file name "Hierarchical_clustering_diagram.png".

Bottom-up agglomerative clustering

We start by building a distributional similarity model, using the settings that gave the best results in the study presented in Sect. 4.2. We use the following parameter settings (the parameters were introduced in Sect. 2.4.1):

- Window size: 500 (in each direction)
- Distance weighting: flat
- Feature weighting: none
- Left/right distinction: not made
- Minimum feature frequency: 50
- Dimensionality reduction: SVD, 200 dimensions

The distributional similarity model itself is not at the center of this experiment, therefore we will not analyze the settings in any greater detail here. For clustering, we employ a version of average linking, where we start by calculating a centroid representation for each cluster, and we measure the similarity between two clusters by calculating the *cosine* of their normalized centroid vectors.

Hierarchical k-means clustering

The agglomerative clustering approach, described above, produces a binary tree. Because we wish to cluster a relatively large number of terms (1,164 in

total – all the terms with a minimum frequency of 50 in the English Wikipedia corpus), the result is a deep tree, especially if contrasted with the much flatter FMA ontology. Further, there are some hierarchical relations a binary tree will never be able to capture correctly. E.g., the relations between 'finger' and 'thumb', 'index finger', 'middle finger', 'ring finger' and 'little finger' are not binary (one-to-one) but n-ary (one-to-many). We would prefer to model the relationship with 'finger' directly dominating all the others. Using hierarchical k-means clustering, we are no longer forced to produce binary trees; we can simply tell the system how many times we would like to split the cluster at each iteration. Though we still do not get a structure where one term directly dominates other terms, but rather a one-to-many variant of the structure shown in Fig. 6.1, we at least have a chance of producing a model which is *closer* in structure to the FMA ontology.

For each clustering step, we try to find the appropriate k for splitting that particular cluster. We iterate through different values of k and evaluate each clustering by calculating the harmonic mean of cohesion and separation between the clusters (discussed in Sect. 4.2) and choose the best performing k at each step. In our experiments, we set an upper limit for k to 20, since it would be very time consuming to evaluate *every* possible k value.[2]

6.1.2 Clustering from cross-language evidence

To test the effects of including evidence from more than one language when performing the clustering, we start by building four separate distributional models, one for each language. Next, for each term in every non-English model, we look up if it is listed as a translation in the FMA ontology of any of the English terms. If it is, we concatenate the vector for this non-English term to the vector of the English term, resulting in a vector that is twice the length of the original vector. This process is repeated for every non-English language, which means that the final vectors we are working with are four times the length of the original vectors (since we are using four languages, all with the same vector lengths). Fig. 6.2 illustrates the idea behind such a cross-language vector.

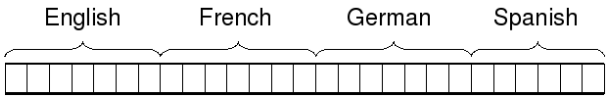


Figure 6.2: Distributional information from each language is concatenated to form an elongated version of the co-occurrence vector. The vectors used in the single-language experiments consist only of the part marked "English".

[2]Choosing 20 as upper limit as opposed to any other, higher, number was a practically rather than theoretically motivated choice.

6.1.3 Evaluating the hierarchical clustering

We apply the PMCC-measure, described in Sect. 4.3.1, for calculating a similarity measure between the learned ontologies and the gold standard. We are unable to use the two other evaluation measures discussed in the same chapter, since we are learning a prototype-based ontology (see the discussion for Experiment 7 in Sect. 4.3.2).

We start by comparing the bottom-up agglomerative clustering to the hierarchical k-means clustering described in Sect. 6.1.1. Table 6.1 shows that the k-means approach gives a result which is substantially closer to the gold standard than the bottom-up agglomerative approach does. Since the k-means clustering uses a random initialization, we repeat the experiments ten times and report the average correlation and the standard deviation for this approach. These experiments were carried out using only the English terms and texts.

	$\bar{\rho}$	σ	ρ_{min}	ρ_{max}
Bottom-up agglomerative	0.109	N/A	0.109	0.109
Hierarchical k-means	0.166	0.043	0.114	0.237

Table 6.1: *Comparing bottom-up and k-means single language clustering. ρ is the correlation, σ the standard deviation.*

We see two possible explanations for the improvement we measure when using the k-means approach. One is that, since we are evaluating different k for each new split and sticking with the best one, it is possible that we then are able to cluster the terms in a way that is more conformant to the data. Another explanation could be that we are mimicking the flatter structure of the FMA ontology better this way, than we are with the bottom-up approach. As ever, a combination of these two factors seems most likely.

Turning our attention now to the comparison of the models built from strictly English and the cross-language data, Table 6.2 shows that the cross-language data on average gives a considerable increase in correlation. This increase is paired with a marked decrease in standard deviation, which indicates that the cross-language method is less sensitive to different random initializations.

	$\bar{\rho}$	σ	ρ_{min}	ρ_{max}
Strict English	0.166	0.043	0.114	0.237
Cross-language	0.201	0.027	0.137	0.229

Table 6.2: *Comparing strict English and cross-language k-means clustering.*

Now, one might argue that these improvements are not surprising – more data is always more data. However, since the additional data comes from other languages than the original data, it is not self evident that the added data would help to clarify the ontology learning process, rather than confuse the models by introducing noise. Our results, however, do support using the cross-language evidence for this application.

We want to point out that because not all English terms are translated, we get an effect where the vectors for the translated terms will be more similar to each other, simply because they are translated (the same reasoning applies to the vectors for the non-translated terms). We believe this has a detrimental effect on the cross-language clustering results, because the system will mix translated and non-translated terms in the same cluster less frequently based on this largely arbitrary condition. It is however possible that we get a positive effect in some cases, where clustering translated and non-translated terms separately happens to conform to the gold standard. Our intuition says that the system would benefit from removing this difference between translated and non-translated terms and that the positive effects we measure mainly are due to the added cross-language information. We already saw a slight benefit from using cross-language data when extracting term equivalents (results in Table 5.3) and we will see the effects more clearly again in our experiments in Sects. 6.4 and 6.5.

The approach for building the cross-language model presented here assumes that we have access to a domain-specific bilingual (or multilingual) dictionary. One could imagine getting by without such a dictionary and instead use statistical word alignment techniques to identify term equivalents (see Chap. 5). Because we are dealing with comparable rather than parallel texts here, we would have to resort to methods designed for such texts, like the ones described in Sect. 5.3. Methods for comparable texts have the disadvantage of being much less accurate than techniques developed for parallel texts. To avoid evaluating the quality of a term translation system rather than the effects of cross-language evidence, we decided to use the translation information coded in the FMA ontology as our lexicon. This is not too far fetched a scenario: having access to a domain-specific bilingual dictionary and wishing to learn an ontology for the terms listed there. We explore the possibility of using automatically word-aligned data when working with a parallel corpus in Sect. 6.3.

Summing up, the increase in average correlation and decrease in standard deviation, when evaluating against a gold standard, supports the use of cross-language evidence for the task of learning a prototype-based ontology. What is more, the resulting resource has added value when compared with the single language approach, since we are now free to switch between languages at will, while staying within the same hierarchical structure.

6.2 Features for Recognizing Hyperonymy and Cohyponymy

When constructing a hierarchy of terms, there are two relations which are more prominent than others. The first is the hyperonymy relation, that connects two terms in the vertical plane. The second is the cohyponymy relation, that instead connects two terms in the horizontal plane. Also Snow et al. (2006) and Ryu and Choi (2006) use these two relations as their basic building blocks. We have shown in Sect. 6.1 how distributional similarity can be used to order terms into a (pseudo) hierarchy. Here we will look at all features which we think will be useful for recognizing either or both of our two basic relations, which in turn will enable us to learn a terminological ontology. We will use these features to train a support vector machine classifier in Sect. 6.4, so the selection of some of these features is made with this type of classifier in mind. The experiments in this section and the rest of this chapter are carried out on the JRC-ACQUIS parallel corpus and evaluated against the Eurovoc thesaurus.

Many of the features described in the following share the fact that they use the context of the term (or term pair) in focus. When discussing distributional similarity in Sect. 2.4, we mentioned two different ways of looking at context: using first order co-occurrences (document-based) and using second order co-occurrences (context window-based). Further, using second order co-occurrence, we have a choice between using narrow or wide windows, or something in between. Rather than commit ourselves to one particular context model, we use four different models: one using first order co-occurrence, one using a narrow context window (three words on each side of the focus word), one using a wide window (500 words to each side) and one in between (50 words to each side). We then let the classifiers decide which context model (or combination of models) is most effective for recognizing a particular type of relation between two terms.

Note that we do not necessarily expect any single feature *by itself* to be able to identify term pairs belonging to either of the two relations – rather we expect the collected evidence, when several features simultaneously point to the same result, to be the decisive factor. We use a total of 22 features in our experiments and we describe them in the remainder of this section.

6.2.1 The subsumption measure

This measure was described in Sect. 2.5.5; it is the main feature in the system developed by Sanderson and Croft (1999). A similar measure, formulated in a manner to make it more widely applicable, was also used in Cimiano (2006). There it was found to confuse the classifier, or, rather, to produce negative weights, which means that it would be indicative of *no* hyperonymy relation being present. Cimiano presents a suggestion for how to alter this measure,

effectively changing the roles of the subsuming and the subsumed term, but it is his original measure we employ in our experiments – with positive results.[3] We have no clear explanation for this discrepancy, but give a theoretical motivation for our approach in the following.

Figures 6.3 and 6.4 show different situations where the occurrences of a term t subsume the occurrences of a term u. Three main sets of interest are formed in the prototypical case: T (occurrences of t), U (occurrences of u) and $T \cap U$, shown in Fig. 6.4. First, for us to consider term t to subsume term u to any degree whatsoever, we introduce a condition that $|T \setminus U| > |U \setminus T|$. If this condition is not met, we would rather say that term u subsumes term t, or that the terms are neutral with regard to subsumption. To determine the degree of subsumption, when the condition is met, we calculate:

$$\frac{|T \cap U|}{|U|}$$

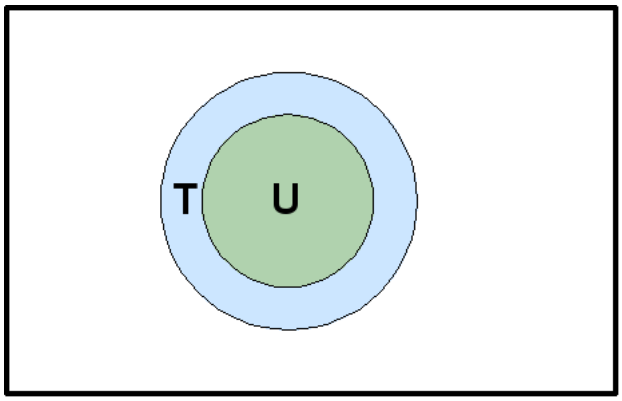


Figure 6.3: Full subsumption – the ideal case, where the occurrences of one term completely subsume those of another.

The subsumption measure is similar to the *overlap coefficient* (Manning and Schütze, 1999), which is written as follows:

$$\frac{|T \cap U|}{min(|T|, |U|)}$$

The modifications to the overlap coefficient introduced in the subsumption measure aim at capturing asymmetry in the relation; for the overlap coefficient, only the *degree* of overlap is of interest, not the *direction* (i.e., which term subsumes the other).

We compute four versions of the subsumption feature, one for each context model described at the beginning of Sect. 6.2. The features are meant to

[3] Cimiano distinguishes between first and second order co-occurrence and keeps the formulation closest to Sanderson and Croft's for first order co-occurrence, but uses the new formulation for second order co-occurrence.

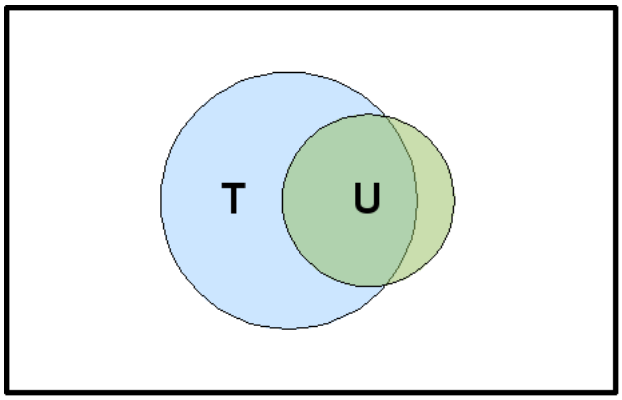


Figure 6.4: Partial subsumption – the prototypical case, where the occurrences of one term partly subsume those of another. Also, the condition that set $T \setminus U$ should be bigger than set $U \setminus T$ is met in this example.

be useful for recognizing the hyperonymy relation; the subsuming term is the hyperonym and the subsumed term is the hyponym. The rationale is that the hyperonym, being more general in meaning, is likely to appear in a number of contexts where the hyponym does not, in addition to a large number of contexts where both terms appear. That this holds for first order co-occurrences is uncontroversial; it has already been demonstrated by Sanderson and Croft (1999). Consider the following set of examples to see the reasoning behind using the measure also for second order co-occurrences:

- The dog barked.
- *?*The dog mooed.
- The animal barked.
- The animal mooed.
- *?*The cow barked.
- The cow mooed.

The question marks are used to indicate semantic markedness in the previous examples. The more generic hyperonym *animal* has greater flexibility to be used in different settings than have the more specific hyponyms *dog* and *cow*.

6.2.2 Distributional similarity

We discussed distributional similarity at some length is Sect. 2.4. We said there that it is a measure of relatedness *of some kind*, but that we are generally not able to distinguish between different relations solely based on this measure. We expect the distributional similarity to be high both for terms related by hyperonymy and for terms related by cohyponymy. Distributional simi-

larity by itself might not be enough for the classifiers to make the distinction between different relations, but together with the information from the other features we expect it to provide useful evidence.

We again have four context models for this measure, but we also use two different representations: one using the unprocessed co-occurrence matrices and another using matrices where singular value decomposition has been applied (we use a standard dimensionality of 200 in our experiments). This results in a total of eight features describing distributional similarity: four different context models for each of the two different representations.

6.2.3 Hearst-patterns

We introduced Hearst-patterns in Sect. 2.5.3, where we also described their use in a number of different systems. We use a total of twelve English patterns, which we gather from Hearst (1992), Iwanska et al. (2000) and Cimiano et al. (2005b) and we translate these into the other languages. Most previous studies have considered Hearst-patterns a binary feature and set it to *true* if a term pair has been observed in one of the patterns in the text. We calculate the number of times a particular term t occurs as the hyponym of another term u, but also consider the total number of times t occurs as the hyponym of any other term:

$$hearst(u,t) = \frac{pattern(u,t)}{pattern(\_,t)},$$

where $pattern(u,t)$ gives the number of times term u has occurred as a hyperonym of term t in any of the patterns. The idea, which is also used in Cimiano (2006), is to model that the more often a term pair occurs in one of the patterns, the more sure we can be that the hyperonymy relation holds, but also that if term t has occurred as a hyponym for many other terms than u, this again should make us less sure of our decision.

6.2.4 Head matching heuristic

This feature is described in Sect. 2.5.2, and the idea is simply to say that 'consumer credit' is a kind of 'credit' or that a 'direct investment' is an 'investment' – a simple way of recognizing the hyperonymy relation by using head matching rules. We also have a feature for the cohyponymy relation, which is set to *true* when we have two terms such as 'indirect tax' and 'direct tax', where both share the same lexical head, but neither *is* the lexical head of the other. Because we do not have a parser as a part of our system, our way of determining what is the lexical head of a term simply amounts to locating the rightmost word(s) of the term.[4] When we, e.g., compare the two terms 'power plant' and 'nuclear power plant', we set the hyperonym feature to "true", since 'power plant' matches the two rightmost words of 'nuclear power plant'. If a

[4] For French, we use the leftmost word(s) instead.

term has a preposition as one of its constituents (we determine this by looking in a list of prepositions for the respective language), the word(s) directly preceding the preposition is instead considered the lexical head. This enables us to give the feature the correct value for cases like 'pollution' – 'pollution from land based source'.

For German and Swedish we often have to perform decompounding to find the lexical head (we do this for English and French as well, but there the effect is not as big). We described the decompounding in Sect. 3.1.1.

6.2.5 Difference in distributional entropy

This feature has a lot in common with the subsumption feature described above. We already mentioned Caraballo and Charniak's use of this feature in Sect. 2.5.5; here we look at how it applies to our data.

Basically, entropy measures the average amount of "surprise" you feel at the outcome of a process described by a random variable (such as rolling a die). If you have a biased die, which nine times out of ten rolls a six, you will normally not be very surprised at the outcome of a roll – the entropy is low (about 0.701). A fair die is of course different; its surprise value is higher and so is its entropy (about 2.585). The formula for calculating the entropy of a discrete random variable X is written:

$$H(X) = -\sum_{x \in X} p(x) \log p(x)$$

We have four features for describing the difference in distributional entropy between two terms – one feature for each context model (see Sect. 6.2).

Corroboration of Caraballo's method

We performed an experiment to see if we could confirm Caraballo and Charniak's hypothesis that a more general word (term in our case) has a higher distributional entropy than a more specific term, using the FMA ontology and the Wikipedia anatomy corpus. Again, the intuition is that the more general term is freer to occur in different kinds of contexts, whereas the specialized term tends to occur in more restricted settings.

For this experiment we build a distributional similarity model for English terms from the FMA ontology, occurring at least 50 times in the Wikipedia anatomy corpus. We assign each term a number according to the level at which it appears in the FMA hierarchy (starting with the root as 0). We then calculate the average entropy of the co-occurrence vectors of all the terms at each level in the ontology. The results when using a window size of 50 are shown in Fig. 6.5 and when using a window size 3 in Fig. 6.6.

The curves are not very smooth, but there is a distinct tendency in both that the entropy decreases as the hierarchy level increases.[5] Judging from this,

[5] The zigzag-like pattern described by the curves we credit to noise.

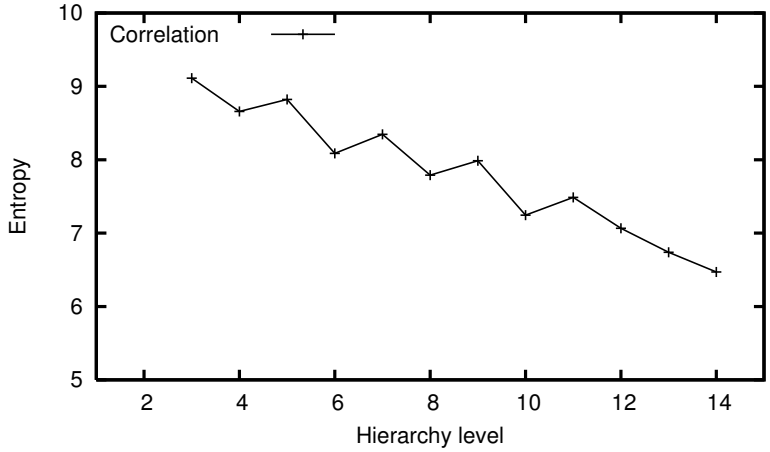


Figure 6.5: Window size: ±50 from the focus word. The graph maps the depth of the hierarchy to the average entropy of the co-occurrence vectors of terms at this hierarchy level in the FMA ontology.

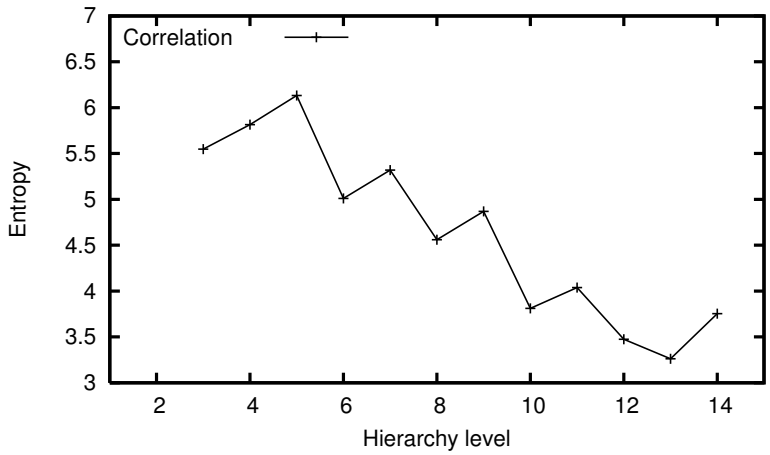


Figure 6.6: Window size: ±3 from the focus word.

there is useful information available from this kind of analysis and our experiments have provided additional support for Caraballo and Charniak's theory.

6.2.6 Difference in frequency

The previous section described Caraballo and Charniak's (1999) entropy-based measure of term specificity. In the same paper, the authors also find that the much simpler approach of assuming that the more frequent noun is the more general, is equally effective. We therefore add a feature describing the difference in frequency between the two terms.

We also have two features that give the absolute frequency of the two terms. The reasoning behind adding the absolute frequencies as features is that we believe that the dependability of some of the previously listed features will be influenced by the term frequencies. E.g., if two terms have a high distributional similarity but one or both terms occur infrequently in the corpus, the classifier should put less trust in this information. By including this information in our feature set, we leave the possibility of discovering such connections to the classifiers (given that the classifiers are non-linear).

6.2.7 Listing of all features

In order for us to be able to refer to the different features in a convenient way throughout the rest of the chapter, we list them all in Table 6.3, and give each feature an index and an identifier.

Table 6.3: Listing of all features used for identifying hyperonymy and cohyponymy. The indices and identifiers will be used to refer to the features throughout the remainder of the chapter.

Index	Identifier	Description
1	Subsumption-1st	The subsumption measure (Sect. 6.2.1), using first order co-occurrence
2	Subsumption-2nd-3	The subsumption measure, using second order co-occurrence and window size 3
3	Subsumption-2nd-50	The subsumption measure, using second order co-occurrence and window size 50
4	Subsumption-2nd-500	The subsumption measure, using second order co-occurrence and window size 500
5	Similarity-1st	Distributional similarity (Sect. 6.2.2), using first order co-occurrence and a non-reduced matrix
6	Similarity-2nd-3	Distributional similarity, using second order co-occurrence, window size 3 and a non-reduced matrix
7	Similarity-2nd-50	Distributional similarity, using second order co-occurrence, window size 50 and a non-reduced matrix

Continued on next page

Table 6.3 – continued from previous page

Index	Identifier	Description
8	Similarity-2nd-500	Distributional similarity, using second order co-occurrence, window size 500 and a non-reduced matrix
9	SVD-1st	Distributional similarity (Sect. 6.2.2), using first order co-occurrence and an SVD matrix
10	SVD-2nd-3	Distributional similarity, using second order co-occurrence, window size 3 and an SVD matrix
11	SVD-2nd-50	Distributional similarity, using second order co-occurrence, window size 50 and an SVD matrix
12	SVD-2nd-500	Distributional similarity, using second order co-occurrence, window size 500 and an SVD matrix
13	Hearst	Measure based on Hearst-patterns (Sect. 6.2.3)
14	Head-hyperonym	Head matching heuristic for hyperonymy (Sect. 6.2.4)
15	Head-cohyponym	Head matching heuristic for cohyponymy
16	Frequency-diff	Difference in absolute frequency (Sect. 6.2.6)
17	Entropy-diff-1st	Difference in distributional entropy (Sect. 6.2.5), using first order co-occurrence
18	Entropy-diff-2nd-3	Difference in distributional entropy, using second order co-occurrence and window size 3
19	Entropy-diff-2nd-50	Difference in distributional entropy, using second order co-occurrence and window size 50
20	Entropy-diff-2nd-500	Difference in distributional entropy, using second order co-occurrence and window size 500
21	Frequency-1	Absolute frequency of the first term in the term pair (Sect. 6.2.6)

Continued on next page

Table 6.3 – continued from previous page

Index	Identifier	Description
22	Frequency-2	Absolute frequency of the second term in the term pair

6.3 Merging Evidence across Languages

By using the translational equivalence relation (our subject of study in Chap. 5) for merging the data across languages, we are hoping to achieve a positive effect when training support vector machine classifiers, just as we did when learning a prototype-based ontology in Sect. 6.1. We test two main approaches for performing the merging: one assumes we have a domain-specific bilingual dictionary, the other assumes we do not. In the latter case we instead use the results of the automatic translation techniques presented in Sect. 5.2.

We were discussing how to view text in different languages previously – is it simply "more data" or are there special characteristics that should make us think of it differently? We will continue this discussion below and illustrate with examples from our corpora and ontologies.

6.3.1 Strategies for merging evidence

We start with the same scenario that we envisioned in Sect. 6.1, i.e., that we have access to a bilingual (or multilingual) domain-specific dictionary, but that we this time wish to learn a *terminological* ontology for the terms listed in it. We use the translation information that is encoded in the Eurovoc thesaurus for this purpose.

The English data forms our baseline for the experiments to follow. For each term pair in Eurovoc, where both terms occur at least once in the JRC-ACQUIS parallel corpus, we have 22 feature, as described in the previous section. We next look up the translation of the two English terms in Swedish. If the two Swedish terms also occur at least once in the Swedish texts, we add the 22 Swedish features to the English, giving us a total of 44 features for the term pair. We repeat the same process for German and French, so that we in the end have 88 features per term pair (not all of which need to be instantiated in all languages for all term pairs). Term pairs where one or both terms do not occur at least once in the English texts are excluded from the experiments.

In the second approach, we envision that we have access to the output of an ideal term extraction system (see Sect. 5.1) but that we do not have access to a translation dictionary. Instead of relying on the translation information in

Eurovoc, we this time use the output of the automatic translation system as the basis for merging the evidence across languages. In order not to confuse the classifiers, we would like to merge the data only in cases where we are relatively sure that the translations are correct, so we use the filtering method described in Sect. 5.2.2. This means that the translations will be correct in about 98.4% of the cases, but that we are only merging data for about 36.5% of the terms. The effects of weighing precision against recall here is a parameter in our system we have yet to explore; we discuss this again in Chap. 7.

We will be working with and contrasting three different datasets in the following experiments. The first uses strictly English data and we will be referring to this set as "Mono". The second set uses the results from merging data across languages (English, Swedish, German and French), where we use Eurovoc for translating the terms – we will refer to this set as "All" (because "all" languages are included). The third set instead uses the output of the automated translation system for merging, and this set we refer to as "MT" (for "machine translation").[6]

6.3.2 Idiosyncrasies of cross-language data

There are two extreme standpoints when considering cross language data, especially data taken from a parallel corpus. One could argue that, by adding texts translated into another language, we are in fact doubling the amount of data, basing the argument on crude numbers such as bytes used for storage or the like. The other extreme would be to say that we have not added anything at all, but are merely repeating exactly the same information, just using a different "encoding". Though the last point may be true in a sense, what we are in fact hoping for is that the use of a different language will reveal certain conditions that were hidden in the initial language. We will continue by looking at how switching languages can affect the features we are using.

Distribution-based features

A large number of the features we are using to train the classifiers are based on the distribution of terms or the joint distribution of term pairs. This is the case for the subsumption measure, for distributional similarity and for the difference in distributional entropy. Unless we have access to the perfect corpus, containing all relevant occurrences of our terms and no irrelevant occurrences, the distributional profile of a term will be fragmentary and contaminated by noise. However, switching languages, we are unlikely to have the *exact same* noise and fragmentation repeated; we can thus see the cross-language data as a way of abstracting away from such shortcomings, by giving the classifiers access to distributional data gathered from different languages. At the same

[6]Note that "machine translation" here does not refer to an entire machine translation system. Rather, it should be considered shorthand for "the word alignment part of a machine translation system".

time, we are of course introducing a new error source – new noise – by adding the cross-language data, but our hypothesis is that the overall effect will be positive.

We also expect differences in the distributional profile for a term in different languages, caused by the fact that homonymy and polysemy in a word or term rarely are kept intact when the word is translated. This is likely to have an effect on features that use second order co-occurrences. Consider the following sentence from the JRC-ACQUIS corpus, given in English, German and French, respectively (terms are marked in bold):

- *The* **European Police College** *shall have its* seat *in Bramshill.*
- *Die* **Europäische Polizeiakademie** *hat ihren* Sitz *in Bramshill.*
- *Le* **Collège européen de police** *a son* siège *à Bramshill.*

The term/proper name 'European Police College' (or its translations) occurs in all three sentences, along with a number of context words. We have emphasized the word 'seat' and its translations in the example; we focus on this word not because of any remarkable traits that it possesses, but rather because it's behavior is quite typical. We illustrate what happens when we translate the English 'seat' and the French 'siège' into German in Fig. 6.7.

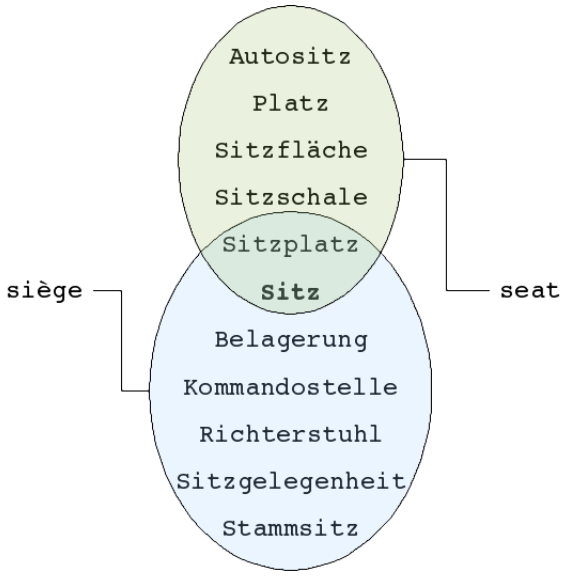


Figure 6.7: Non-transitivity of homonymy/polysemy.

Both 'seat' and 'siège' share the translations 'Sitz' and 'Sitzplatz'; English has an additional four and French an additional five translations that are not shared.[7] This discrepancy will lead to 'seat' and 'siège' having different dis-

[7] The translations are taken from http://leo.dict.org.

tributional patterns even in a parallel corpus, which will affect any feature making use of second order co-occurrences when measuring the similarity between terms – even though 'seat' itself is not a term. We believe that giving the classifiers access to cross-language distributional data will allow them to abstract away from homonymy and polysemy in one language by weighing in information from the others.

Pattern-based features

Our only pattern-based features are the ones where we use Hearst-patterns to recognize our lexico-semantic relations of interest. We do not see the same theoretical motivation for the usefulness of cross-language data for pattern-based features as we do for distribution-based features. However, the patterns we employ are rather rigid – if the text in one language happens to word an expression slightly differently from what we anticipated, our pattern matching component will miss it. Using more than one language then becomes a way of covering our bases – we get more chances of getting it right, working with more than one language. Below is an example from the JRC-ACQUIS corpus where the hyperonymy relation between 'vegetable oil' and 'olive oil' is captured in the German but not in the English, due to the use of ellipsis:

- *Für den Sektor Fette muß für die Bereitstellung von* **Olivenöl** <und anderen> **pflanzlichen Ölen** *eine zweckentsprechende Regelung getroffen werden.*
- *Whereas, in the oils and fats sector, appropriate rules and procedures should be laid down for mobilizing olive* <and other> **vegetable oils***;*

Terms are again marked in bold and the triggering Hearst-pattern is delimited with braces.

Head matching features

We make use of two head matching features when training the classifiers: one for recognizing hyperonymy and one for cohyponymy. Both depend on the lexical head of a term explicitly marking the hierarchical semantic relation it holds to its hyperonym – something which is rather an exception than a rule. We cannot expect the explicit expression of this relation to always occur for the same words in different languages. Whether it is expressed or not depends on different factors such as language-specific word formation rules and word-specific lexicalization factors. Table 6.4 contains a series of words from Eurovoc where the relation is not expressed in the English but is expressed in at least one of the other languages.

English	German	French	Swedish	Hyperonym
jurisdiction	gerichtliche **Zuständigkeit**	**compétence** juridictionelle	domstolars **behörighet**	competence
concessionaire	Vertrags**händler**	concessionnaire	general**agent**	trader/agent
cohabitation	freie **Partnerschaft**	**union** libre	sambo	partnership
incorporation	Gesellschafts**gründung**	**constitution** de société	bolags**bildning**	formation
conglomerate	Misch**konzern**	conglomérat	konglomerat	concern
cannery	Konserven**fabrik**	conserverie	konserv**fabrik**	factory
assessment	Leistungs**kontrolle**	**contrôle** des connaissances	kunskaps**kontroll**	inspection
crustacean	Krebs**tier**	crustacé	kräft**djur**	animal
devolution	Dekonzentration	déconcentration	begränsat själv**styre**	government
delinquency	Straffälligkeit	délinquance	kriminellt **beteende**	behavior

Table 6.4: *Explicit expression of hyperonymy relations in different languages. The fifth column contains an approximate translation of the hyperonym, marked in bold in the non-English relevant cases.*

6.4 Training Classifiers for Recognizing Related Term Pairs

We train two classifiers: one for recognizing hyperonymy relations and one for recognizing cohyponymy relations. We use the features listed in Table 6.3 for training the hyperonym classifier; for the cohyponym classifier, we use a subset of the features, listed further down in this section (Table 6.7). We form the classifier data by taking all ordered pairs of terms within each partition of Eurovoc (the partitioning is described in Sect. 3.3.1). When preparing data for the hyperonym classifier, we mark all term pairs (t,u) as positive examples where t is the direct parent of u in the Eurovoc thesaurus, and we do the same for siblings for the cohyponym classifier. All other term pairs are marked as negative examples, so the positive examples for one classifier are negative for the other.

We diverge from Cimiano (2006) and Snow et al. (2006) by not using the *transitive closure* of the dominance relation when collecting data for the hyperonym classifier. This is a deliberate choice, since we believe that the direct dominance relation will give more prototypical examples of hyperonyms than the transitive closure of the relation will. We return to discuss this point in Chap. 7.

There are many more term pairs within each partition of the ontology that are unrelated to each other than there are hyperonym or cohyponym pairs. We have a total of 1,078,480 examples, out of which 2,030 are positive hyperonym examples and 13,078 are positive cohyponym examples. This means that the data we use to train the classifiers is skewed (it contains many more negative examples than positive) and that we must find a strategy for dealing with this skewness – otherwise the classifiers might rationalize the problem by simply classifying every term pair as unrelated. We discuss this further in Sect. 6.4.2 below.

When evaluating the classifier results, we also have to take the data skewness into consideration. If we consider the overall accuracy of the classifier, we would get 99.8% correct for the hyperonym classifier and 98.8% correct for the cohyponym classifier, simply by classifying all examples as negative. Instead, the standard way of evaluating such datasets is to consider precision, recall and F-score of the positive class only and this is the evaluation method we employ in the following experiments.

6.4.1 Single feature classifiers

Our first step is to look at each single feature in isolation and examine how effective a classifier based solely on this information could be. Because all our features are either binary or give a value in the range $[0,1]$,[8] we use a threshold that lies within this range. We initialize the threshold to 0 and then raise it gradually to 1 by steps of 0.01 at a time. At each step, we classify all examples at or above the threshold as positive and the rest as negative, and we report the results for the threshold that gives the highest F-score. We perform the experiment on the Swedish data, using partition 2 of Eurovoc, described in Sect. 3.3.1 (this partition was chosen because it is about average in size among the ten partitions).

Table 6.5 gives the precision, recall and F-score for the hyperonym data and all our features (the absolute frequency features, 21–22 in Table 6.3, are left out). We also give scores for a measure which we derive by calculating the arithmetic mean of the scores of all features, and a further measure, which is the most effective guessing approach (Table 6.6). It may seem counterintuitive that our best guessing strategy should consist in guessing that all examples are positive (hence the recall of 1.0 for this measure). This is so because no matter

[8] An exception is the feature measuring the difference in frequency, which we therefore normalize first.

how many examples we guess to be positive, we will always get about 0.25% of them correct (on this dataset), so we might as well maximize the recall and guess that all are positive. Because all asymmetrical features (those that do not give the same result for the pair (t,u) as for the pair (u,t)) are designed for the hyperonym classifier, we do not report on results for them for the cohyponymy relation (Table 6.7).

To exemplify: 'vessel' is listed as a hyperonym to 'tanker' in Eurovoc. The scores for features 1–4 (the subsumption features) for this term pair are (0.655, 0.640, 0.869, 0.934). At a threshold of 0.86, features 3 and 4 would classify the pair as hyperonym/hyponym, whereas features 1 and 2 would say they are unrelated. The "Average" measure adds these four features and the 16 others (remember that we leave out the absolute frequencies of the terms for this particular experiment) and calculate the arithmetic mean (0.369 in this case, thus negative at this threshold).

The results are better than what to expect if we were to apply these classifiers to unseen data – remember that we have chosen the thresholds to precisely fit this dataset. The single best feature, according to the F-score, is the "Head-hyperonym" feature (the head matching heuristic for hyperonym detection). The Hearst-patterns give a higher precision but a lower recall. The best distribution-based feature is "Subsumption-2nd-50" (the subsumption measure using second order co-occurrences and a context window size of 50 words). Note that the "Average" measure does not perform better than the "Head-hyperonym" feature alone. All features (except "Head-cohyponym", for obvious reasons) are significantly better than guessing.

Table 6.7 contains the results for the cohyponym classifier. For this dataset, the "Head-cohyponym" feature has the highest precision, but recall is low and it has the lowest F-score of all features. The "Average" feature (Table 6.8) performs best here, while "Similarity-2nd-3" (distributional similarity, second order co-occurrence, context window size 3) is the single best feature. Again, all features improve on the "Guess" feature, which means that they all contain information useful for a classifier. We do not claim the results in Tables 6.5–6.8 will look the same for all languages or partitions, but they will function as a frame of reference when we look at the performance of the support vector machine classifiers presented in the following section.

Figure 6.8 gives a different way of looking at how well each feature distinguishes between related and unrelated term pairs. Features that have a clear separation of the blue (unrelated), green (hyperonymy) and red (cohyponymy) lines should be the most useful for the classifier (note that we are then only taking linear separation of the data into consideration). We see, as predicted, that features based on subsumption (1–4) and difference in entropy (17–20) separate the hyperonyms rather well. It is interesting to see the behavior of the distributional similarity measures when it comes to differentiating between hyperonyms and cohyponyms (unrelated term pairs are easier to distinguish for these features – also according to expectation). For first order co-occurrence

Index	Identifier	Precision	Recall	F-score
1	Subsumption-1st	0.0380	0.116	0.0572
2	Subsumption-2nd-3	0.0344	0.159	0.0565
3	*Subsumption-2nd-50*	*0.0851*	*0.0976*	*0.0909*
4	Subsumption-2nd-500	0.0673	0.0854	0.0753
5	Similarity-1st	0.0279	0.0671	0.0394
6	Similarity-2nd-3	0.0295	0.0915	0.0446
7	Similarity-2nd-50	0.0331	0.0610	0.0429
8	Similarity-2nd-500	0.0392	0.0732	0.0511
9	SVD-1st	0.0215	0.0732	0.0332
10	SVD-2nd-3	0.0095	0.0915	0.0173
11	SVD-2nd-50	0.0171	0.122	0.0300
12	SVD-2nd-500	0.0291	0.0305	0.0298
13	Hearst	0.800	0.0244	0.0473
14	**Head-hyperonym**	**0.458**	**0.0671**	**0.117**
15	Head-cohyponym	0.0	0.0	0.0
16	Frequency-diff	0.0329	0.0488	0.0393
17	Entropy-diff-1st	0.0140	0.152	0.0256
18	Entropy-diff-2nd-3	0.0146	0.159	0.0267
19	Entropy-diff-2nd-50	0.0157	0.0976	0.0270
20	Entropy-diff-2nd-500	0.0144	0.140	0.0262

Table 6.5: *Results using an optimal threshold and single features, partition* 2*, Swedish hyperonym data. The best F-score is marked in bold, the second best is written in italics.*

	Precision	Recall	F-score
Average	0.0712	0.152	0.0971
Guess	0.0025	1.0	0.0050

Table 6.6: *Results for the best guessing strategy ("Guess") and a measure which we calculate by taking the arithmetic mean of the scores of all features in Table 6.5 ("Average"). Evaluated on partition 2, Swedish hyperonym data.*

and second order co-occurrence with a small context window (features 5–6 and 9–10), hyperonyms get a (slightly) higher score, for bigger context windows (features 7–8 and 11–12) the situation is reversed or neutral. We see no

index		Precision	Recall	F-score
5	Similarity-1st	0.0431	0.223	0.0722
6	**Similarity-2nd-3**	*0.0542*	*0.121*	*0.0750*
7	Similarity-2nd-50	0.0345	0.128	0.0543
8	Similarity-2nd-500	0.0386	0.149	0.0613
9	SVD-1st	0.0425	0.104	0.0604
10	SVD-2nd-3	0.0734	0.0447	0.0556
11	SVD-2nd-50	0.0444	0.0872	0.0589
12	SVD-2nd-500	0.0417	0.0979	0.0585
15	Head-cohyponym	0.117	0.0340	0.0527

Table 6.7: *Results using an optimal threshold and single features, partition 2, Swedish cohyponym data. The best F-score is marked in bold. All asymmetric features in Table 6.5 are left out.*

	Precision	Recall	F-score
Average	0.0596	0.123	0.0804
Guess	0.0140	1.0	0.0280

Table 6.8: *Results for the best guessing strategy ("Guess") and the arithmetic mean measure ("Average"). Evaluated on partition 2, Swedish cohyponym data.*

clear differences between the raw distributional similarity data (5–8) and the data where singular value decomposition has been applied (9–12).

6.4.2 Support vector machines

A short and readable introduction to support vector machines (SVMs) can be found in Schölkopf (1998). In short, SVMs are a set of machine learning methods that can be used for classification or regression. They are used to find the *maximum margin hyperplane* that separates two datasets, or, rather, the *support vectors* that define this hyperplane. An example is shown in Fig. 6.9.

Figure 6.10 shows an example taken from the Swedish data, partition 2, with 100 datapoints randomly selected from each of the three classes: cohyponymy, hyperonymy and unrelated terms. We use only two dimensions (i.e., features) in this example because it is hard to represent more on a two-dimensional surface. The task of the SVM thus consists in separating the datapoints by class in all 22 dimensions. For the particular features shown in this example (features 4 and 8), we can see that the task of separating related from

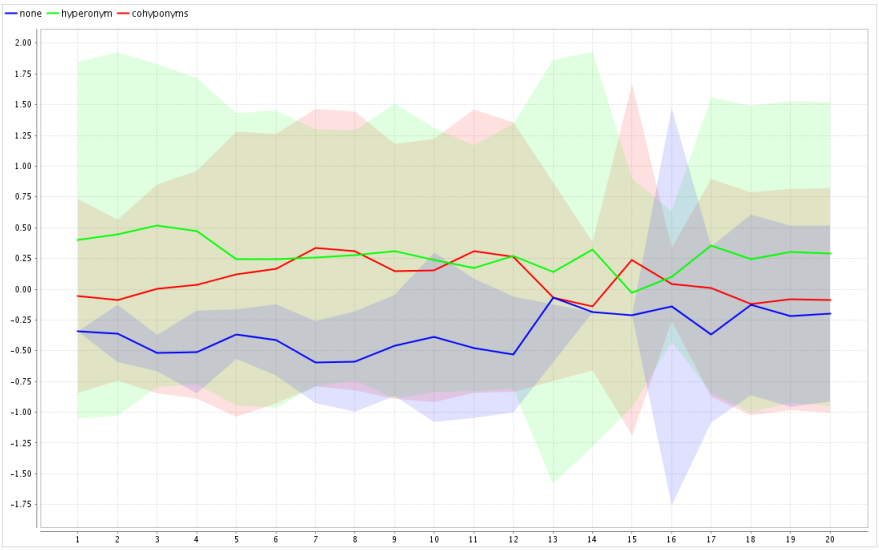


Figure 6.8: Separation per feature for Swedish data, partition 2, 100 randomly selected term pairs per class. Solid lines indicate average values (the data has been normalized), shadowed areas show the variation. Blue represents unrelated terms, red is for cohyponyms and green for hyperonyms. Feature indices on the x-axis coincide with those in Table 6.3.

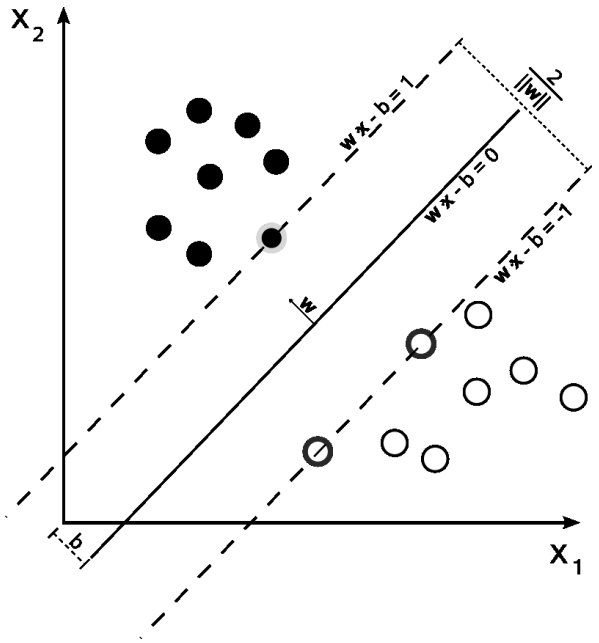


Figure 6.9: Source: "Svm_max_sep_hyperplane_with_margin.png", taken from Wikimedia Commons (http://commons.wikimedia.org), showing the maximum margin hyperplane.

unrelated term pairs is much easier than distinguishing between the two different types of relatedness.

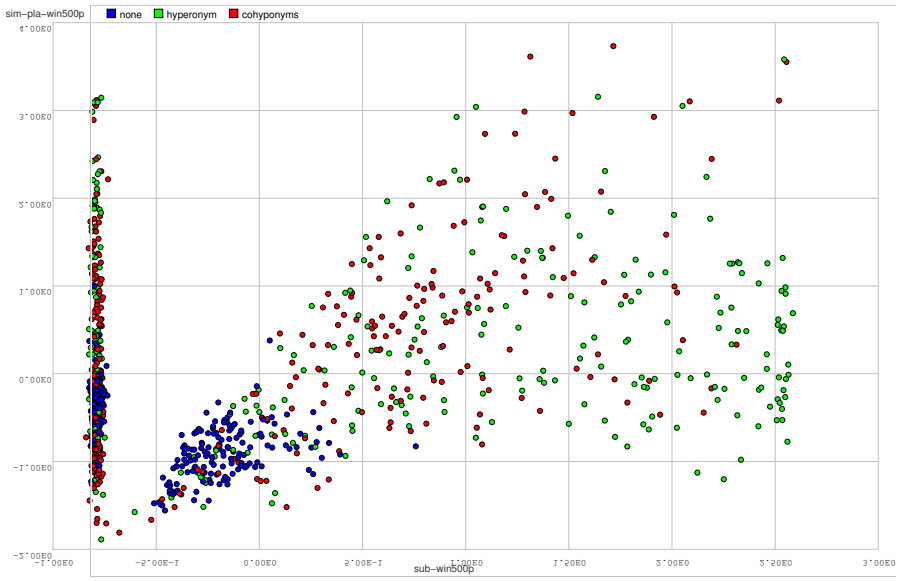


Figure 6.10: Scatter plot for SVM problem. Features "Subsumption-2nd-500" is shown on the x-axis and "Similarity-2nd-500" is shown on the y-axis. The data is from Swedish, partition 2. Hyperonyms are shown in green, cohyponyms in red and non-related terms in blue.

We use an SVM implementation called "libsvm" (Chang and Lin, 2001). As a final preparatory step, for each partition, we scale the training data so that the values of all features lie in the interval [0,1], and the test data is in turn scaled on the basis of the training data. Both these steps we perform using the tool "svm-scale", which is part of the "libsvm" package. We keep all other "libsvm" parameters in their original, default settings.

To come to grips with the aforementioned problems with skewed data, we set aside one data partition (number 9) to use as a development set. This means we exclude partition 9 from all further experimentation, to avoid mixing testing and training data.[9] Cimiano (2006) tries over-sampling (copying positive examples to increase their number) and under-sampling (removing negative examples) but achieves the best results with a one-class SVM (where only the positive examples are considered). We were unable to reproduce Cimiano's favorable results for the one-class SVM and instead we use the option of assigning more weight to the positive examples, so that each mistake for the positive class is counted as more serious than mistakes for the negative class. In libsvm this is done through a parameter *w* (for "weight"), which we set to 32 for the positive class, for both classifiers. This value was set through

[9]Because of this, we are performing 9-fold cross validation in all experiments instead of the standard 10-fold.

tests on partition 9, where we increased the weight logarithmically in the interval [1,128]. An appealing aspect of this approach is that we are neither throwing data away (as with under-sampling), nor reproducing data (as with over-sampling), just shifting the focus of the classifier towards the cases we are most interested in.

We also perform an optimization, again on partition 9, for both classifiers for selecting the kernel, where we compare the linear, polynomial (2nd and 3rd degree) and the RBF (radial basis function) kernels. We achieve the best results with the RBF kernel for both classifiers – a non-linear kernel, which effectively increases the dimensionality of the original feature set.

Cross-language SVM classification

We perform a series of tests on the classifier data for partition 2, using the other partitions (excluding partition 9) for training. We use partition 2 to get results comparable to those in Sect. 6.4.1. Each training and testing run lasts for anything between one and three days on a Pentium 4 architecture, so we are therefore not able to perform a cross validation using all partitions and languages on this experiment set. Tables 6.9–6.14 give the F-score, precision and recall for the SVM classifiers, using single language data, as well as all possible combinations of data combining two languages.

Assume we are looking at the hyperonymy data and that we have a recall of 0.1. This means that, out of all term pairs that are related via the hyperonymy relation in the gold standard, the classifier has identified 10% correctly. If we for the same data have a precision of 0.3, this means that 30% of the term pairs that the classifier has identified as hyperonym/hyponym pairs are also marked as such in the gold standard. The F-score balances recall and precision against each other by giving the harmonic mean (0.15 in this case).

	Single	Swedish	German	French	English
Swedish	0.125	-	*0.155*	0.144	0.148
German	0.123	**0.183**	-	0.107	0.164
French	0.0909	0.109	0.0668	-	*0.115*
English	0.0975	0.120	*0.134*	0.122	-

Table 6.9: *Comparing single language and all combinations of dual-language models, hyperonym classification task, F-score on the positive class, partition 2. The highest score overall is marked in bold, the highest score per language is marked in italics. The rows identify the language used as the basis for the model, and the columns indicate which language is used to extend the base model ("Single" means that just the base model is used).*

For the hyperonym classification task, we see a dramatic increase in precision for the dual-language data, with little or no decrease in recall (some

	Single	Swedish	German	French	English
Swedish	0.174	-	**0.500**	0.192	0.187
German	0.194	*0.415*	-	0.152	0.274
French	0.0809	0.110	0.0778	-	*0.117*
English	0.0975	0.140	*0.181*	0.132	-

Table 6.10: *Single and dual-language models, hyperonym classification task, precision on the positive class.*

	Single	Swedish	German	French	English
Swedish	0.0976	-	0.0915	0.116	**0.122**
German	0.0897	*0.117*	-	0.0828	*0.117*
French	0.104	0.108	0.0586	-	*0.113*
English	0.0975	0.106	0.106	*0.114*	-

Table 6.11: *Single and dual-language models, hyperonym classification task, recall on the positive class.*

	Single	Swedish	German	French	English
Swedish	0.0879	-	0.0768	*0.114*	0.0997
German	0.105	0.116	-	*0.131*	0.125
French	0.126	0.125	**0.133**	-	0.119
English	0.108	0.114	0.123	*0.124*	-

Table 6.12: *Comparing single language and all combinations of dual-language models, cohyponym classification task, F-score on the positive class.*

	Single	Swedish	German	French	English
Swedish	0.0642	-	0.0714	*0.0812*	0.0725
German	0.0826	0.0909	-	*0.103*	0.0987
French	0.0952	0.0953	**0.105**	-	0.0903
English	0.0808	0.0881	*0.0991*	0.0949	-

Table 6.13: *Single and dual-language models, cohyponym classification task, precision on the positive class.*

language combinations give an increase in both at once). There is also a tendency for combinations of Swedish and German data to give good results, as

	Single	Swedish	German	French	English
Swedish	0.139	-	0.0830	**0.189**	0.160
German	0.146	0.161	-	*0.179*	0.170
French	*0.185*	0.184	0.181	-	0.176
English	0.162	0.161	0.162	*0.181*	-

Table 6.14: *Single and dual-language models, cohyponym classification task, recall on the positive class.*

well as for combinations of English and French. Improvements on the single language results are less dramatic for combinations *across* these two groups, with the exception of English and German.

The dual-language data generally also improves the results for the cohyponym classification task, but to a smaller extent. The language pairs that gave the biggest improvements for hyperonym classification do not stand out in the same way for the cohyponym classification. We do not have a theoretical explanation for the cases of reduced F-score (one for the hyperonym classifier and two for the cohyponym classifier), but since we have not performed cross-validation on this data, it is possible that the picture would change for the other partitions (this of course holds for possible further reductions in F-score as well).

We perform the same experiments, this time *with* cross validation, on all the English data, comparing the results using the "Mono", "All" and "MT" datasets. We use only the English data as our baseline here, because of the long running times for these experiments, as discussed previously in this section. We give the results in Tables 6.15–6.16.

	Mono	All	MT
F-score	0.151	**0.179**	0.174
precision	0.202	0.234	**0.435**
recall	0.139	**0.154**	0.112

Table 6.15: *9-fold cross validation, "Mono", "All" and "MT" data, hyperonym classification task, scores on the positive class.*

Both the "All" and the "MT" data give increased F-scores for both classifiers. Again, we see that using the "MT" data results in higher precision, while the "All" data improves on both recall and precision at once. The improvement in F-score is again bigger for the hyperonym classifier and we also see, contrary to what seemed to be the case in Tables 6.9 and 6.12, that the cohyponym classifier is more accurate than the hyperonym classifier. Remember, as we

	Mono	All	MT
F-score	0.204	**0.213**	0.208
precision	0.159	0.160	**0.175**
recall	0.297	**0.344**	0.275

Table 6.16: *9-fold cross validation, "Mono", "All" and "MT" data, cohyponym classification task, scores on the positive class.*

discussed in Sect. 6.4.1, that using this method of evaluation, the cohyponym classifier has a slightly easier task to solve than the hyperonym classifier.

The most striking figure in Tables 6.15–6.16 is the remarkable increase in precision for the hyperonym classifier using the "MT" data. The non-linearity of the classifiers makes it hard to say just what is causing this change, but one hypothesis is that the classifier switches to looking more exclusively at the high-precision features, such as the head matching heuristic or the Hearst-patterns. Since many feature values are simply missing (because many terms are not translated), the low-precision features become even less reliable, forcing a change of focus towards the high-precision features.

In Sects. 6.4.1–6.4.2, we have shown how it is possible to gradually increase the performance of our two classifiers. Going from our naïve single feature classifiers, we saw how formulating the problem for an SVM classifier brought performance up a notch. We have now shown that further qualitative improvements are possible, by taking a cross-language perspective on the problem of recognizing hyperonymy and cohyponymy, and that automatic word alignment techniques can be used to, at least partially, achieve these improvements, in the absence of a domain-specific multilingual dictionary.

6.5 Probabilistic Ontology Learning

We discussed previous efforts towards incorporating probability theory in ontology learning approaches in Sect. 2.5.7. Our approach is based on the method described by Snow et al. (2006), with some alterations, and we add a cross-language perspective by using probabilities based on evidence from different languages. Snow et al. define two base relations that are used to construct the hierarchy: hyponymy (and its inverse relation hyperonymy) and cohyponymy. The hierarchy is constructed by iteratively adding instances of either of the two relations, in such a way as to maximize the probability of the resulting hierarchy after each step. After a few terms have been added to the hierarchy, one is no longer free to add new relations at will, but the hierarchy constructed to that point imposes restrictions on any new relations

to be added (e.g., term t and term u cannot be cohyponyms if t has already been specified as the hyperonym of u).

A new relation added to the ontology sometimes *implies* a further set of relations, because of the resulting new structure of the hierarchy. The probability of these implied relations is also taken into account when considering which instance of which of the two base relations to add next to the hierarchy. Snow et al. thus introduce a global perspective, considering the hierarchical structure in its entirety, into their ontology learning process; something which is absent in the (otherwise similar) approach suggested by Ryu and Choi (2006). We will study some specifics of Snow et al.'s approach alongside with that of our own in the rest of this section and also look at experimental results.

6.5.1 Selecting the best relation to add

The algorithm we describe here is greedy for reasons of tractability; generating and testing every possible configuration for arranging the terms hierarchically would be much too expensive. We therefore try to find, in each step, the relation to add that will maximize the probability of the ontology at that given point. Assume we have an ontology O, evidence used to learn the ontology E (in our case the output of the SVM classifiers), two terms t and u and a relation R_{tu} between the two terms. We then, along with Snow et al., calculate the multiplicative change in the overall probability of ontology O, caused by adding R_{tu}, like this:

$$\Delta_O(R_{tu}) = k\left(\frac{P(R_{tu} \in O|E^R_{tu})}{1 - P(R_{tu} \in O|E^R_{tu})}\right) \tag{6.1}$$

In the above formula, k is considered a constant, independent of O, t and u. Snow et al. use the same k for both base relations in their experiments; we investigate the effect of using different k values for the different base relations in our experiments in Sect. 6.5.2.

As mentioned, adding a relation R_{tu} to the ontology is likely to not only affect terms t and u, but also to result in other terms in the ontology entering into new relations with each other or with t or u. When considering whether or not to add R_{tu} to the ontology, we also want to consider the probability of the set of relations implied by R_{tu}. We denote this implied set $I(R_{tu})$ and we calculate the total multiplicative change for adding R_{tu} to O using the following formula:

$$\Delta_O(I(R_{tu})) = \prod_{R \in I(R_{tu})} \Delta_O(R) \tag{6.2}$$

This is the formula we use in each step of the iterative ontology learning process, to determine which relation maximizes the return value and thus which relation to add to the ontology.

Snow et al. also introduce a threshold for Equation 6.2 (the threshold is 1 in their case) for determining when to stop adding relations to the ontology. This threshold allows a trade-off between recall and precision; using a high threshold means that we only add the most probable relations to the ontology but also that many terms and relations will not be included. We examine the effect of altering this threshold in our experiments in Sect. 6.5.2.

For each ordered pair of terms in our term set, we have stronger or weaker evidence that either of the two base relations holds. The evidence in our case is based on the scores from the SVM classifiers, described in Sect. 6.4. Classifiers typically give a score in the range [-1,1], so first of all we normalize the score so that it instead lies in the range [0,1]. To achieve this, we perform a sigmoid transformation of the scores, using the following formula:

$$P(s) = \frac{1}{1+e^{-s}} \tag{6.3}$$

where s is the classifier score. One can view this as a way of approximating the probability of an event by fitting the data to a sigmoid (S-shaped) curve.

We can control the *average branching factor* of the ontology by altering the constant k in Equation 6.1. Setting a higher k for the cohyponymy relation makes the algorithm biased towards adding instances of this relation to the ontology rather than instances of the hyperonymy relation, which in turn results in a higher average branching factor for the ontology. Which average branching factor to strive for will differ depending on the domain, available resources and intended application of the ontology. In Chap. 7, we discuss a possibility for finding an optimal value for k, based on the classifier scores and derived probabilities. We investigate the effects of using different k values experimentally in Sect. 6.5.2.

Implied relations

We consider the hyperonymy relation to be transitive, meaning that if t is a hyperonym of u in our ontology, and we wish to add that u is a hyperonym of v, this implies a hyperonymy relation between t and v. So far, we are in agreement with Snow et al. (2006), but when it comes to cohyponymy, we only consider relations of the first degree, whereas they take other variants of the relation into consideration. For this purpose, they introduce the notion of (m,n)-cousinhood, where a term t is an (m,n)-cousin of term u, if the shortest path from t to u in the ontology consists in going m steps up and n steps down. E.g., in Fig. 6.11, 'hard cheese' and 'cream' are $(2,1)$ cousins, and 'pasteurised milk' and 'soft cheese' are $(2,2)$ cousins.

Instead of just training a cohyponym classifier, as in our approach, Snow et al. train separate (m,n)-cousin classifiers, one for each (m,n).[10] They can afford this because they use a radically different cohyponym classifier – using our SVM classifier, this would be far too inefficient.

[10] (m,n) is considered equal to (n,m)

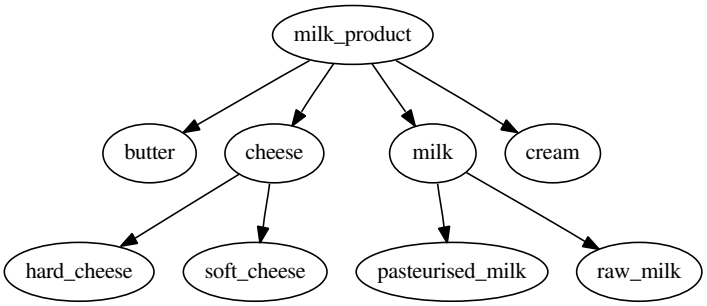


Figure 6.11: Eurovoc excerpt, terms involving milk products.

Apart from practical issues, we see a conceptual problem with (m,n)-cousinhood as well, in that it is not intuitively clear what this relation represents. Adding additional layers of the hyperonymy relation between two terms keeps the is-a link between them intact, while increasing the level of abstraction of the connection. Additional layers added between two terms related by the cohyponymy relation create a new type of relation, the nature of which is not well defined.

In Sect. 6.4, we discussed not using the transitive closure of the hyperonymy relation when assembling the training data for the SVM classifiers. This decision was taken in order to make sure that only the most prototypical term pairs are used as positive examples when learning the hyperonymy relation. The fact that we are now looking at the transitive closure for hyperonymy when calculating the multiplicative change stands in contrast but not in contradiction to our previous decision. It is still possible to learn chains of hyperonyms, as long as all subordinate terms in the hyperonym chain look like typical hyponyms of all terms at higher levels. We return to this discussion again in Chap. 7.

6.5.2 Experimental results

We report results on the three sets of data for all experiments, as described in Sect. 6.3: “Mono”, “All” and “MT”. We compare the three resulting models by using them to build ontologies in our probabilistic ontology learning system. The learned ontologies are evaluated using our PMCC measure and the CSC measure, both discussed in Sect. 4.3. We showed in Sect. 4.3.2 that the CSC measure is sensitive to whether or not the root element is shared between the reference and the learned ontology. In order not to overestimate the similarities when testing with the CSC measure, we use a version of Eurovoc where the root concept has been changed to a neutral one (not occuring in the learned ontology) – the same setting that we used for the experiment shown in Fig. 4.8.

Varying the average branching factor

As mentioned, we can influence our ontology learning system towards building flatter or deeper trees, with higher or lower average branching factors, by using different values of k in Equation 6.1. We compare the results for different k values and all three models to the Eurovoc thesaurus. If we use k for the cohyponymy relation, we use $k' = 1 - (k - 1)$ for the hyperonymy relation – in other words, when the constant for one relation goes up, the constant for the other relation goes down, and vice versa. The results for the PMCC measure are given in Fig. 6.12 and for the CSC measure in Fig. 6.13.

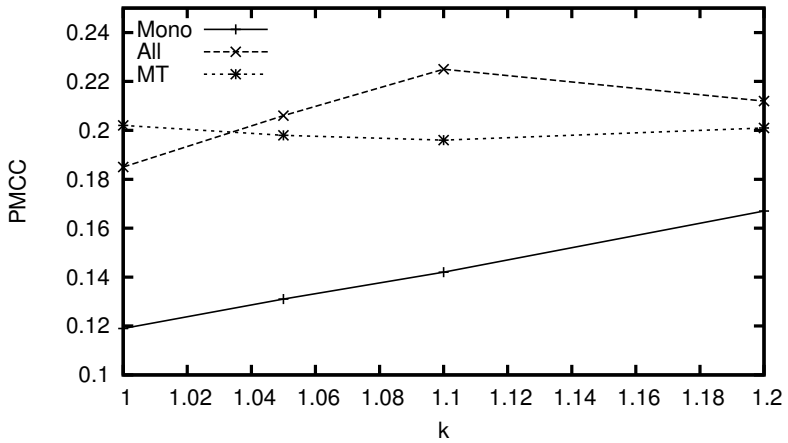


Figure 6.12: Increasing values of k for cohyponymy, 9-fold cross validation, single and cross-language data, evaluated with the PMCC measure.

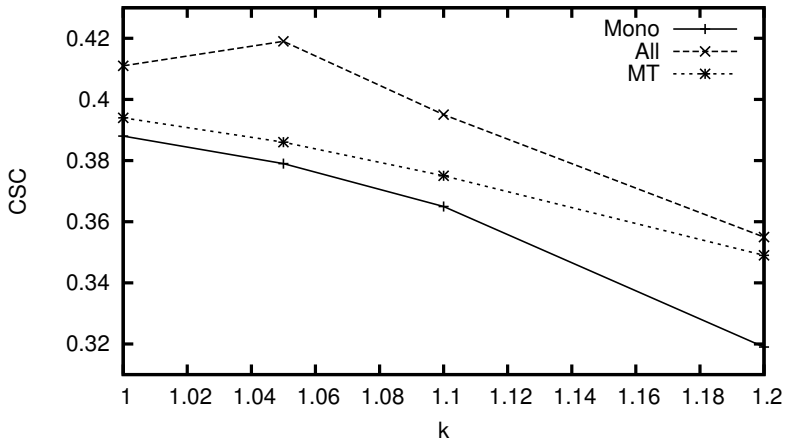


Figure 6.13: Increasing values of k for cohyponymy, 9-fold cross validation, single and cross-language data, evaluated with the CSC measure.

We set the threshold to 1 for the multiplicative change (same as used in Snow et al., 2006) for deciding when to stop adding new relations to the ontology in these experiments. This means, in our case, that we are on average adding about 80% of the terms scored by the classifiers to the ontology (same for all three models); for the rest, the evidence of them entering into any relation with the other terms is deemed too loose.

We see a sizable increase in the PMCC measure when using the "All" and the "MT" data, compared with using the "Mono" data. For the "All" data, the increase lies between 27% – 55%, depending on the k value, and for the "MT" data it lies between 20% – 70%. The results for the "MT" data lie almost flat across the different k values, the "All" data results peak at $k = 1.1$ and the "Mono" data shows an almost perfect linear increase. We stop investigating k values at 1.2, because beyond that, no or very few hierarchical relations are added, which defeats the purpose of our efforts. Using $k < 1$ would mean producing a tree with a branching factor less than two, and this is intuitively the wrong way to go.

Turning to the results for the CSC measure (Fig. 6.13), the picture is different. We still see improved results for the "All" and "MT" data, but on a smaller scale: between 6% – 11% for the "All" data and 2% – 9% for the "MT" data. The "All" data results peak at $k = 1.05$, otherwise the tendency is that the CSC measure decreases as k increases – practically the inverse situation to what we saw for the PMCC measure. This should not come as a big surprise; as discussed in Sect. 4.3, the CSC measure is "blind" to horizontal relations. As we increase the k value for the cohyponymy relation, fewer hyperonymy relations are added and the CSC score drops.

Choosing the appropriate k value for our purposes thus proves a less straightforward decision than we had hoped. We settle on using $k = 1.05$ for our future experiments strictly as a compromise between the results from the two evaluation measures. At $k = 1.05$, the CSC measure has not dropped too much from its state at $k = 1.0$, and the PMCC measure has had the chance to improve slightly. To some degree, setting $k = 1.05$ is an arbitrary decision, but the most important issue for the next experiment series is the *fixation* of k, rather than the value assigned.

Varying the probability threshold

The experiments described here have the same basic setup as the ones in the preceding section, and once again we compare results using the three different datasets. We fix the value of $k = 1.05$ and instead measure what happens when we vary the threshold for when to stop adding new relations to the ontology. Snow et al. use a threshold of $t = 1$ in their system; here, we vary t in the range [0,2], starting at 0 and increasing it by steps of 0.5 for each run. We give the results for our PMCC measure in Fig. 6.14 and for the CSC measure in Fig. 6.15.

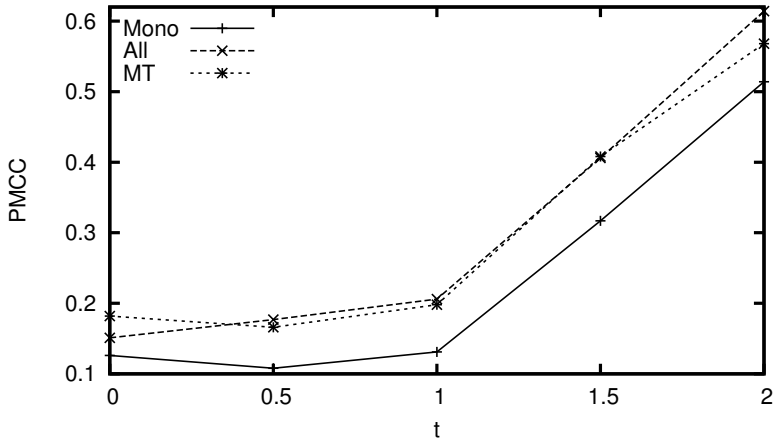


Figure 6.14: Increasing values of t, 9-fold cross validation, single and cross-language data, evaluated with the PMCC measure.

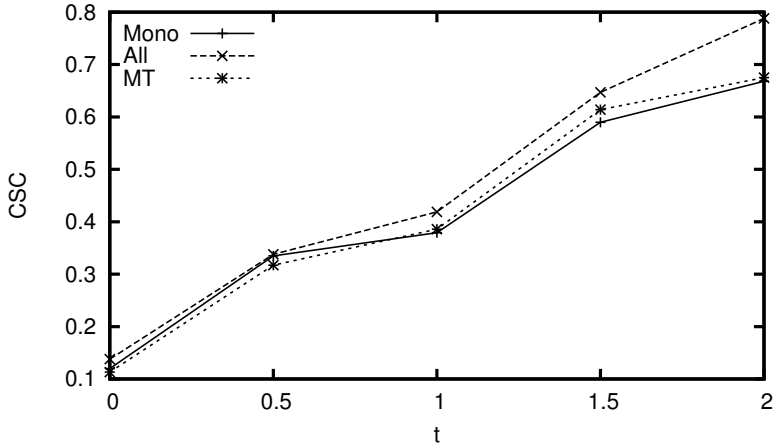


Figure 6.15: Increasing values of t, 9-fold cross validation, single and cross-language data, evaluated with the CSC measure.

We see an increase in PMCC value using the "All" data over the "Mono" data of between 19% – 64%; for the "MT" data the same figures are 11% – 54%. The difference between using the "All" data and the "MT" data seems small here, with the "MT" data even giving slightly better results at $t = 0$. We see this as an effect of the hyperonym classifier working with higher precision when using the "MT" data (see Table 6.15). Using a more conservative strategy, trading recall for precision, is probably a good approach at this threshold level ($t = 0$).

For the CSC measure, the picture again is changed. The "All" data boosts the results with between 1%–15% but the "MT" data makes the result vary between -6%–4%. It seems that the conservative strategy that was useful for

the PMCC evaluation here becomes a disadvantage. The high precision of the added hyperonymy relations cannot fully compensate for the loss in recall, when evaluating with the CSC measure (remember that cohyponymy relations are not considered by this measure).

Using a specific threshold, the number of concepts added to the ontology will vary slightly between our three datasets, depending on the distributions of the scores from the classifiers. Of course, it is easier to score high in these evaluations if one only adds relations for which the evidence is clear. In other words: the more concepts and relations added to the ontology, the harder the task, given that there is a sliding scale of certainty for the validity of the evidence for relations between term pairs. To verify that the differences in the evaluations in Tables 6.12–6.15 are not caused by such differences in lexical coverage, we also plot the percentage of concepts added against the different values of t for all three datasets in Fig. 6.16.

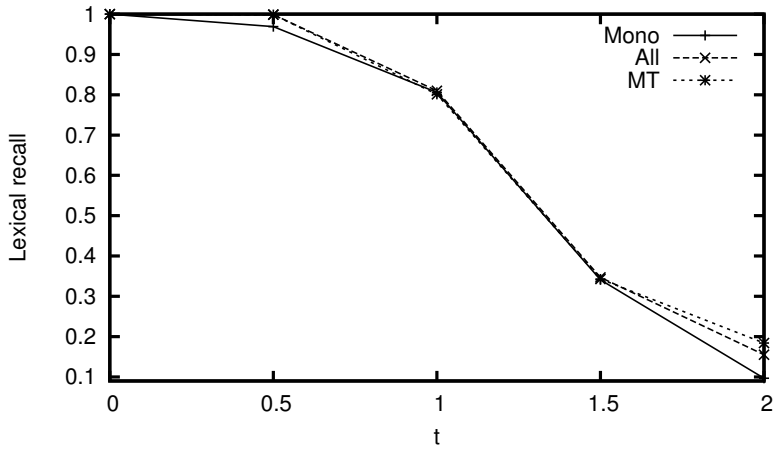


Figure 6.16: Increasing values of t, 9-fold cross validation, single and cross-language data, percent of concepts added.

There is obviously little or no difference between the three models in this aspect. Where they do differ, it is towards the "All" and "MT" data including more terms than the "Mono" data. Differences to the advantage of using the cross-language data shown in Tables 6.12–6.15 are thus *not* due to the system solving an easier problem in those cases; if anything, it solves a slightly harder problem, with a considerable increase in accuracy.

6.5.3 Example system output

To give an idea of how the output of the system might look, we include part of the ontology for partition 3, learned using the "All" data, in Fig. 6.17 (we have to exclude parts of the structure due to the restricted size of the page).[11]

[11] The Appendix shows two additional examples of ontologies learned with our method.

We see that at the settings used to learn this ontology ($k = 1.0$, $t = 1.0$), a lot of confidence is placed in the head matching heuristic, both for hyperonymy and cohyponymy. But that is far from the whole story, especially for the cohyponymy relation; see, e.g., nodes 1, dealing with different kinds of cereals and grains, and 25, dealing with environmental issues. Also the hyperonymy relation shows some interesting departures from the head matching heuristic, e.g., between the terms 'meat' → 'beef', 'sweetener' → 'honey', and 'natural_resource' → 'mineral_resource' (the last example actually goes against the head matching heuristic for cohyponymy).

The numeric nodes in the learned ontology are produced when the ontology learning algorithm adds cohyponymy relations between terms not already in the ontology. By allowing the creation of these abstract nodes, we ensure that the best relation, according to the definition given in Sect. 6.5.1, gets added in each iteration of the algorithm (otherwise we would only be allowed to add cohyponymy relations under an already specified hyperonym). It on the other hand means that the structure we produce is not a strict terminological ontology, but rather a structure which has characteristics both of a terminological and of a prototype-based ontology. We thus give precedence to producing the structure with the highest probability over producing a structure which adheres to the rules for terminological ontologies. This behavior could be changed easily, by disallowing the insertion of the numeric nodes.

Using SVM classifiers, especially with non-linear kernels such as the RBF kernel used in our experiments, it is not transparent which feature or combination of features lead to an item being classified as a positive or negative case. For the example with 'beef', it seems obvious that cross-language information has been useful, since all three other languages involved can use the head matching heuristic. The translations are: 'Rindfleisch' (German), 'viande bovine' (French) and 'nötkött' (Swedish), where 'Fleisch', 'viande' and 'kött' all are translations of 'meat'. For the examples involving 'honey' and 'mineral_resource', the situation is less straight forward – there is no single feature we can point to for an explanation. We simply have to regard the outcome as a result of all available information working together to produce the classification.

The average branching factor of the ontology in Fig. 6.17 is much higher than two (two being the average branching factor of a *full binary tree*). We saw in Sect. 6.4.2 that the cohyponym classifier is solving a slightly easier problem, and also that its F-score on average is higher than the F-score for the hyperonym classifier. We get a greater number of relations with high confidence values from the cohyponym classifier than we get from the hyperonym classifier, which in turn leads to more cohyponymy relations being added, even when using identical k values for both relations. If we want to create a system that is truly unbiased as to which of the two relation types it prefers, such factors will have to be weighed in when setting the k values for the two relations.

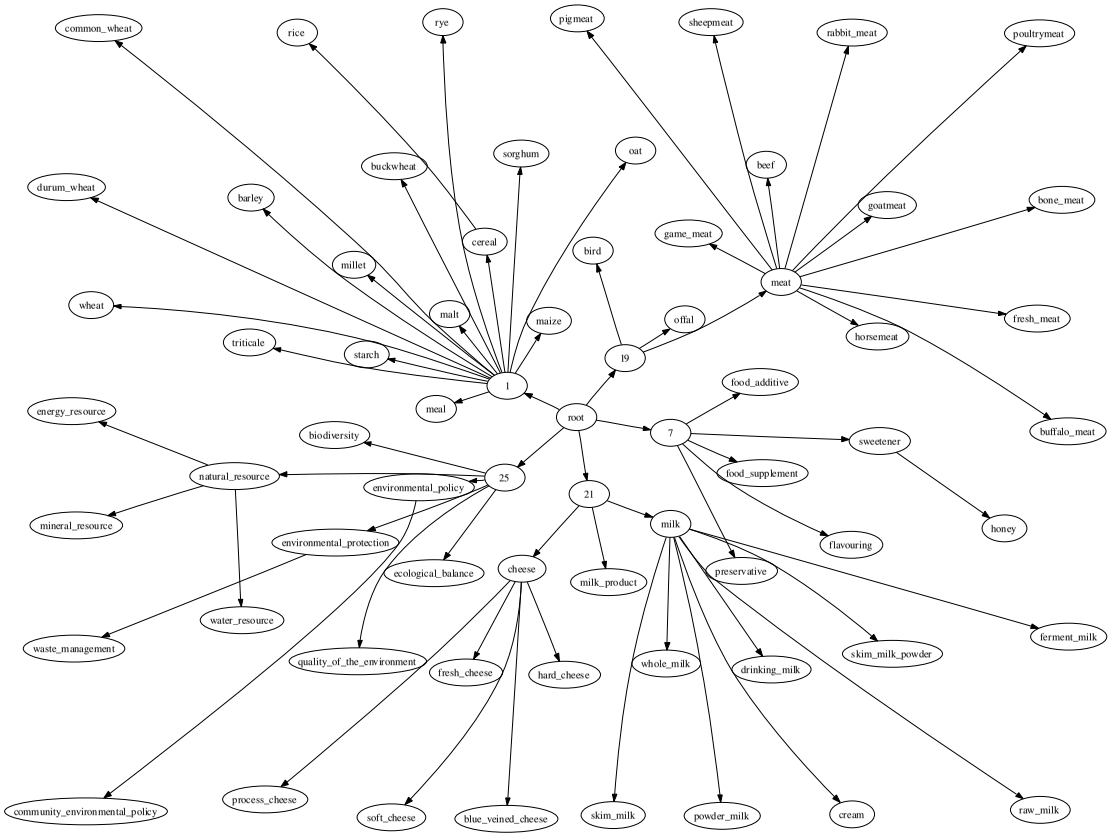


Figure 6.17: Excerpt of the ontology learned using the "All" data for partition 3, $k = 1.0$, $t = 1.0$.

6.6 How Good are the Results?

Maedche (2002) presents an experiment for evaluating how closely two ontologies built by humans resemble each other. He uses an existing ontology from the tourist domain, constructed by knowledge engineering experts, as a gold standard. The ontology contains 310 concepts, in addition to some top level concepts defining the basic structure. Four undergraduate students in industrial and business engineering were given 310 terms, corresponding to the 310 concepts, and asked to arrange them in a hierarchy. Each student received a total of six hours of training in ontology building before starting the task. The students' results are compared to the gold standard, but Maedche also performed a mutual comparison of the results within the student group. The task set for the students is very similar to the one we have evaluated in this thesis: given a set of terms, identify their relations when modeled in a hierarchical structure.

Maedche uses the measure we refer to as SC in Sect. 4.3 to compare the ontologies to one another. Because there are five ontologies to compare (four from the students plus the gold standard), we end up with a total of 20 comparisons. The SC values for these 20 comparisons range from 0.47 to 0.87, with an average value of 0.56 (the measure gives a value in the range [0,1]). Unfortunately we have not been able to perform any experiments with human subjects using our resources, so we are not sure what the corresponding figures under our conditions would be. It is not unreasonable to ascribe Maedche's figures some level of general applicability though; if our system manages to learn ontologies that give similarity figures that are in proximity of the ones in Maedche's study, we should take this as a favorable sign of the quality of the output of our system.

Here is where we run into problems with the SC measure, as discussed in Sect. 4.2.2. It is impossible to separate the values for lexical coverage from the values for precision and recall of the hierarchical relation (hyperonymy). Because we are using cross-validation in our experiments, we are only learning a small part of the ontology in each run, which means that all recall values will be very low using SC (typically below 0.01). Even if we disregard all other partitions of the ontology than the one we are using at the moment, we are only including a subset of the concepts (terms) within each partition in the learned ontology – namely those terms that occur at least once in the English texts (and the size of this subset will also vary with different values for the threshold t). Therefore, we cannot use the SC measure and expect meaningful results, in our case.

The CSC measure, however, provides exactly what we need: it calculates values of precision and recall taking *only* those concepts into consideration that occur in both ontologies. If we consider the CSC values at $t = 1.0$ (the standard threshold suggested in Snow et al., 2006) in Fig. 6.15, we get values in the range 0.38–0.42, depending on the dataset used for the experiment.

So, the highest result (0.42) at this threshold is still lower than the lowest result (0.47) in the human–human comparison and another step away from the average human–human result (0.56). This should not have us too worried, considering that the size of the gap is relatively modest. We should point out, though, that comparing evaluation results from the SC and CSC evaluation measures is not unproblematic, as demonstrated in Chap. 4.

If we look back to Sect. 1.2, where we discussed motivations for using ontology learning, we said that we do not expect the ontology learning process to be *fully* automated for all applications – a domain expert can always be consulted for post editing the results, if this should be deemed necessary. In this light, the results measured in the experiments in Sect. 6.5.2 are rather encouraging, though there is still plenty of qualitative and methodological improvement that can and should be made, before the system can start performing on a human level, in terms of accuracy.

7. Conclusions

We have presented a framework for cross-language ontology learning, focusing on providing a setting in which cross-language evidence (data) can be integrated and quantified. In this chapter, we will summarize our findings and contributions to the ontology learning field, discuss some issues raised in the thesis and also give prospects for future research directions.

7.1 Recapitulation and Contributions

Having established, in Chaps. 1–2, what ontology learning is, why it is useful and what we expect from approaching it from a cross-language perspective, we next discussed the need for reliable and robust evaluation metrics. In Chap. 4, we described two of the most commonly used metrics. We showed that they have certain problem areas, such as their non-applicability in given settings, as well as a degree of unpredictability and an exaggerated reliance on agreement for the top concept(s) for the compared ontologies. This, together with a lack of a standardized set of (cross-language) documents and an accompanying gold standard against which to evaluate, has prohibited comparisons of methodologies within the field. We therefore suggested a new cross-language document collection with an accompanying thesaurus for evaluating learned ontologies, as part of an effort to standardize evaluation (Chap. 3). Our first major contribution in this thesis is the development of a new evaluation measure, which remedies some of the previously mentioned problems regarding robustness and predictability. We can thus answer one of our initial questions, formulated in Sect. 1.3: *Can we improve or complement the evaluation measures used today?* The theoretical and experimental results presented in Chap. 4 show that the answer is yes.

Our next topic was translational equivalence between terms (Chap. 5). We investigated this as a matter in its own right, but also because we were interested in using the output of the translation system for providing equivalence links in our ontology learning system. Our contributions from these studies consist of a comparison of distributional similarity models and a statistical word alignment system on the task of bilingual dictionary extraction, as well as the introduction of an ensemble method for combining the two approaches. We also demonstrate a way of increasing the precision of the system by cross-checking the results, using a total of four languages for a task typically thought of as involving only two (Sect. 5.2.2).

The following chapter (Chap. 6) contains the major part of our contributions to the ontology learning field. We noted in Sect. 2.8, that our task largely consisted in finding a way of combining the various sources of information at our hands. Another question formulated in the introduction deals with this: *What are the chief sources of information for ontology learning and how can they be combined effectively?* The answer to this question is complex, but we showed in Sect. 6.4.1 that all features considered (described earlier in Sect. 6.2) in fact also contributed to solving the problem. We then demonstrated how to go about merging information across languages in Sect. 6.3. Finally, Sect. 6.4.2 showed how to join all information sources by training a support vector machine, exploiting knowledge drawn from all languages involved. Having said that, there are of course features we have left unconsidered, other ways of merging the cross-language data and alternative machine learning approaches one could use. We have focused on the state of the art, while striving towards a resource-lean, language independent approach, in cases where we were forced to choose between alternative methods.[1]

Our final question from the introduction has two parts; the first part asks *Can cross-language data teach us more than data from a single language for solving the ontology learning task?* We answer this question in the affirmative, both through showing increases in F-scores for the support vector machine classifiers in Sect. 6.4.2 and through the improvements in the ontology learning evaluation measures for the experiments in Sect. 6.5.2. The improvements for the latter are substantial. The second part of the question, *Can we use automatically extracted term equivalents to achieve this improvement?* is also answered positively, using the same argumentation. We demonstrated some interesting effects; perhaps the most striking was how the cross-language data increased the precision in the learned models.

7.2 Discussion and Outlook

In a sense, we are at the end of the road here; we have answered the questions we set out to investigate and indicated ways of improving ontology learning systems through the exploitation of cross-language data. In another sense, as discussed in Sect. 6.6, we still have a long way to go before our ontology learning system performs on a human level. We end this thesis by discussing ideas for further improving our system and we also consider widening the scope of the problem towards including other kinds of semantic relations.

Throughout the thesis, we are working with two corpora (and accompanying ontologies/thesauri). The JRC-ACQUIS corpus is parallel and meticulously controlled for its quality and content. The Wikipedia anatomy corpus on the other hand is comparable as opposed to parallel, and was constructed

[1] The methodology for merging the cross-language data is our own, so this part of the system does not belong to the state of the art in any established sense.

on the fly. Between them, they cover a large space of the type of corpora that might be available when wishing to learn an ontology from text. By putting the two corpora to use in different experiments, we have indicated what kinds of results we can expect from an ontology learning system, depending on the type of corpus that we feed the training component. We hope further to have shown that, in the absence of an "ideal", high-quality corpus such as the JRC-ACQUIS, useful results can still be produced from employing web-crawling techniques such as those we used for building the Wikipedia anatomy corpus.

In Sect. 2.3, we mentioned that we have ignored the issue of non-continuous strings of words, when performing term spotting. This has a possible effect of worsening the sparse data problem, because some occurrences of certain terms go unnoticed. If we were to allow non-continuous strings, this would on the other hand lower the precision for the term spotting task. We need an experimental investigation of the effects before we can say whether the system would benefit from allowing non-continuous strings or not. Words occurring "inside" the term are interesting to consider from a distributional aspect – how would they contribute to the co-occurrence profile of a term, when dealing with second order co-occurrences? We leave this as an open question, to be investigated in future experiments.

When considering the use of automatically generated term equivalence links, we mentioned in Sect. 6.3.1 that it would be possible to trade precision for recall when deciding which equivalents to merge and which not. In our current system, we are focusing on precision, but we cannot exclude the possibility that increasing the recall (at the cost of precision) would produce better results overall – again an area which would need further experimentation to get the needed clarification. Taking this line of reasoning further, it would be interesting to see the effects of using the results from the automatic translation based on a *comparable* corpus, where precision is even lower, for learning a prototype-based ontology (see Sect. 6.1).

We discussed possible repercussions of not using the transitive closure of the hyperonymy relation, when creating the training examples for the hyperonym classifier (Sects. 6.4 and 6.5.1). To see the rationale behind excluding the indirect relations, consider the Eurovoc excerpt in Fig. 7.1. The relation between 'animal product' and 'beef' is different from the relation between 'meat' and 'beef'. In text, we would expect 'beef' to be compared to other kinds of 'meat', but rarely to other kinds of 'animal products' like 'silk' or 'wool'. This will affect tendencies of the two term pairs to appear in Hearst-patterns together, and also their distributional patterns in general. On the other hand, training separate classifiers for recognizing hyperonymy relations for each different number of intervening concepts will most likely mean throwing away many commonalities, and instead create a sparse data problem. One possible compromise would be to build one classifier for direct hyperonyms, and one for indirect ones, whatever the number of intervening concepts, or, to allow a number of intervening concepts up to a certain threshold. It is again

difficult to predict what kind of effect this would have on the overall quality of the system output; we would need to look at results from contrastive ontology learning evaluations.

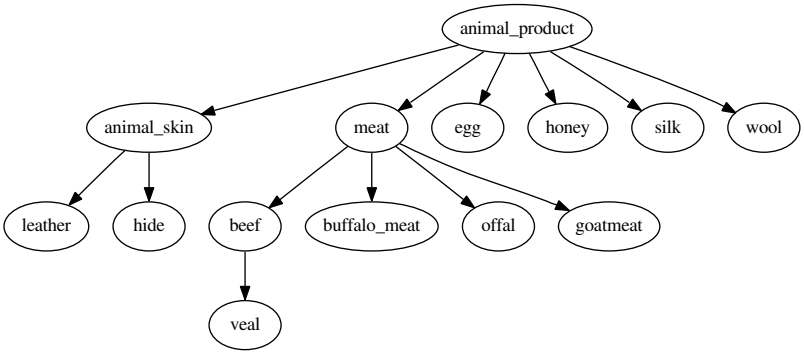


Figure 7.1: Eurovoc excerpt, concepts dealing with animal products. 'Leather' and 'hide' have been added to the original Eurovoc structure.

In Sect. 6.5.1, we described how we can steer the ontology learning system towards producing broader or narrower hierarchies (with larger or smaller average branching factors), by altering the value of a constant k. By setting the k value higher for either the hyperonymy or the cohyponymy relation, the system gets biased towards adding instances of one relation more frequently than the other. One could imagine finding an optimal k value for a particular dataset by generating ontologies using different parameterizations, and choosing the k value which generates the ontology with the highest probability. There is no guarantee that the most probable ontology (in this sense of 'probable') will also result in the best ontology by human standards, or by the standards of an ontology learning evaluation measure, but it is a possibility worth investigating.

We have made the simplifying assumption in this thesis of assuming a one-to-one correspondence between terms and concepts. Introducing the cross-language perspective, this assumption still holds, as long as we consider each language as an isolated system. If we were to add the synonymy relation to our two base relations hyperonymy and cohyponymy, we would have to abandon this simplification. Assume we allow the synonymy relation, and that we have currently learned the ontology in Fig. 7.1. Assume further that we also have good evidence for the relations in Fig. 7.2 (with an unspecified top node) and that we would like to add an instance of the synonymy relation between the terms 'animal skin' and 'skin'. Adding this instance means that a lot of implied relations will have to be considered, as we discussed in Sect. 6.5.1, e.g., that 'animal product' will be a hyperonym of 'skin' and 'fur', and that 'skin' and 'fur' will be cohyponyms of 'meat', 'egg', 'honey', etc. But we should also consider whether 'fur' and 'rawhide' can be merged via synonymy with any of the terms in matching positions in the "main" ontology ('hide'

and ‘rawhide’ would be good candidates for a merge in this example). This would result in the ontology in Fig. 7.3, where alternative terms (synonyms) are separated by a ‘|’.

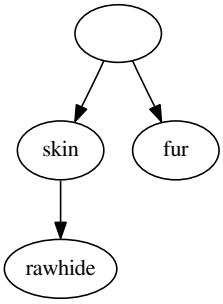


Figure 7.2: Partially learned ontology dealing with animal products, unspecified top node.

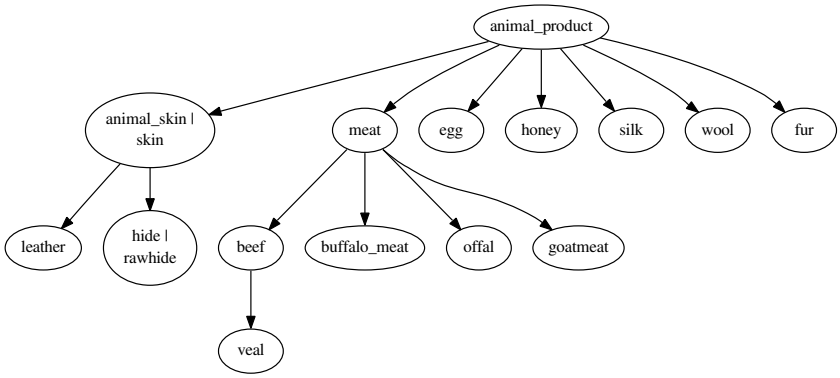


Figure 7.3: Ontology after merging (near) synonyms.

These further merges are not “implied” in the previous sense, but we would still like to consider their probabilities when deciding on the addition of a synonymy instance to the ontology. One approach would be to take positive evidence for further merges into account (such as for ‘hide’ and ‘rawhide’), while the rest can be treated as terms in their own right (as for ‘fur’, which has no real candidates for merging). Adding a cross-language dimension to the discussion, we would have to consider synonymy relations for all languages involved, which would add further complexity (e.g., two English terms appear to be synonyms, but their German translations do not – how should this be handled?). Making synonymy a third base relation, in addition to hyperonymy and cohyponymy, would thus pose some interesting challenges, especially when wanting to fit all relations into the same framework.

When discussing formal ontologies in the introductory chapter, we noted that they typically involve a number of other relations in addition to the ones

we consider in this thesis, e.g., meronymy and *related-to* (e.g., 'racket', 'ball' and 'net' are related-to each other in the sports domain). Once we have established the ontology skeleton via the base relations, we can add arbitrarily many networks of links on top, without them interfering with the basic hierarchical structure built up from hyperonymy and cohyponymy. There are no implications from meronymy or related-to relations that carry over to hyperonymy or cohyponymy; we can therefore treat the inclusion of additional layers of relations as independent problems, to be dealt with in a separate framework. On the other hand, having established the taxonomic backbone of the ontology via our base relations will most likely have a positive effect when trying to learn additional relations, since we then can exploit knowledge encoded in this hierarchical structure during the learning process.

The most obvious way of improving our system would be to make our 22 features produce more accurate results. 19 of our 22 features are based on distribution and frequencies, which means that they all most likely would benefit from using version 3.0 of the JRC-ACQUIS corpus, which is almost three times bigger than version 2.2, used in this thesis. The Hearst-pattern feature would probably also benefit from a bigger corpus, because related terms are more likely to appear in a Hearst-pattern, the more text we analyze. A cheap way of getting more co-occurrence data would be to use large general purpose corpora, in addition to our domain-specific ones. But this introduces problems with ambiguity, since we then can rely less on the "one sense per discourse" heuristic which we use for our domain-specific corpora.

In Sect. 6.4.2 we perform contrastive experiments to see the effects of merging data from different language pairs. We see some effects that indicate that typologically similar languages benefit the most from merging the data, but the results are not clear-cut. It would be interesting to incorporate languages that are typologically further removed from the others in future experiments (e.g., to include Finnish or any of the Slavic languages from the JRC-ACQUIS corpus and Eurovoc in the experiments). One could argue that merging the data should be simpler, the more similar the involved languages are. On the other hand, dissimilar languages could be argued to introduce a higher degree of "orthogonality" in the data, meaning that they would express *new* information, where similar languages merely would repeat what is already known.

Including part-of-speech information along the lines of Padó and Lapata (2007) would be another possibility for improving the distribution-based features. As discussed in Chap. 2, though, it remains to be proven that the syntactic models outperform the best word-based models.

If we had access to a high quality coreference resolution system, this would be another factor in dealing with the sparse data problem, allowing us to exchange pronouns for the non-pronominal nouns or noun phrases they refer to, in the running text. Hendrickx et al. (2008) show the reverse side of this; how distributional similarity can be used for improving coreference resolu-

tion, which means that there is an opportunity for creating a bootstrapping effect between the two fields.

Another way of further improving our system such as it stands today, would be to perform more extensive feature weighting and parameter optimization for the support vector machine classifiers. Such optimizations are time consuming when performed on large enough datasets, which is why we had to settle for using standard parameterizations for certain parts of our system. It is also possible that we could improve our system further by a more sophisticated handling of missing feature values in our data, perhaps by reduplicating or interpolating data from other languages. The latter primarily applies to experiments using the "MT" data in Chap. 6.

Finally, we feel that there is a lot of interesting work to be done in evaluating the effects of incorporating the semantic knowledge captured in a learned ontology in other NLP applications. We gave some examples in Sect. 2.1 of where learned or handcrafted ontologies brought improvements in the accuracy of a number of different NLP systems. We see some trends also in commercial settings towards incorporating ontological knowledge in information access-oriented companies such as Powerset,[2] Twine[3] and Hakia.[4] From a research perspective, it would of course be interesting to quantify the effects of incorporating ontological knowledge on the large scale (in terms of data and users) such commercializations allow. On a somewhat smaller scale, Nagypál (2007) has performed evaluations towards this end, that we feel would be worthwhile pursuing further.

To sum up, we have indicated many fields in NLP (machine translation is another such field of interest) where we suspect that ontology learning-like approaches have yet to demonstrate their full worth. In this we feel to be fully in line with dominating currents in NLP today, where, as noted in the introduction, more and more attention is given to handling semantics.

[2] www.powerset.com
[3] www.twine.com
[4] www.hakia.com

A. Example Output

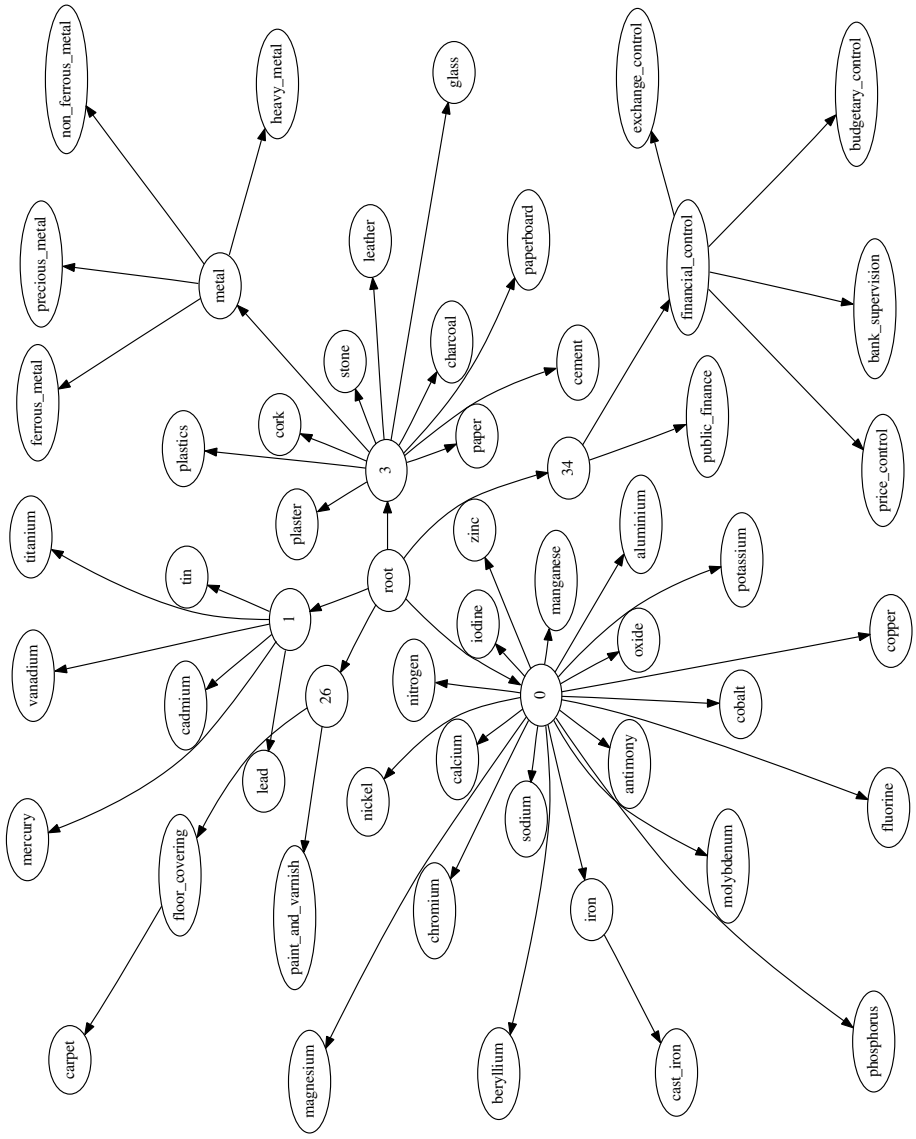


Figure A.1: Excerpt of the ontology learned using the "All" data for partition 4, $k = 1.0$, $t = 1.0$.

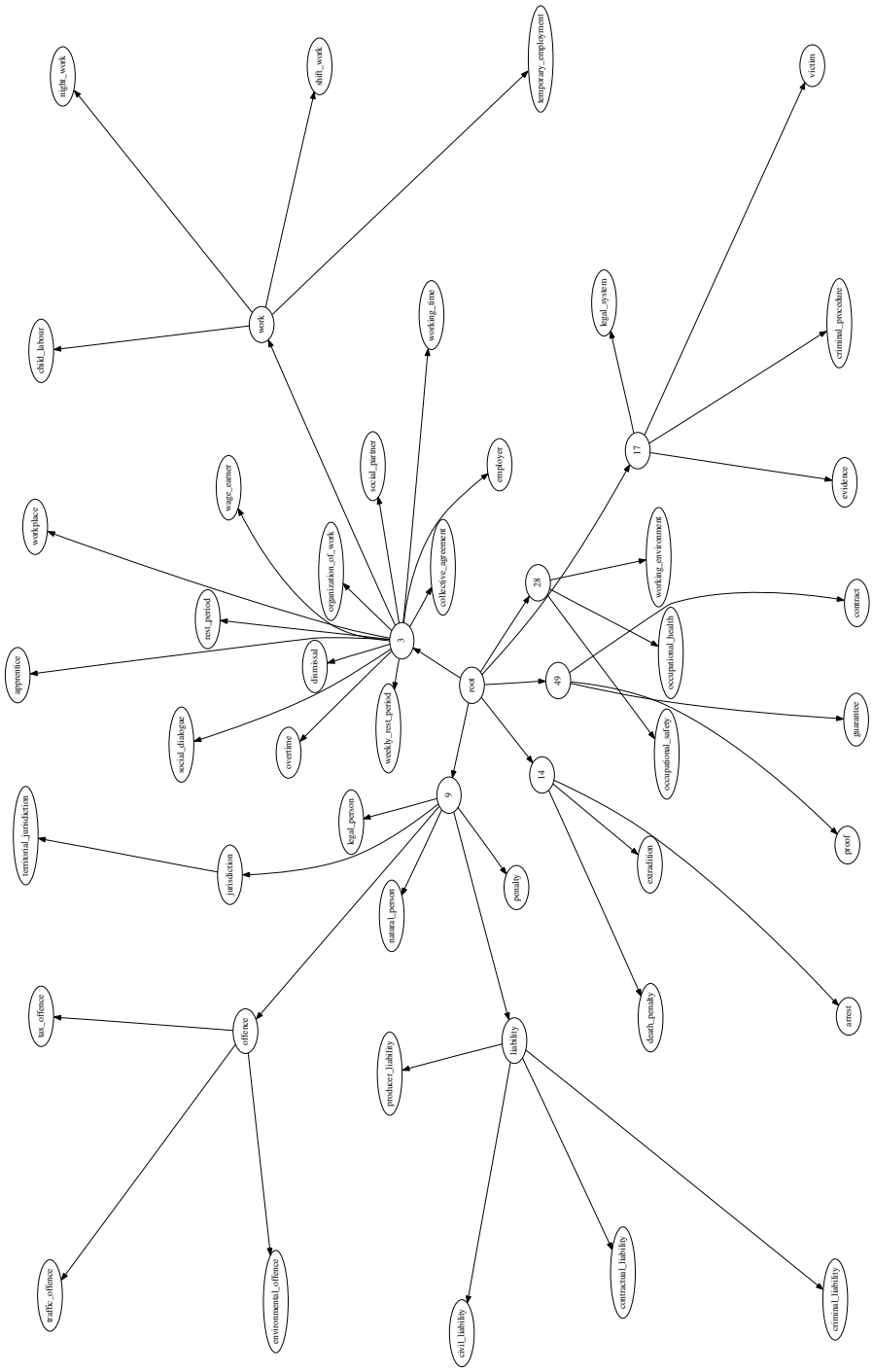


Figure A.2: Excerpt of the ontology learned using the "All" data for partition 6, $k = 1.0$, $t = 1.0$.

Bibliography

Towards a Shared Task for Multiword Expressions (MWE 2008), Marrakech, Morocco, 2008.

Akiko Aizawa and Kyo Kageura. A graph-based approach to the automatic generation of multilingual keyword clusters. In Didier Bourigault, editor, *Recent Advances in Computational Terminology*, chapter 1, pages 1–27. John Benjamins Publishing Company, Philadelphia, PA, USA, 2001.

William Alston. *Philosophy of Language*. Foundations of Philosophy. Prentice-Hall, Englewood Cliffs, NJ, USA, 1964.

David T. Barnard, Gwen Clarke, and Nicholas Duncan. Tree-to-tree correction for document trees: Technical report 95-372. Technical report, Dept. of Computing and Information Science, Queen's University, Kingston, ON, Canada, 1995.

Edward Eryl Bassett, editor. *Statistics: Problems and Solutions*. World Scientific Publishing, Singapore, 2000.

Tim Berners-Lee, Dieter Fensel, James A. Hendler, Henry Lieberman, and Wolfgang Wahlster, editors. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. The MIT Press, Cambridge, MA, USA, 2005.

Chris Biemann. Ontology learning from text – a survey of methods. *LDV-Forum*, 20(2):75–93, 2005.

Stephan Bloehdorn and Andreas Hotho. Boosting for text classification with semantic features. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Mining for and from the Semantic Web Workshop*, pages 70–87, 2004.

Stephan Bloehdorn, Philipp Cimiano, and Andreas Hotho. Learning ontologies to improve text clustering and classification. In Myra Spiliopoulou, Rudolf Kruse, Andreas Nürnberger, Christian Borgelt, and Wolfgang Gaul, editors, *From Data and Information Analysis to Knowledge Engineering: Proceedings of the 29th Annual Conference of the German Classification Society (GfKl 2005)*, volume 30 of *Studies in Classification, Data Analysis, and Knowledge Organization*, pages 334–341, Magdeburg, Germany, 2006. Springer-Verlag.

Stephan Bloehdorn, Philipp Cimiano, Alistair Duke, Peter Haase, Jörg Heizmann, Ian Thurlow, and Johanna Völker. Ontology-based question answering for digital

libraries. In *Proceedings of the 11th European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, pages 14–25, Budapest, Hungary, 2007.

Olivier Bodenreider, Anita Burgun, and Thomas Rindfleisch. Lexically-suggested hyponymic relations among medical terms and their representation in the UMLS. In *Proceedings of Terminology and Artificial Intelligence, TIA'2001*, pages 11–21, Nancy, France, 2001.

George Boolos and Richard Jeffrey. *Computability and Logic*. Cambridge University Press, Cambridge, UK, 3 edition, 1989.

Lars Borin. You'll take the high road and I'll take the low road: Using a third language to improve bilingual word alignment. In *Proceedings of the 18th International Conference on Computational Linguistics*, volume 1, pages 97–103. COLING, 2000.

Janez Brank, Dunja Mladenić, and Marko Grobelnik. Gold standard based ontology evaluation using instance assigment. In *Proceedings of the 4th International EON Workshop*, Edinburgh, United Kingdom, 2006.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.

Alexander Budanitsky and Graeme Hirst. Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

Paul Buitelaar, Philipp Cimiano, Stefania Racioppa, and Melanie Siegel. Ontology-based information extraction with SOBA. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)*, pages 2321–2324, Genoa, Italy, 2006.

Sharon A. Caraballo. *Automatic Construction of a Hypernym-Labeled Noun Hierarchy from Text*. PhD thesis, Brown University, Providence, RI, USA, 2001.

Sharon A. Caraballo and Eugene Charniak. Determining the specificity of nouns from text. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 63–70, College Park, MD, USA, 1999.

Johan Carlberger and Viggo Kann. Implementing an efficient part-of-speech tagger. *Software: Practice and Experience*, 29(9):815–832, 1999.

Marine Carpuat, Grace Ngai, Pascale Fung, and Kenneth Church. Creating a bilingual ontology: A corpus-based approach for aligning WordNet and HowNet. In *Proceedings of the 1st Global WordNet Conference*, Mysore, India, 2002.

M. Teresa Cabré Castellví, Rosa Estopà Bagot, and Jordi Vivaldi Palatresi. Automatic term detection: A review of current systems. In Didier Bourigault, editor, *Recent Advances in Computational Terminology*, chapter 3, pages 53–87. John Benjamins Publishing Company, Philadelphia, PA, USA, 2001.

B. Chandrasekaran, John R. Josephson, and V. Richard Benjamins. What are ontologies, and why do we need them? *IEEE Intelligent Systems*, 14(1):20–26, 1999.

Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Gary Chartrand, Grzegorz Kubicki, and Michelle Schulz. Graph similarity and distance in graphs. *Aequationes Mathematicae*, 55(1–2):129–145, 1998.

Philipp Cimiano. *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer-Verlag, New York, NY, USA, 2006.

Philipp Cimiano, Andreas Hotho, and Steffen Staab. Comparing conceptual, partitional and agglomerative clustering for learning taxonomies from text. In *Proceedings of the European Conference on Artificial Intelligence (ECAI'04)*, pages 435–439, Valencia, Spain, 2004. IOS Press.

Philipp Cimiano, Andreas Hotho, and Steffen Staab. Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence Research*, 24:305–339, 2005a.

Philipp Cimiano, Aleksander Pivk, Lars Schmidt-Thieme, and Steffen Staab. Learning taxonomic relations from heterogeneous sources of evidence. In Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini, editors, *Ontology Learning from Text: Methods, Evaluation and Applications*. IOS Press, Amsterdam, Holland, 2005b.

Thomas Connolly, Carolyn Begg, and Anne Strachan. *Database Systems: A Practical Approach to Design, Implementation and Management*. Addison-Wesley, Harlow, England, 2 edition, 1998.

Anthony Paul Cowie, editor. *Phraseology*. Oxford Studies in Lexicography and Lexicology. Clarendon Press, 1998.

David Allan Cruse. *Lexical Semantics*. Cambridge Textbooks in Linguistics. Cambridge University Press, Cambridge, UK, 1986.

Pernilla Danielsson. Automatic extraction of meaningful units from corpora. *International Journal of Corpus Linguistics*, 8(1):109–127, 2003.

Scott Deerwester, Susan Dumais, Thomas Landauer, George Furnas, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

Klaas Dellschaft and Steffen Staab. On how to perform a gold standard based evaluation of ontology learning. In *5th International Semantic Web Conference*, Athens, GA, USA, 2006.

Helge Dyvik. Translations as a semantic knowledge source. In *Proceedings of the Second Baltic Conference on Human Language Technologies*, Tallinn, Estonia, 2005.

Hervé Déjean, Éric Gaussier, and Fatia Sadat. Bilingual terminology extraction: an approach based on a multilingual thesaurus applicable to comparable corpora. In *Proceedings of COLING*, Taipei, Taiwan, 2002.

Hervé Déjean, Éric Gaussier, Jean-Michel Renders, and Fatia Sadat. Automatic processing of multilingual medical terminology: Applications to thesaurus enrichment and cross-language information retrieval. *Artificial Intelligence in Medicine*, 33 (2):111–124, 2005.

Andreas Faatz and Ralf Steinmetz. Precision and recall for ontology enrichment. In *Proceedings of the ECAI-2004 Workshop on Ontology Learning and Population*, Sevilla, Spain, 2004.

Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, Massachusetts, USA, 1998.

Pascale Fung and Kathleen McKeown. Finding terminology translations from non-parallel corpora. In *The 5th Annual Workshop on Very Large Corpora*, pages 192–202, Hong Kong, 1997.

Roxana Girju, Adriana Badulescu, and Dan Moldovan. Automatic discovery of part-whole relations. *Computational Linguistics*, 32(1):83–135, 2006.

Cliff Goddard. Universal units in the lexicon. In Martin Haspelmath, Ekkehard König, Wulf Oesterreicher, and Wolfgang Raible, editors, *Language Typology and Language Universals*, volume 2, pages 1178–1190. Walter de Gruyter, Berlin, Germany / New York, NY, USA, 2001.

Gene H. Golub and Charles F. van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, USA, 3 edition, 1996.

Nelson Goodman. *Ways of Worldmaking*. Hackett Pub Co Inc, Indianapolis, IN, USA, 1978.

Gregory Grefenstette. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Boston, MA, USA, 1994.

Leif Grönqvist. *Exploring Latent Semantic Vector Models Enriched With N-Grams*. PhD thesis, Växjö University, Växjö, Sweden, 2006.

Zellig Harris. Distributional structure. *Word*, 10(2–3):775–793, 1954.

Marti Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics*, Nantes, France, 1992.

Marti Hearst and Hinrich Schütze. Customizing a lexicon to better suit a computational task. In *Proceedings of the ACL SIGLEX Workshop*, Columbus, Ohio, USA, 1993.

Iris Hendrickx, Véronique Hoste, and Walter Daelemans. Semantic and syntactic features for anaphora resolution for Dutch. In *Proceedings of the CICLing-2008 conference*, volume 4919 of *Lecture Notes in Computer Science*, pages 351–361. Springer Verlag, Berlin, Germany, 2008.

Hans Hjelm. Identifying cross language term equivalents using statistical machine translation and distributional association measures. In *Proceedings of Nodalida 2007, the 16th Nordic Conference of Computational Linguistics*, Tartu, Estonia, 2007.

Hans Hjelm and Paul Buitelaar. Multilingual evidence improves clustering-based taxonomy extraction. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI 2008)*, Patras, Greece, 2008. IOS Press.

Hans Hjelm and Christoph Schwarz. LiSa - morphological analysis for information retrieval. In Stefan Werner, editor, *Proceedings of the 15th NODALIDA conference, Joensuu 2005*, volume 1 of *University of Joensuu electronic publications in linguistics and language technology*. NoDaLiDa, Ling@JoY, 2006.

Jon Holmlund, Magnus Sahlgren, and Jussi Karlgren. Creating bilingual lexica using reference wordlists for alignment of monolingual semantic vector spaces. In *Proceedings of the 15th Nordic Conference on Computational Linguistics, NoDaLiDa*, Joensuu, Finland, 2005.

Andreas Hotho, Steffen Staab, and Alexander Maedche. Ontology-based text clustering. In *Proc. of IJCAI 2001*, Seattle, WA, USA, 2001.

Lucja Iwanska, Naveen Mata, and Kellyn Kruger. Fully automatic acquisition of taxonomic knowledge from large corpora of texts. In Lucja Iwanska and Stuart Shapiro, editors, *Natural Language Processing and Knowledge Representation*, chapter 10. The MIT Press, London, England, 2000.

Christian Jacquemin. *Spotting and Discovering Terms through Natural Language Processing*. The MIT Press, Cambridge, MA, USA, 2001.

J. Jin, B.K. Sarker, V.C. Bhavsar, H. Boley, and L. Yang. Towards a weighted-tree similarity algorithm for RNA secondary structure comparison. In *Proceedings of the 8th International Conference on High Performance Computing in Asia Pacific Region (HPC Asia 2005)*, pages 639–644, Beijing, China, 2005.

Pantti Kanerva, Jan Kristoferson, and Anders Holst. Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, 2000.

Fred Karlsson. SWETWOL: A comprehensive morphological analyzer for Swedish. *Nordic Journal of Linguistics*, 15(1):1–45, 1992.

Károly Kerényi. *Dionysos: Archetypal Image of Indestructible Life*. Princeton University Press, 1976.

Adam Kilgarriff. Googleology is bad science. *Computational Linguistics*, 33(1): 147–151, 2007.

Peter Koch. Lexical typology from a cognitive and linguistic point of view. In Martin Haspelmath, Ekkehard König, Wulf Oesterreicher, and Wolfgang Raible, editors, *Language Typology and Language Universals*, volume 2, pages 1142–1178. Walter de Gruyter, Berlin, Germany / New York, NY, USA, 2001.

Olivier Kraif. From translational data to contrastive knowledge. *International Journal of Corpus Linguistics*, 8(1):1–29, 2003.

Thomas Landauer and Susan Dumais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

Hang Li. Word clustering and disambiguation based on co-occurrence data. *Natural Language Engineering*, 8(1):25–42, 2002.

Dekang Lin. Automatic retrieval and clustering of similar words. In *COLING-ACL98*, Montreal, QC, Canada, 1998.

Krister Lindén and Jussi Piitulainen. Discovering synonyms and other related words. In Sophia Ananadiou and Pierre Zweigenbaum, editors, *COLING 2004 CompuTerm 2004: 3rd International Workshop on Computational Terminology*, pages 63–70, Geneva, Switzerland, 2004. COLING.

Carl Linnaeus. *Systema Naturae*. Trustees of the British Museum, London, England, 1939.

Zhenyu Liu and Wesley W. Chu. Knowledge-based query expansion to support scenario-specific retrieval of medical free text. *Information Retrieval*, 10(2):173–202, 2007.

Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, and Computers*, 28(2):203–208, 1996.

Alexander Maedche. *Ontology Learning for the Semantic Web*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

Alexander Maedche, Viktor Pekar, and Steffen Staab. Ontology learning part one – on discovering taxonomic relations from the web. In Ning Zhong, Jiming Liu, and Yiyu Yao, editors, *Web Intelligence*, chapter 14. Springer Verlag, New York, NY, USA, 2003.

Juha Makkonen, Helena Ahonen-Myka, and Marko Salmenkivi. Simple semantics in topic detection and tracking. *Information Retrieval*, 7(3–4):347–368, 2004.

Véronique Malaisé, Pierre Zweigenbaum, and Bruno Bachimont. Mining defining contexts to help structuring differential ontologies. *Terminology*, 11(1):21–53, 2005.

Inderjeet Mani, Ken Samuel, Kris Concepcion, and David Vogel. Automatically inducing ontologies from corpora. In *Proceedings of CompuTerm 2004: 3rd International Workshop on Computational Terminology*, Geneva, Switzerland, 2004. COLING.

Christopher Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, MA, USA, 1999.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

Tony McEnery, Richard Xiao, and Yukio Tono. *Corpus-Based Language Studies*. Routledge, London, England and New York, NY, USA, 2006.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

Dan Melamed. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249, 2000.

Magnus Merkel. *Understanding and Enhancing Translation by Parallel Text Processing*. PhD thesis, Linköping University, Linköping, Sweden, 1999.

George Miller and Florentina Hristea. WordNet nouns: Classes and instances. *Computational Linguistics*, 32(1):1–3, 2006.

Gábor Nagypál. *Possibly Imperfect Ontologies for Effective Information Retrieval*. PhD thesis, Universität Karlsruhe (TH), Karlsruhe, Germany, 2007.

Roberto Navigli and Paola Velardi. Learning domain ontologies from document warehouses and dedicated web sites. *Computational Linguistics*, 30(2):151–179, 2004.

Goran Nenadic, Irena Spasic, and Sophia Ananiandou. Mining term similarities from corpora. *Terminology*, 10(1):55–80, 2004.

Eugene Albert Nida. *Componential Analysis of Meaning: an Introduction to Semantic Structures*. Mouton, The Hague, Netherlands, 1975.

Sergei Nirenburg and Victor Raskin. *Ontological Semantics*. Language, Speech and Communication. The MIT Press, Cambridge, Massachusetts, USA, 2004.

Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

Norihiro Ogata and Nigel Collier. Ontology express: Statistical and non-monotonic learning of domain ontologies from text. In *Proceedings of the ECAI-2004 Workshop on Ontology Learning and Population*, Sevilla, Spain, 2004.

Charles Kay Ogden and Ivor Armstrong Richards. *The Meaning of Meaning*. Harcourt, Brace, and World, New York, NY, USA, 8th edition 1946 edition, 1923.

Ulrike Oster. Classifying domain-specific intraterm relations. *Terminology*, 12(1): 1–17, 2006.

Henrik Oxhammar. Evaluating feature selection techniques on semantic likeness. In *Proceedings of the Workshop on Semantic Content Acquisition and Representation (SCAR) 2007*, Tartu, Estonia, 2007.

Sebastian Padó and Mirella Lapata. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199, 2007.

Patrick Pantel and Dekang Lin. Discovering word senses from text. In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining (KDD-02)*, pages 613–619, Edmonton, AB, Canada, 2002.

Patrick Pantel and Deepak Ravichandran. Automatically labeling semantic classes. In *Proceedings of Human Language Technology/North American chapter of the Association for Computational Linguistics (HLT/NAACL-04)*, pages 321–328, Boston, MA, USA, 2004.

Fernando Pereira, Naftali Tishby, and Lillian Lee. Distributional clustering of english words. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Columbus, OH, USA, 1993.

Willard Van Orman Quine. *Word and Object*. The MIT Press, 1960.

Willard Van Orman Quine. *Ontological Relativity and Other Essays*. Columbia University Press, 1969.

Reinhard Rapp. Identifying word translations in non-parallel texts. In *Proceedings of the 33rd Meeting of the Association for Computational Linguistics*, pages 320–322, Cambridge, MA, USA, 1995.

Reinhard Rapp. Automatic identification of word translations from unrelated English and German corpora. In *Proceedings of the 37th Annual Meeting of the ACL (ACL'99)*, College Park, MD, USA, 1999.

Philip Resnik. Exploiting hidden meanings: Using bilingual texts for monolingual annotation. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, number 2945 in Lecture Notes in Computer Science, pages 283–299. Springer-Verlag, 2004.

Philip Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11(1), 1998.

Philip Resnik and Noah A. Smith. The web as a parallel corpus. *Computational Linguistics*, 29(3):349–380, 2003.

António Ribeiro, Gabriel Pereira Lopes, and João Mexia. Extracting equivalents from aligned parallel texts: Comparison of measures of similarity. In M. C. Monard and J. S. Sichman, editors, *Advances in Artificial Intelligence: International Joint Conference, 7th Ibero-American Conference on AI, 15th Brazilian Symposium on AI, IBERAMIA-SBIA 2000, Atibaia, SP, Brazil, November 2000. Proceedings*, Lecture Notes in Computer Science, pages 339–349. Springer-Verlag, Berlin Heidelberg, Germany, 2000.

Fabio Rinaldi, Gerold Schneider, Kaarel Kaljurand, Michael Hess, and Martin Romacker. An environment for relation mining over richly annotated corpora: the case of GENIA. In *Second International Symposium on Semantic Mining in Biomedicine (SMBM)*, Jena, Germany, 2006.

Brian Roark and Eugene Charniak. Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 1110–1116, Montreal, QC, Canada, 1998.

Magnus Rosell. *Clustering in Swedish – The Impact of some Properties of the Swedish Language on Document Clustering and an Evaluation Method*. Licentiate thesis, School of Computer Science and Communication, Royal Institute of Technology, Stockholm, Sweden, 2005.

Gerda Ruge and Christoph Schwarz. Term associations and computational linguistics. *International Classification*, 18(1):19–25, 1991.

Bertrand Russell. Introduction. In Ludwig Wittgenstein, *Tractatus logico-philosophicus*. Routledge & Kegan Paul, London, England, 1961.

Sara Rydin. Building a hyponymy lexicon with hierarchical structure. In *Proceedings of the Workshop of the ACL Special Interest Group on the Lexicon (SIGLEX)*, pages 26–33, 2002.

Pum-Mo Ryu and Key-Sun Choi. Determining the specificity of terms based on information theoretic measures. In Sophia Ananadiou and Pierre Zweigenbaum, editors, *COLING 2004 CompuTerm 2004: 3rd International Workshop on Computational Terminology*, pages 87–90, Geneva, Switzerland, 2004. COLING.

Pum-Mo Ryu and Key-Sun Choi. Taxonomy learning using term specificity and similarity. In *Proceedings from the Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge (with Coling/ACL 2006)*, pages 41 – 48, Sydney, Australia, 2006.

Juan Sager. *Language Engineering and Translation: Consequences of automation*. John Benjamins Publishing Company, Amsterdam, 1994.

Magnus Sahlgren. Automatic bilingual lexicon acquisition using random indexing of aligned bilingual data. In *Proceedings of the fourth international conference on Language Resources and Evaluation, LREC 2004*, pages 1289–1292, Lisbon, Portugal, 2004.

Magnus Sahlgren. *The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-Dimensional Vector Spaces*. PhD thesis, Stockholm University, Stockholm, Sweden, 2006.

Magnus Sahlgren and Jussi Karlgren. Automatic bilingual lexicon acquisition using random indexing of parallel corpora. *Natural Language Engineering*, 11(3):327–341, 2005.

Magnus Sahlgren, Anders Holst, and Pantti Kanerva. Permutations as a means to encode order in word space. In *Proceedings of the 30th Annual Meeting of the Cognitive Science Society (CogSci'08)*, Washington, D.C., USA, 2008.

Mark Sanderson and Bruce Croft. Deriving concept hierarchies from text. In *Proceedings of the 22nd ACM Conference of the Special Interest Group in Information Retrieval*, pages 206–213, Berkeley, CA, USA, 1999.

Eric SanJuan, James Dowdall, Fidelia Ibekwe-SanJuan, and Fabio Rinaldi. A symbolic approach to automatic multiword term structuring. *Journal of Computer Speech and Language*, 19(4):524–542, 2005.

Frank Schilder, Andrew McCulloh, Bridget Thomson McInnes, and Alex Zhou. TLR at DUC: Tree similarity. In *Proceedings of the Document Understanding Conference (DUC-2005)*, Vancouver, BC, Canada, 2005.

Christoph Schwarz. Automatic syntactic analysis of free text. *Journal of the American Society for Information Science*, 41(6):408–417, 1990.

Bernhard Schölkopf. SVMs – a practical consequence of learning theory. *IEEE Intelligent Systems*, 13(4):18–21, 1998.

Hinrich Schütze. Dimensions of meaning. In *Proceedings of Supercomputing '92*, pages 787–796, Los Alamitos, CA, USA, 1992.

Hinrich Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123, 1998.

Hinrich Schütze, David A. Hull, and Jan O. Pedersen. A comparison of classifiers and document representations for the routing problem. In *Proceedings of SIGIR '95*, pages 229–237, Seattle, WA, USA, 1995.

Francesco Sclano and Paola Velardi. TermExtractor: a web application to learn the shared terminology of emergent web communities. In *Proceedings of the 3rd International Conference on Interoperability for Enterprise Software and Applications (I-ESA 2007)*, Funchal, Portugal, 2007.

Michel Simard. Text-translation alignment: Three languages are better than two. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 2–11, 1999.

John Sinclair. The empty lexicon. *International Journal of Corpus Linguistics*, 1(1), 1996.

Jonas Sjöbergh and Viggo Kann. Finding the correct interpretation of Swedish compounds: a statistical approach. In *Proceedings of LREC-2004*, pages 899–902, Lisbon, Portugal, 2004.

Frank Smadja, Kathleen R. McKeown, and Vasileios Hatzivassiloglou. Translating collocations for bilingual lexicons: A statistical approach. *Computational Linguistics*, 22(1):1–38, 1996.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1297–1304. MIT Press, Cambridge, MA, 2005.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Semantic taxonomy induction from heterogeneous evidence. In *Proceedings of COLING/ACL 2006*, Sydney, Australia, 2006.

Harold Somers. Knowledge extraction from bilingual corpora. In Maria Teresa Pazienza, editor, *Information Extraction*, number 1714 in LNAI, pages 120–133. Springer-Verlag, Berlin, Germany, 1999.

Min Song, Il-Yeol Song, Xiaohua Hu, and Robert B. Allen. Integration of association rules and ontologies for semantic query expansion. *Data & Knowledge Engineering*, 63(1):63–75, 2007.

John Sowa. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Thomson Learning, New York, NY, USA, 1999.

Karen Spärck-Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufis, and Dániel Varga. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)*, Genoa, Italy, 2006.

Alexander Strehl, Joydeep Ghosh, and Raymond Mooney. Impact of similarity measures on web-page clustering. In *Proceedings of the AAAI Workshop on AI for Web Search (AAAI 2000)*, pages 58–64, Austin, TX, USA, 2000.

Kuo-Chung Tai. The tree-to-tree correction problem. *Journal of the Association for Computing Machinery*, 26(3):422–433, 1979.

David M. J. Tax, Martin van Breukelen, Robert P. W. Duin, and Josef Kittler. Combining multiple classifiers by averaging or by multiplying. *Pattern Recognition*, 33(9):1475 – 1485, 2000.

Terminologicentrum TNC. Fackspråk eller fikonspråk? Om naturvetares språk. Terminologicentrum TNC, Solna, Sweden, 2004.

Jörg Tiedemann. *Recycling Translations: Extraction of Lexical Data from Parallel Corpora and their Application in Natural Language Processing*. PhD thesis, Uppsala University, Uppsala, Sweden, 2003.

Jörg Tiedemann. Optimization of word alignment clues. *Natural Language Engineering*, 11(3):279–293, 2005.

Lonneke van der Plas and Jörg Tiedemann. Finding synonyms using automatic word alignment and measures of distributional similarity. In *Proceedings of ACL-COLING 2006*, Sydney, Australia, 2006.

Martin Volk and Paul Buitelaar. A systematic evaluation of concept-based cross-language information retrieval in the medical domain. In *Proc. of 3rd Dutch-Belgian Information Retrieval Workshop*, Leuven, Holland, 2002.

Martin Volk, Anna-Katharina Pantli, and Anita Mirjam Malka. The length factor in automatic bilingual terminology extraction. In *Proceedings of 6th International Conference on Terminology and Knowledge Engineering*, Nancy, France, 2002.

Julie Weeds and David Weir. Co-occurrence retrieval: A flexible framework for lexical distributional similarity. *Computational Linguistics*, 31(4):439–475, 2005.

Julie Weeds, James Downdall, Gerold Schneider, and David Weir. Using distributional similarity to organise biomedical terminology. *Terminology*, 11(1):107–141, 2005.

Dominic Widdows. A mathematical model for context and word-meaning. In *Proceedings of the Fourth International and Interdisciplinary Conference on Modeling and Using Context*, Stanford, CA, USA, 2003.

Yorick Wilks. The "fodor"-FODOR fallacy bites back. In Pierette Bouillon and Federica Busa, editors, *The Language of Word Meaning*. Cambridge University Press, Cambridge, UK, 1999.

Wilson Wong and Wei Liu. Tree-traversing ant algorithm for term clustering based on featureless similarities. *Data Mining and Knowledge Discovery*, 15(3):349–381, 2007.

Dekai Wu and Pascale Fung. Inversion transduction grammar constraints for mining parallel sentences from quasi-comparable corpora. In *Proceedings of the 2nd International Joint Conference on Natural Language Proceedings (IJCNLP 05)*, Jeju Island, South Korea, 2005.

David Yarowsky, Grace Ngai, and Richard Wicentowski. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of HLT 2001, First International Conference on Human Language Technology Research*, 2001.