Elements of Knowledge-free and Unsupervised Lexical Acquisition

Der Fakultät für Mathematik und Informatik
der Universität Leipzig
eingereichte

DISSERTATION

zur Erlangung des akademischen Grades

DOCTOR rerum naturalium
(Dr. rer. nat.)

im Fachgebiet

Informatik

vorgelegt

von Dipl. Inf. Stefan Bordag

geboren am 3. Juni 1978 in Leipzig

Leipzig, den 8. März 2007

Contents

List of Tables

List of Figures

Preface

Goals

> Einige Sprachwissenschaftler haben sich die Ausarbeitung einer Beschreibungsmethode als Ideal aufgestellt, die den Sinn der bedeutungstragenden Einheiten nicht ins Spiel bringt. [...] Man käme auf diese Weise zu einer vollständigen Analyse der Sprache und könnte so eine Grammatik und sogar ein Lexikon zusammenstellen, dem lediglich die Definitionen fehlen würden, wie sie unsere Wörterbücher geben. In Wirklichkeit hat es sich anscheinend noch kein Sprachwissenschaftler einfallen lassen, eine Sprache zu analysieren und zu beschreiben, die er überhaupt nicht verstand. Aller Wahrscheinlichkeit nach würde ein solches Unternehmen zu seiner Durchführung einen Aufwand von Zeit und Kraft erfordern, der selbst die abgeschreckt hat, die diese Methode für die einzige theoretisch annehmbare halten. [...]
>
> Several linguists have stated the ideal to produce a description method [of language] that excludes the meaning of meaningful [language] units. [...] This would result in a complete description of the language and it would be possible to compile a grammar and a lexicon that would lack only the definitions [of the words] in the way they are present in our current lexicons. In reality no linguist has yet come to the idea of analyzing and describing a language he does not know at all in such a manner. Such an undertaking would by all accounts require an expense of time and energy that has deterred even those who consider this approach as the only one theoretically acceptable. [...] (free translation by the author of this work) (Martinet, 1969)

The quotation from André Martinet's book, written at the time when computers just started to be become publicly available, can be taken as a motto of this work, except for the conclusion Martinet drew. When he wrote those lines, it was hardly imaginable that computers could be used to handle the tedious parts of analyzing a specific language without understanding it. Despite this fact, it is obvious that Martinet believed that it should be possible to take one specific language (in the form of a collection of samples), analyze it using statistical methods only, and arrive at a complete and precise description of the structure of that particular language. A further assumption is that the fact that the sign represents meaning should be ignored for the purposes of such a task. This coincides with the fact that all current natural language processing algorithms operate exactly in this manner. They disregard the link between the sign and the world and operate solely on the structure of the language.

According to Martinet, once the prohibitively tedious part of the analysis can be relenquished (for example) to computers, the rest of the analysis should be very simple. A mere measurement of word co-occurrences, for example, should lead to a language description. However, precisely this part proves to be the difficult one, because it presupposes the 'proper' kinds of co-occurrence measurements, based on an encompassing understanding of the language structure. A complex system of measurements, instructions and operations, commonly known as an algorithm, is needed that would perform the automated process of language analysis which would yield a 'complete description of the language'.

The construction of such a system should account for all relevant aspects. For example, it could be guided by linguistic hypotheses, while at the same time providing empirical evidence for or against them. Typology could provide information about what is generally possible in languages, as well as what kind of universals actually hold true for all languages or at minimum groups of languages. Psycholinguistics could provide further hypotheses for principles of mechanisms that operate in the human brain in order to better understand the possible limitations to the structure of language. However, as airplanes which can fly without flapping their wings demonstrate, not the exact reconstruction of the bird's wings is important. Instead, it is essential to understand the principles of aerodynamics to enable the construction of airplanes. Yet these principles can be understood from observations of birds.

The aim of this work is to formally describe the most general principles operating in language that currently are reliably observable, broken down to the most essential ones. These formal descriptions are open to changes and additions, allowing for a later integration of new language-independent mechanisms, but also for improvement of previously introduced ones and the discarding of formalizations which were proved wrong or inappropriate. In order to support the formal description, an additional and more practical aim of this work is to examine several specific areas such as the compuation of word similarity, ambiguity or morphology.

The remaining parts of this work focus on specific language independent algorithms[1], which are based solely on the described mechanisms and well known linguistic hypotheses. These algorithms extract certain structural elements of language in a fully automatic manner. Evaluations of the algorithms are also provided, but the focus of this works remains on the breadth so as to cover a wider variety of language phenomena and levels to accommodate the formalizations. The breadth-first approach requires sacrifices at several points with respect to certain questions

[1] In the sense of producing good results for a large variety of natural languages, not just for one.

which are identified and discussed, but remain unanswered. The breadth approach allows to find new possible interactions between the examined algorithms, some of which are explored in greater detail in the last chapter.

Overview

Since computers in general, and algorithms in particular, are key to dissolving the tedious part of language analysis, algorithms constitute the starting point of the first chapter. Based on similarities and differences between various algorithms, a line is drawn to more traditional theoretical considerations. A closer look at the essential methods employed for lexical acquisition and other algorithms demonstrates that with their reliance on context they resemble de Saussurean structuralism and Harris' distributional hypothesis. Several issues, such as the distinction between language structure and meaning or the relationship between paradigmatic and syntagmatic relations, are discussed. The topic of proper sampling for empirical observations is also briefly addressed. The later paragraphs of this chapter review the premises on which this work is founded.

The second chapter introduces a formalization of several observable language properties. Building on basic notions, including the existence of language levels, composition and abstraction principles, and context, syntagmatic and paradigmatic relations are formalized. The discrepancy between possible and experienced utterances[2] is bridged by means of statistically significant observations and generalizations which can be derived from them. It is shown, how the definitions of syntagmatic and paradigmatic relations can be used to define syntactic (e.g. grammatical gender) and semantic (e.g. animacy) attributes. These can then be employed to describe complex structural dependencies and compliances, as well as coherence. An example at the end of the chapter demonstrates how the defined coherence could be used for automatic abstracting.

In the third and following chapters, it is demonstrated how existing algorithms fit into the formalizations introduced in the second chapter and how new algorithms can be derived by applying the formalizations.

Chapter three first deals with measures of co-occurrences of words within sentences. A review gives insights into the various state of the art methods for extracting information about syntagmatic relations between words (for example linguistic collocations) using co-occurrence measures. Further, the various comparison-based methods of computing similar word pairs, or paradigmatic relations in general, are evaluated. Finally, possibilities are shown for expanding the extraction to finer

[2] The critiqes of corpus-based approaches (see Pinker (1994) for arguments against and Sampson (2005) for an answer and generally an overview of the 'language instinct' debate) argue that there are always utterances which never occur in a corpus but which still can be produced and understood by human beings. They question the theoretical ability of learning algorithms to correctly analyze such sentences.

grained relations such as synonymy, antonymy and hyperonymy, which are addressed and discussed in detail in Chapter 6.

The fourth chapter addresses word ambiguity. An observation is made that the significant word usages, expressed by co-occurring words as computed in Chapter 3, tend to cluster according to the various meanings of the word. Based on that, an algorithm (Bordag, 2006b) is designed which finds lexical ambiguities and consequently splits the set of co-occurring words. An evaluation measure is introduced which additionally measures the impact of other factors, including syntactic ambiguity, frequency and over-representation.

The goal of Chapter 5 is to create an unsupervised morpheme segmentation algorithm. First, various algorithms designed to extract morphological information in a more or less unsupervised and automatic manner are reviewed and compared. According to the formalizations given in Chapter 2, they can be classified into several classes according to how much and which information they utilize. A new algorithm is introduced and evaluated, additionally to the evaluations of the MorphoChallenge 2005 (Kurimo et al., 2006), where this algorithm successfully participated. It is a combination of the letter successor variety, first formulated by Harris (1968), and context similarity as described in the previous Chapter 3. Possibilities on how to effectively use the results are discussed and some briefly evaluated.

The results from Chapters 3, 4 and 5 can be used to attempt further analyses. One such analysis would be to attempt disentangling the various relations found in automatically computed similar word pairs. Various possible combinations of data obtained from similarity and co-occurrence computations with other observations or algorithms, such as disambiguation and morphology are discussed and prototypically tested. Contrary to the previous chapters, several such combinations cannot yet be implemented in an unsupervised manner. This is because some of the crucial parts, such as unsupervised acquisition of syntactic word classes (Biemann, 2006b) would be required, but are still in development. However, the partial results already outperform the naive approaches of just computing co-occurrences or distributionally similar word pairs.

The final chapter summarizes the most important findings and discusses the directions of further research. This work is not intended to cover all possible algorithmic solutions for language structure extraction extraction, due to the time and space restrictions of a single thesis. Open topics include, for example, the automatic induction of syntactic structure on the word, sentence (Klein, 2005) and text level. Other topics and applications are only mentioned, such as the induction of semantic relations between morphemes or sentences (an algorithm to find paraphrases in an existing corpus, for example), as well as algorithms handling the ambiguity of word units on any level - either to resolve it or induce its various meanings.

Acknowledgments

I would like to thank everyone who supported me during my work on this thesis. The first one to be thanked is Prof. Gerhard Heyer, my supervisor. He inspired me during our numerous discussions about language structure, natural language processing methods, but also potential applications and provided me with the opportunity to develop my own ideas. I am also very grateful that he trusted in my abilities, which had been a great encouragement for me all the three years I had been working on my thesis.

I would also like to thank my colleagues at the Natural Language Processing Department (University of Leipzig) for the many fruitful discussions and good ideas. Most particularly I would like to thank Christian Biemann, Hans Friedrich Witschel, Matthias Richter, Thomas Wittig, Fabian Schmidt, Ronny Melz, Patrick Mairif, Markus Ackermann and Chun Cui, and also Renate Schildt. Fabian Schmidt introduced me to many clever programming tricks to handle large amounts of data effectively. Matthias Richter helped me with intricate problems concerning Linux and Latex. Special thanks go to Prof. Uwe Quasthoff for his tireless efforts to create the 'Projekt Deutscher Wortschatz', which was both inspirational and extremely useful for this thesis. Thanks go also to my students Michael Bart, Marco Büchler, Michael Welt and Christian Beutenmüller for their interest and implementations that were useful for this thesis. Further thanks go to Uwe Tönjes for providing me with that extra bit of computational power in terms of lended hardware to run some of the more demanding experiments.

Extra thanks go to Mikko Kurimo, Mathias Creutz and Krista Lagus (Helsinki University of Technology), the organizers of the MorphoChallenge 2005, which took place just in time to evaluate the results of my morphology algorithm. I also thank Delphine Bernhard for the fruitful discussions about morphology.

Many thanks also to Martin Voigt and Jason Konoske for English proof-reading.

Warm thanks go to the great people who helped us with the children in the times of need: Vlasta Votrubová, Katrin Baasch, Alexandra Zielonka, Anna Wròbel, Marcela Řezníčková, Hans Friedrich Witschel, Charlotte Möller and Uljana Tschistjakow.

My warmest and deepest thanks clearly go to Denisa for not only being The Best Wife Ever, but also the best friend I ever had. All the years I was working on this thesis she would tolerate my bad moods, take care of the children when I needed some extra time and still be extremely kind to me. I also cannot emphasize enough how important it was that I could discuss with her the most intricate details of any strange algorithm or idea the moment I would make them up. With her lightning fast mind she would immediately grasp everything and help me sort apart the good from the bad ideas. She also taught me a great deal about psycholinguistics and how to work more 'scientifically', which I think greatly improved the overall quality of this thesis. I also thank my daughter Ellen and my son Jonathan for being such

wonderful children, better even then I ever imagined my own children could be. What can be more motivating (to finish a thesis rather sooner than later) than a beautiful daughter, smiling at me at every time of the day?

1. Theoretical Issues

1.1. Introduction

Natural language processing is comprised of multiple approaches to a variety of problems on all language levels, such as morphological, word, phrase and sentence level. On the one hand, there is the scientific motivation to understand what language is and how it works. On the other hand, there are applications requiring various language resources. One of the common aims is to create an automatic extraction system handling natural language phenomena, hence being able to automatically generate language resources. Such a system should rely on as little manually added language-specific knowledge as possible while producing maximally precise results. Increased automation and accuracy indicates a better understanding of the matter, while conversely minimizing the necessary post-processing when employing such a system to produce data for a real-world application.

To avoid misunderstandings about the terms used frequently in this work, the starting paragraphs offer a short description of them. Where appropriate, references indicate sections offering a more in-depth treatment of the associated concepts.

- **Language:** Apart from the standard perception of what language is, in this work its function as a tool (to communicate or store information) is emphasized. As shown in Section 1.3.1 it can be worthwhile to examine the structure of the language without also having to model the real world or an interpretant.

- **Language structure vs. language knowledge/content:** Language structure comprises structural elements that are invariant for many languages whereas *language knowledge/content* is the set of language specific assignments. For example, between the two words *deep* and *shallow* there is at least one semantic relation such as antonymy. While antonymy is generally observable in many languages, which word pairs are antonyms or which words have an antonym at all is specific for each language. Chapter 2 represents an attempt to model this distinction.

- **Automatic extraction system:** A system consisting of several algorithms that can extract language specific assignments (i.e. language content) reliably[1] for several types of structural language elements. The algorithms

[1] A reliable algorithm produces good results not only for one language or parameter setting, but

presented from Chapter 3 to 6 in their entirety represent such an extraction system.

- **Algorithm, finite set of instructions:** The term algorithm is used synonymously with “a set of rules”, assuming that there is a specific order in which the rules or instructions are executed. Language specific knowledge such as ‘tree is a noun’ is excluded from this notion and referred to as assignments or training instances. Again, Chapters 3 to 6 represent such algorithms. They all generate such language specific assignments.

- **Knowledge-free algorithm:** An algorithm can extract language content *knowledge-free* if it has no hard-coded knowledge specific only for the language it is applied to. Such an algorithm will not have received any samples of explicit language knowledge in any form.

- **Unsupervised algorithm:** An algorithm is *unsupervised* if no post-processing (which often is essentially the same as providing training sets in advance) or careful manual tweaking of thresholds for each language is necessary for it to operate successfully.

The central topic of this work concerns the hypothetical existence of such an automatic extraction system (of algorithms). It should be able to handle complex language phenomena, including: building syntax trees, parsing sentences, or extracting semantic relations between words, finding morpheme boundaries, morpheme classes, etc. Such tasks and their corresponding solutions often seem to cover only vaguely related research areas. One of the initial steps towards such a system is to examine the similarities of these tasks (apart from the fact that they handle language phenomena) and their most relevant differences (besides the fact that they analyze different pieces of language).

By means of these comparisons, it should be possible to lay a foundation for a language model based on algorithms[2], which would represent a strictly empirical approach[3]. Such a model is useful to produce a methodological framework for algorithms extracting language knowledge. The model is also useful for predicting new solutions to known problems, quite similar to how Mendeleev’s periodic table was useful in predicting previously unknown chemical elements.

Such a model cannot be considered a complete language model at any time. For one, it is based on algorithms operating solely on language form, excluding a modeling of ‘meaning’, in the sense of referring to objects in the real world. It also excludes the cognitive processes involved in associating language signs with world objects, actions, etc. Thus, in this work the term ‘language model’ refers solely

for most, or ideally all.

[2] In the sense of supported by existing algorithms

[3] Section 1.2 discusses empiricism in greater detail

to a model that describes signs, without the other two components (referral to objects and cognitive processes), see also Section 1.3. On the other hand, further insights into the structure of language could necessitate changes to the model. Treating any such model as complete interdicts the falsifiability of the associated hypotheses, therefore contradicting the empirical approach, see also Section 1.2.

Initially, these restrictions may appear too strong. However, the aim is to show that given such restrictions, it is possible to present a uniform approach to a variety of common natural language processing (NLP) problems. Along with that, prototypical knowledge-free and unsupervised solutions, including morphological boundary detection (Chapter 5), extraction of specific linguistic relations (Chapter 6) or clustering of word senses (Chapter 4), can be derived more easily. Furthermore, the development of new algorithms within the constraints of the model avoids the problem that affects solutions where it is difficult to provide explanations for why they function.

The structure of this chapter is laid out as follows. First, similarities and differences between present-day algorithms are discussed, along with the possible resulting generalizations. Then, relevant theoretical topics such as the empirical approach, the notion of meaning and the distinction between syntagmatic and paradigmatic relations are explored. The majority of these topics will be covered in-depth, but in a rather comprehensive than in a complete way, in order to clarify the assumptions used throughout the remaining work.

1.1.1. Similarities between NLP algorithms

Context: The most apparent similarity between natural language processing algorithms is their dependence on context. Be it a Markov Model (Markov, 1913), a collocation extraction algorithm (Church and Hanks, 1990), a word sense disambiguation algorithm (Lesk, 1986), a syntactic tagger (Brill, 1992), a terminology extraction algorithm (Witschel, 2004) or any other type of algorithm, they all utilize some notion of context. Contexts can be as different as co-occurrences of words within sentences, words within syntactic patterns or co-occurrences of morphemes within words. Even pattern-based algorithms for extracting named entities (Biemann et al., 2003) have the notion of context inherently built in: the patterns are fixed definitions of context. The Hidden Markov Model is entirely built upon a formalized notion of context. Any part-of-speech tagger utilizes various parts of the sentence in order to assign a syntactic word class tag. A terminology extraction algorithm may compare the text(s) and a corpus as context for individual words, but also n-grams of letters to identify domain-specific words (Heyer and Witschel, 2005).

Although it is possible to envision wordlist based part-of-speech taggers (or other kinds of algorithms) that could do without any context, variants using a context will always be able to outperform simple versions not using context. It could be argued, that i.e. Goldsmith's morphology extraction algorithm (Goldsmith, 2001) makes no use of context, but that depends on the definition of context: In this case the algorithm certainly does not use context on the sentence level. Since each word is analyzed based on the existence of other morphologically complex words, it could then be claimed that the algorithm does use context, but on the morphological level. Clearly, the notion of context is variable and correlates with the kind of units observed. If the unit is a word in a sentence, then it follows naturally that other words in the same sentence are its context. If the unit is a type (all occurrences of a given word), then all sentences in which this word occurs can be considered as the context of this type. However, if the unit is a morpheme, then the other morphemes within the same word form constitute the context of that morpheme.

Two immediate conclusions can be drawn from these observations. First, the definition of context needs to be sufficiently broad so that it is not limited to word forms in sentences only, for example. Second, the various language levels (phoneme, morpheme, and word form level, etc.) need to be treated formally in an equal manner with respect to the model (while of course allowing algorithmic differences).

Coverage and Accuracy: Another similarity across all natural language processing algorithms is that they represent attempts to find structure in something where structure is assumed to exist. Independent of the level of linguistic annotation, all algorithms introduce new structural elements or new types of structures to some form of natural language input. This also yields the fact that the majority of algorithms can easily be measured by coverage and accuracy performance.

Coverage performance, or recall in the Information Retrieval (IR) literature (hence used synonymously in this work), is defined as the percentage of all instances of a certain structural element that the algorithm was able to detect. In contrast to that, accuracy performance, or precision in the IR literature (also used synonymously), is defined as the percentage of correct vs. all found instances of structural elements. Both precision and recall are measured according to expectations of the designer of the algorithm.

Only the combination of both measures is meaningful, as it is trivial to create an algorithm that reaches maximum performance in one aspect while achieving zero-performance in another. For example, an algorithm supposed to assign POS tags to all words in a corpus assigns a single tag V (verb) to all 4 964 occurrences of the token *runs* in the British National Corpus. Precision (or accuracy) is 100%, but recall (or coverage) is 0.004% with respect to all tokens.

1.1.2. Differences between NLP algorithms

The primary difference between natural language processing algorithms is their varying **dependence on prior knowledge** about the language to be analyzed. Some algorithms, such as the aforementioned Goldsmith's morphology extraction algorithm, work in a genuinely knowledge-free way. However, there are other tasks such as the syntactic sentence parsing requiring taggers trained on rather large training sets of manually tagged sentences or utilizing manually created rule sets.

Types of dependence on prior linguistic knowledge

A common goal for all algorithms is that their operation should involve a minimum of manual work, or ideally none. They should retrieve as much knowledge as possible (recall) and make few to none mistakes (precision). An ideal goal is to construct language independent, fully unsupervised algorithms with both maximized precision and recall, which ultimately would minimize costly manual work and additionally maximize understanding of language (structure). In reality, the less manual work involved (in the form of smaller training sets for taggers or starting words for bootstrapping algorithms), the poorer the results. This effect is called the **acquisition bottleneck**: in order to maximize knowledge retrieval, the amount of manual work has to be increased, which is contrary to the goals of the exercise.

The manual work usually invested in almost all hitherto mentioned algorithms involves the deployment of a statistical tagger (Brants, 2000), a rule-based tagger (Johanessen, Hagen, and Nøklestad, 2000), or one combining statistics with rules (Brill, 1992). The tagger either preprocesses the corpus they are working on or sometimes detects the units to be analyzed, e.g. noun phrases in Hearst (1992) or Berland and Charniak (1999). A statistical tagger often requires a large set of manually annotated sentences for training (however see Chanod and Tapanainen (1995) for an example where a few thousand words suffice). Online sources are available for some major languages, such as PennTreeBank, Susanne or Negra, for the major languages (as well as for some minor languages, e.g. the Czech National Corpus). However, analyzing a language with these algorithms requires the availability of such resources for this language as well.

Currently, no state of the art POS taggers exist which would perform this task in a fully unsupervised manner, without training sets or word class annotations, out of a universal notion of grammar. Work on this topic by Resnik (1993), Schütze (1995), Schone and Jurafsky (2001b), Freitag (2004) or Klein (2005) shows that the quality of the results barely compares to supervised taggers, because as described by Resnik (1993), when using statistically significant co-occurrences as clustering features, the clusters tend to be both syntactically and semantically motivated. Most bootstrapping algorithms require an additional small set of manually given knowledge to begin with. Moreover, due to their design they rely heavily on the quality of this initial set, and errors at this point usually produce further errors in the results (Haghighi and Klein, 2006).

To clarify the distinctions underlying this work, linguistic information extraction algorithms can be divided into four classes.

Definition

A linguistic information extraction algorithm in its basic form is a finite set of (language independent) operations with a natural language input of finite length, which extracts information about the properties of the natural language or its units.

- *Type 0* (knowledge complete) algorithms encompass complete knowledge about the structure of the language used (e.g. rule-based systems) and apply it to the new input, eventually enriching the initial knowledge base.
- *Type 1* (machine learning, supervised) algorithms are only allowed to use training sets (like treebanks), or rule sets (like grammars).
- *Type 2* (bootstrapping, semi-supervised) algorithms are only allowed to employ language universals or a small, closed set of possibly language specific rules.
- *Type 3* (knowledge-free and unsupervised) algorithms extract structural information about any (natural) language or its units, without any additional language-specific knowledge.

Ideally, it should be possible to provide an algorithmic description of Type 3 for any kind of (natural) language structure to be extracted. This is because Type 3 implies the language mechanisms (e.g. structure) must have been truly understood - otherwise such a description would contain mistakes. This is also due to no further manual work being needed to accommodate for hitherto unseen input in that language, which could possibly even carry changes to that language.

Many algorithms, however, are at best of Type 2, including bootstrapping algorithms or those using language universals. For all algorithms depending on a POS tagged corpus (the default prerequisite for most algorithms), the Type 1 is applicable, because POS tagging must to be done manually or through the employment of a statistical tagger trained on a treebank. Algorithms of Type 0 are rarely in use, because language structure comprises such an expansive variety of information that encoding it all is considered too costly, even if it were possible at all. Rule-based taggers constitute examples of Type 0 algorithms.

In fact, if a Type 0 algorithm is applied on a large, varied corpus, its performance is likely to decrease (as opposed to applying it on a small sample corpus). This is because more aspects of language are likely to occur that the designer of the rules has not thought of. Contrary to that, the performance of a Type 3 algorithm usually increases the larger the corpus is on which it is being applied. This is because the statistical observations become more stable and generalisations are less likely to be based on local peculiarities.

Human supervision

Another source of differences between algorithms is the **degree of human supervision**. Apparently this correlates with the size of training sets needed for such algorithms. Generally, the larger the training set, or the more complex training data is required for the proper functioning of some algorithm, the higher the requirements are towards supervision, i.e. correcting errors made by the system, during or after the process in order to achieve satisfying results. The training sets and the supervision essentially add further sources of information about language. Supervision by humans is always based on intuition and (theoretically) complete knowledge of the language.

The question arising at this point is: can manually created training sets and supervision be (at least partly) replaced by a system of algorithms which produce the training sets for each other and supervise each other - in a manner perhaps similar to the mechanism operating in the human brain[4]? The underlying language model and the resulting algorithmic framework should allow a system design based on mutual support and improvement. This implies that at no layer of the system a task correctly solvable solely with human supervision is allowed, other than as a placeholder until a corresponding algorithm has been developed. The resulting goal is that a final instantiation of this model works completely (human-) knowledge-free and unsupervised.

Language dependence

Another clear difference between the algorithms is their **degree of language dependence**. Algorithms are often designed to function solely on one given language, e.g. grammar checkers. Even more commonly, while the algorithm itself is not truly language dependent, the training sets make an instantiation of that algorithm language dependent.

Of course, there are several known phenomena in specific languages, whose presence is constrained mostly to the corresponding language families (e.g. aspect in Slavic languages), therefore making any algorithm handling those phenomena language dependent. However, any language independent algorithm for a particular structural element should be designed a way such that it does not detect anything if the structure is not present in that language. Hence, if a system existed that

[4] See Weissenborn and Höhle (2001) for an overview

induced all knowledge by itself without initial help and later supervising, it would be inherently language independent.

Expectation preciseness

The structure induced by an algorithm has to meet certain expectations imposed by the designer of the algorithm. Consequently, a third source of differences between algorithms is the **degree of expectation preciseness**. For some algorithms the expectations can be formulated very precisely and almost unambiguously (e.g. POS tagger). In such cases, coverage and accuracy performance can be measured directly against a previously defined gold standard.

In other cases, such as Word Sense Induction (WSI) (Rapp, 2004), the expected results either deviate too strongly from the gold standard (WordNet for example (Miller, 1990; Fellbaum, 1998)), or the expectations are wrong. WSI is a particularly striking example of where neither appropriate coverage nor accuracy performance can be measured properly. This is because the expectations in the form of available gold standards (e.g. lexical semantic word nets or dictionaries) define a vastly different set of word senses then what is observable in any kind of corpus.

1.2. Empirical approach to language analysis

Building algorithms to extract language knowledge can be more generally subsumed under the empirical approach to language analysis and so the notion of 'empirical' (Greek - 'the experience') must be examined closer. Simply put, an approach is empirical if it involves observing real-world data, as opposed to using artificially constructed examples or intuition. It is also known as a method to construct hypotheses or disprove them using observations and experiments, or as the inductive (contrary to deductive) reasoning or formulation of hypotheses based on such observations (Popper, 1959).

Empiricism is also frequently associated with the rejection of 'innate ideas' (concepts of philosophers such as David Hume or John Locke), where 'innate ideas' is knowledge present in the human intellect prior to any sensory input. Other, more radical empirical views, such as behaviorism or rejection of innate ideas have been subject to strong criticism by Chomsky (1959) and Kuhn (1962), for example. Kuhn argues that creative new solutions to existing problems often occur outside of existing frameworks and not within the gradual theory development through observation and experimentation. He further argues that scientific experimentation is not as truly unbiased and neutral as claimed, because scientists have prejudices and prior knowledge which influences their experimental setup and interpretation of results.

Instead of providing arguments for, or against such views (however, see Sampson's detailed line of reasoning (2005)), a concise description of an empirical approach is laid out. It will serve as a foundation for this work and it attempts to connect a moderate view of empiricism and deductive logic. Some arguments against the criticisms mentioned above will arise from the facts provided in later Chapters (3-6) and be summarized in the end.

- **Observe:** In order to understand or structure something (a matter), unbiased observations or experiments are needed, or, more generally, experience with that matter is required. These observations must be as complete as possible, meaning that they need to capture as many different kinds of effects involved as possible. At the same time, they must not (and cannot) be complete to the point of including all possible observations of the observed matter.

- **Learn:** Based on the results of the exploration it is possible to detect regularities and formulate hypotheses or generalizations about the mechanisms responsible for the observed phenomena.

- **Verify:** Finally these findings must be verified through further observation and experimentation.

The result is a maximally generalized description of the mechanisms underlying the regularities observed in the data. It cannot be considered as complete until all possible observations have been performed which would verify the correctness of the constructed hypotheses. However, given that a mechanism was well understood, after a period of observing and learning the description should be approaching perfection. Observing copious amounts of further data should not then turn up examples disagreeing with the learned regularities.

The strongest possible indication that a given generalization of observed regularities is nearly complete (and correct) is the possibility of creating an algorithm which reliably detects these regularities in completely new, previously unobserved data. In the case of language, this would be an algorithm for annotating syntactic word classes applied to a randomly chosen **other** language (Klein, 2005; Biemann, 2006b). The fact that a given algorithm assigns a class to each word of a particular language does not necessarily show that the mechanism itself was understood (although the human that built this algorithm might well have understood it). Such an algorithm might be entirely unable to produce correct assignments for newly observed words or even words of another language. On the other hand, another algorithm producing such mappings correctly for an arbitrary language (previously not seen by the designer of that algorithm) shows that the mechanism of word classes has been understood. It further shows that it really is a mechanism and not a mere peculiarity of a particular language observation.

Given this, the description above can be restated more specifically in the following way, based on two assumptions:

- **Assumption 1:** There exists a method to determine whether a given sample of natural language contains regularities which are not yet covered by the current language knowledge, such as a set of extracted structural elements. This method can either be based on human introspection, or on statistical methods.
- **Assumption 2:** Each hypothesis about structural elements of a language can be reformulated as an algorithm that can extract these elements. The extraction quality of the algorithm depends directly on the quality of the hypothesis.

Given a language sample, it is possible to determine whether there are some regularities using the method from assumption 1. By beginning with any type of regularity, and assuming that it is caused by a phenomenon X (in the sense of underlying mechanism or structural element), the following steps must be applied until no more regularities can be detected.

- **Step 1:** Formulate one or several hypotheses explaining X, possibly in relation to other existing hypotheses.
- **Step 2:** Reformulate the hypotheses as one or several algorithm(s) Y, and using these, attempt to extract all occurrences of X without contradicting the existing hypotheses. Hence, X is responsible for a set of regularities (or occurrences), whereas Y detects another set of regularities. There are two possibilities then (see restriction below):
  - Ideally, both sets are equal. In this case the algorithms work flawlessly for the given sample and the hypothesis appears to be correct (and cannot yet be refuted).
  - Alternatively, some occurrences are only in one set ($X \backslash Y \neq \emptyset$) or in the other ($Y \backslash X \neq \emptyset$). Thus, it is either an occurrence of X that occurs but is not extracted, or something is extracted that is not an occurrence of X. In this case return to Step 1, formulating one or several new hypotheses along with revised or new relations between them and the existing ones.
- **Step 3:** In any case, the addition of a new hypothesis requires the existing ones to be checked for potential contradictions: if they exist, then Step 1 must be reapplied to all hypotheses involved in the contradictions.

One restriction is necessary for this description: realistically, it is virtually impossible to produce an algorithm that truly explains and extracts all occurrences of a given underlying phenomenon. This makes it necessary to proceed with only temporary maximally correct hypotheses, instead of absolutely correct ones. This results from the fact that observable effects of most language phenomena are interconnected with each other. Thus, the initial hypothesis formulated for any observable regularity must be either extremely complex or not absolutely correct. However, adding more hypotheses to the system of hypotheses enables a better reformulation of the existing ones.

Furthermore, algorithms representing the hypotheses should extract language specific knowledge. Therefore the mentioned hypotheses are meant to include only those which generalize over several languages.

1.2.1. Other notions of empiricism

This differs from some of the other definitions of empirical principles. Therefore a comparison is drawn with Hjelmslev's definition (Hjelmslev, 1974), which appears to differ most and can be translated as follows:

- The goal is to create a description of language.
- The description must be free of contradictions, complete, and as simple as possible.
- The completeness is more important then the absence of contradictions.
- The absence of contradictions is more important then the simplicity of the description.

Hjelmslev's definition of an empirical approach seems to differ from the description introduced above in one crucial aspect: Instead of defining a **method** on how to arrive at an understanding of language as empirical, it defines the conditions the **result** must fulfill in order to be empirical. This difference is of no consequence though, because the conditions required by Hjelmslev's definition can be derived from the description introduced further above. If after some observations a description of a mechanism is produced containing contradictions, then a corresponding algorithm will not withstand the verification step because it will make mistakes.

If the observations and subsequent generalizations about the learning step are incomplete, then regularities will remain in the observed data which cannot be covered by algorithms produced from those generalizations. For example, an algorithm aimed solely at extracting syntactic information from an untagged text corpus is condemned to produce erroneous results (even if it produces more correct than faulty information) as long as it does not utilize information from a morphology algorithm, a word class extraction algorithm, etc.

The simplicity of a solution will arise from the mere fact that overly complex generalizations and corresponding algorithms tend to be error prone, because they incorporate too many stray hypothesized mechanisms or respect too many weakly relevant parameters. Besides, overly complex generalizations tend to be too complex to be computed in an acceptable time-frame. It is also difficult to find supportive evidence for similar structures (e.g. a fully fledged generative grammar) which operate in the human cognitive processes in psycholinguistic experiments. The design of the model should therefore provide for the greatest possible unification and contradiction-free hypotheses, as well as simple algorithmic solutions.

A possible realization of such an empirical approach needs but one resource for proper functioning, a sample of the language to be analyzed. The adequate sampling of language in the form of a corpus spawned significant controversy where the participation therein is clearly not a goal of this work. Nevertheless, the problems associated with inadequate sampling need to be addressed, see Section 1.5. Particularly because sampling is the first step in the observation phase, the widest possible variety of phenomena needs to be captured in order to enable solid generalizations.

An issue even more crucial regarding this approach applied to the extraction of structurual elements of language is the apparent inseparability of meaning from language. In the following section this is further investigated.

1.3. Meaning

One of the goals of linguistics is to describe 'meaning' in a formal way, such that it is possible to compute what a given utterance means. This includes understanding the structure (syntax), the meanings of the parts of the utterance and its significance or extension (semantics). Syntax and semantics are commonly viewed as strictly separate topics.

Syntactic theories like generative grammar (Chomsky, 1957), or the more recent HPSG (Pollard and Sag, 1994) or extensible dependency grammar (Debusmann, Duchier, and Kruijff, 2004) analyze utterances on the sentence level. They do not only attempt to understand how linguistic units can be combined to form or parse structurally correct sentences in a given language. They are also designed to analyze how the meanings of the participating words alter the meaning of the sentence. Usually their formalizations do not rely on an empirical approach but on explicit human introspection.

Semantic theories like Montague's formalization (Dowty, 1979), the Discourse Representation Theory (DRT) (Kamp, 1981; Kamp and Reyle, 1993) or the Rhetorical Structure Theory (RST) (Mann and Thompson, 1987) offer a wide range of complex, formal descriptions of meaning. All of them struggle to explain what 'meaning' is supposed to be.

Montague's theory, for instance, attempts to map each sentence into a boolean function: a meaningful sentence is the same as a true sentence. However, what the meaning is remains unexplained. The DRT, on the other hand, is an attempt to construct a representation of the discourse as given by a succession of sentences.

The variables (nouns), modified by properties (adjectives) are changed, updated and related to each other through actions (verbs). The resulting discourse representation can then be connected to a given situation or context in the real world context through means of an unspecified cognition process. All these approaches are not empirical, because they presume various prior knowledge about language obtained introspectively. In the DRT for example, precise knowledge about the transitivity of a verb is necessary in order to elicit a correct discourse representation.

Nevertheless, these approaches to semantics introduce two important notions. First, the language-inherent structure that must be known without any recurrence to the actual meanings of the objects. For example, the knowledge that *to put* is a verb similar to *to give* in that it is ditransitive can be acquired without knowing the meaning of the word. Second, the actual meaning of an utterance makes only sense if an interpreting cognition process connects the discourse with a given real or imaginary set of circumstances, possibly by utilizing further sensorial input (such as seeing that the sentence *It rains* is in fact true in the given situation).

Similarly, the scope of this work is restricted in a way which encompasses the algorithmic description of syntactic and semantic *structural* phenomena. This work has a purely empirical basis, utilizing statistical methods while excluding any relations of language units with real world objects as well as excluding cognitive processes and mental concepts. This approach is very similar to Hjelmslev's (Hjelmslev, 1974), who assumes a principle of structure which leads to the possibility of exploring natural languages without recurrence to extralinguistic factors. Before describing the model in Chapter 2 and providing some of the sketched algorithms in Chapters 3 to 6, this restriction will be further described in the following section.

1.3.1. The three components of meaning

Structuralism is a linguistic theory that allows the analysis of language without access or reference to the real world or introspection. De Saussure's understanding of language has commonly been interpreted as a foundation of 'structuralism' (see Harris (2003) p. 78). This interpretation was reached by various authors and for reasons quite different: i.e. Barthes introducing his trans-linguistique (Barthes, 1983), Hjelmslev describing his theory of glossematics as a continuation of de Saussure's work (Hjelmslev, 1968) or Jakobson honoring de Saussure's distinction between syntagmatic and associative relations (Jakobson, 1956). Several aspects relevant to this work lead to this interpretation, assumedly initially made by Jakobson (see de Saussure (2001) p. 316):

- The notion of language as a structured system of signs that has lead to a direct application by Trubetzkoy who laid out the famous system of phonemes (Trubetzkoy, 1939).

- The distinction between language as a synchronic system and a diachronic development has lead to the possibility of dividing systematic factors of language from historical factors.
- The difference between *langue* and *parole* allows for differenciation between the language system common to all individuals speaking that language and actual usage of that language which might deviate from the language system.
- The notions of arbitrariness and relative arbitrariness along with the principle of linearity, which lead to syntagmatic and associative (paradigmatic) relations.
- The division of signs into concepts (signifié) and their sound patterns or graphical representations (signifiant) allowed, among other things, to introduce the previously mentioned principle of arbitrariness and to break away from a conception of language as a system of names applied to real world referents.

De Saussure's diadic distinction between signifier and signified can be perceived as a part of Peirce's model[5], which defines a triadic partition of 'meaning' (instead of partitioning the sign), although this comparison may be slightly far-fetched. After introducing monadic, first- and second-order relations, Peirce proposes that the meaning (of a symbol) is composed of several layers. First comes the sign (comparable to *signifiant* in de Saussurean terminology), which are the concepts and objects merely existing in a monadic relation with themselves. Then there are first-order relationships such as the sign-object and the sign-interpretant, which are then further subdivided into sign and object and then into iconic, indexic and symbolic. Ultimately, meaning is the second-order relation performed by the interpretant between the sign and the object. At this point, an interpretant is also divided into three versions (where each sign has interpretants of all three variants):

- **Immediate interpretant:** As a direct feeling or perception. Refers to the base idea or concept of the object.
- **Dynamic interpretant:** Anything a specific person (or other interpretant) can derive from a sign within a specific context or situation.
- **Logical interpretant:** The possibility of signs to represent other signs.

While the subdivision of the main relationships can be considered arguable, its value lies in the fact that, apart from the sign and the object, Peirce introduces the interpretant as the third component of meaning: Any sign can relate to any object, but this relation can make sense only when perceived by an interpretant.

[5] However, Peirce's model was developed earlier and there is not influence in either way, as the authors were not aware of each other

Peirce's approach can also be viewed as structuralist, because he proposes a dynamic interdependence of 'meaning' of words (Peirce, 1986). Assuming that the meaning of words depends on the meaning of other words, it becomes obvious that the meaning of a word can also influence or determine (in a strong variant of this hypothesis) the meanings of those other words. Dependencies between words can be expressed by relations. Whenever such a relation is changed by the interpretant, or a property of the object it was referring to changes, then the meaning of this sign changes as well. Furthermore, this change will affect all other words it was related to, which will subsequently change (with diminishing effect) the meaning of the words *they* are related to and so forth.

For example, if the concept of *running* disappeared, then the word *hurry* would differ slightly in meaning, appear in different contexts, and refer to different real world actions. Another reason that Peirce's linguistic writings can be considered as structuralist (and empirical) is that he states meaning to be an interpretation of regularities, which are also prerequisites and results of their (linguistic units) usage.

Hjelmslev (Hjelmslev, 1974) was among the first linguists to attempt an explicit formulation and generalization of the structuralist principle. According to him, the goal of linguistics is to explore language systems (as a synonym to simply language) that are only indirectly observable. Through such observations it should be possible to arrive at a calculus which computes all possibilities to combine language units into meaningful utterances. Contrary to generative linguistics, he considered texts instead of sentences as the base unit of observation, because only complete texts give enough contextually meaningful information about language units.

While Hjelmslev also maintained that the notion of 'meaning' is divisible into sign, object and interpretant, he explicitly stated that the system of signs has a hidden calculus. Proper empirical methods are required to uncover that calculus. The principles of combination according to the underlying calculus, are postulated to be based upon syntagmatic and paradigmatic relations. Hjelmslev proposed a method to arrive at the calculus, which works by beginning with text, the highest language level. Divisions and classifications of the divided units must then take place until a complete description of the classes of units and the relationships between them is achieved.

However, Hjelmslev's formulations are at times too restrictive. First, his view of language is static. Language is supposed to be something existing at a given point in time, and that excludes the genealogic aspects of language, which subsequently invalidates the diachronic aspect of de Saussure's language theory. Assuming a calculus which can produce all possible combinations of linguistic units leads to the implicit assumption that this set of possible combinations is fixed (though not necessarily finite). Otherwise the rules (the calculus) would require 'flexibility' rules accounting for dynamic aspects. This static view excludes new genealogic developments in language, because at that point the calculus would either have to incorporate them in advance or not be applicable to that language anymore. Contrary to that, the mechanism proposed in this work allows for the addition of new language utterances at any point, and can adapt to the new data by automatically producing an updated version of the language description.

Second, Hjelmslev's system can be described as too categorical, because there is no room for fuzzy or soft decisions. Something belongs to one fixed category or another, leaving no room for partial memberships or underspecification. As shown by Rieger (1991), the notion of fuzzyness, with particular respect to the semantic parts of language structure, is too important to be left out from considerations or even to be assumed as non-existing or faulty.

To recapitulate the points relevant for the next sections, Peirce and Hjelmslev introduced a triadic division of meaning into sign, object and interpretant (the specific meanings and terminology differ slightly). There is structure in the sign (in other words, there is a system of signs) that can be examined. A complete description of this structure is not a complete description of meaning. De Saussure and Peirce introduce the dynamic notion of language which any language description must account for. Finally, according to de Saussure and Hjelmslev, language consists of several levels, and on each level two identical principles of syntagmatic compositional relations and paradigmatic abstraction (or equivalence) classes (or associative relations in de Saussure's terminology) operate, though with significantly differing characteristics for each level (further examined in the following section).

1.3.2. Syntagmatic vs. paradigmatic relations

The basic principles describing the composition of complex language utterances have been explained in terms of syntagmatic and paradigmatic relations by de Saussure and later reformulated and explained more explicitly by Hjelmslev and others (Happ, 1985). Since the model introduced in the next chapter is based on this distinction, its historical development and existing definitions must be closely examined. Contrary to Peirce, who was particularly interested in a paradigmatic model of language, de Saussure was the first one who introduced a detailed distinction between syntagmatic relations and paradigms.

Linearity of syntagmatic relations

Linearity (also sequentiality) of utterance (as pointed out by de Saussure) is given by the fact that two or more words cannot be uttered (and perceived) simultaneously. Therefore, information must be ordered linearly into a string of codes (symbols). In order to convey the maximum amount of information intended with the shortest possible utterance (according the the principle of least effort (Zipf, 1949)), information can be compressed or encoded in various ways assuming that the perceiver of the utterance also knows the code. Because the transmission of utterances is usually noisy, and a given listener might not always be informed perfectly about the current situational context, redundancy is necessary to allow for correct understanding without violating the principle of least effort (see Martinet (1969) p. 167). Any given utterance can therefore be viewed as the least amount of work necessary to deliver the intended information to the listener while being sufficiently redundant to allow understanding if the context is unknown or the utterance was only partly received.

For example, if the listener is perfectly informed about the given situation and receives the utterance *Peter leave house* or *house Peter leave* undamaged, it suffices to tell him that Peter was in his own house 5 minutes ago and has left it to go to work. But if the listener is unaware about the situation and additionally misses some words randomly due to background noise, then neither of the utterances will suffice. On the other hand *Peter has **** his own house 5 minutes ago in order to go to **** and earn some money* would be an improvement, because several redundant information sources enable the missing parts to be inferred. Grammatical information in the form of previously agreed-upon correct word order and the word *has*, allows to reconstruct the first missing word. Semantically redundant information in the form of *to earn some money* allows to reconstruct the second missing word. Furthermore, the correct word order allows to restrict the search space of possible sentence interpretations by relating only expected items. Finally, a listener uninformed about that particular situation would be introduced to it.

Knowledge about correct word order, semantic information, grammatical information etc., is the result of experience with the language. Experience results from repeated exposure to similar situations with similar utterances and the learning of syntagmatic and paradigmatic regularities. Consequently, syntagmatic relations are either derived instantly from an actual combination of language elements in a given utterance, or drawn from experience.

Experience-based learning

While syntagmatic relations capture one dimension of language, another dimension of language cannot be explained by them. The dimension of equivalences or associations comprises all types of replaceability, or more generally, abstraction. If it is possible to replace one element of a given utterance with another element in order to construct a new utterance (possibly of a related meaning), then a paradigmatic relation exists between them. For example, it is possible to replace *castle* in the sentence *I live in a castle* with the word *cave*. The meaning of the sentence changes drastically, but the relation of *same-word-class* holds between the exchanged elements. This phenomenon is so strong that if the replaced word was an adjective like *nice* which stands in no paradigmatic relation to the noun *castle*, then a human would either consider the sentence nonsense or interpret *nice* as a noun he does not know yet.

Experience with the two different relation types can be summarized as follows:

- Experience with syntagmatic relations is the re-occurrence of formerly used utterances where the collocations are an especially strong variant of such syntagmatic relations. For example, the collocation *to beat about the bush* is known due to frequent previous use, but it would also be understandable to use *to run around the bush* in the same context and if used frequently instead of the first utterance. Gradually, it would become its own collocation, eventually replacing the original.

- Experience with paradigmatic relations is the re-occurrence of certain language elements in similar contexts. For example, based on the two utterances *I saw that wild dangerous tiger.* and *I saw that wild dangerous *recipatum.* it is possible to assume that *recipatum* might a carnivore similar to a tiger.

Free construction of new utterances

De Saussure also questions whether all correct combinations of language units are equally free. The distinction between speech and language (parole and langue) helps clarify this distinction. 'Language' is the experience of one person with a given language at a given time. Speech on the other hand, is the active use of a given language and allows the free combination of language elements to form new utterances not necessarily lying within the constraints given by the experience. The example *Teh voice actors.* drawn from an internet forum carries a deliberate misspelling of *the*. The changed appearance of the element that does not fit the usual experience with English still allows to understand the utterance. However, the correct meaning is revealed only with the community-specific knowledge that this misspelling marks irony (and only if the change is obviously deliberate).

The possibility of free recombination, and especially the difficulty to discern experience from free combination, is a view apparently not shared by Hjelmslev. In his view, free choice is restricted principally by the assumed existence of a calculus describing all possible combinations. Furthermore, an utterance is either proper in that language or not, in the sense that the calculus does or does not include this possibility, assuming that the calculus is correct. This does not allow any fuzzy borders between expressions of that language which rarely used or simply wrong expressions.

Non-linearity of paradigmatic relations

According to de Saussure, different kinds of associations or paradigmatic relations exist. Examples include: groups with identical stems such as *drinks*, *drinking*, *drink* or groups with identical suffixes such as *drinking*, *driving*, *running*, which could be termed morphological paradigmatic relations. Other examples include words with similar pronunciation (might be called phonological paradigmatic relations) and similar meaning (semantic paradigmatic relations). Both syntagmatic and paradigmatic principles operate on each language level. Therefore, a simple classification along the language levels suffices to retain order in all the possibilities of paradigmatic relations while at the same time allowing for a large variety in languages. For instance, in Czech, a language with aspect, the morphological level can differentiate between whether someone *was dying* - *umiral* or *has died* - *umřel*, while in English this must be done on the phrase level.

As opposed to syntagmatic relations, there is no linearization principle constraining the hypothetical order of units standing in a paradigmatic relation. It could be argued that word B is a closer synonym to a word A than to C. Firstly though, another respondee might disagree and secondly, this can change with time. Furthermore, no a priori order exists for elements in paradigmatic relations: for example, there is no meaningful order for the three words *he*, *she* and *it* which clearly stand in a paradigmatic relation with each other. Additionally, the exact number of elements within the majority of paradigmatic groups, according to de Saussure, is unknown at any time and can only be estimated due to fuzzy borders of whether something is in such a group.

Paradigmatic relations, relating words based on their meaning (or equivalence abstractions) to each other, raise the question: should a model based on such relations be concept-centered or lexeme-centered? In de Saussure's view, each language element is the intersection point of several paradigmatic groups (groups of words that stand in a paradigmatic relation with each other). This implies that though an element A is in group G, A's viewpoint of G may differ from unit B's viewpoint of A. For example, in a fictive word association experiment subjects would be given the word *elephant* and associate it with *zebra*, *rhinoceros* and *giraffe*. When given the word *zebra* they might associate it with *antilope*, *rhinoceros* and *giraffe*. While a group of African animals appears to emerge from that experiment, the exact contents would undoubtedly vary depending on the starting point. In this case it is impossible to describe the unique G. Instead it is necessary to describe two views G_A and G_B which still can remain similar or even equal to each other.

Towards automatic learning

A more complex and dynamic notion of paradigmatic relations has been introduced by Peirce. More explicit than in de Saussure's model, the classification of interpretants allows the usage of signs to refer to and interpret other signs. This implies that the meaning of at least some signs can be derived exclusively from other signs relating to them. Because this is circular, it is akin to bootstrapping algorithms which require a piece of knowledge from a previous step to produce new knowledge which is used in the following step. The main difference between de Saussure's and Peirce's language model is the presence of syntagmatic relations in de Saussure's model. It also becomes clear that the paradigmatic relations are at least partially based on syntagmatic relations, as seen in the next chapter.

As mentioned previously, Hjelmslev proposes a stricter language model. It is based on syntagmatic relations and paradigmatic equivalence classes (instead of relations), but there are no fuzzy borders or imprecise assignments of units to paradigmatic groups or classes. Because of this and the assumed calculus describing all possible language unit combinations, Hjelmslev might be considered a true generativist. However, he combines the creation of the calculus with a strict requirement for inductive empiricism, as well as the idea that language can be explored independently of both real-world objects and interpretants.

Without attempting to falsify the existence of such a calculus or sharp categorial assignments[6], this work has a slightly different goal. As stated above, the aim is to produce a learning mechanism (as a system of algorithms) with the ability to learn and extract language structure. This can be seen as only a reformulation of Hjelmslev's goals. Although this learning mechanism may not be able to produce all possible language unit combinations, it could classify new utterances within the learned structure and verify the completeness and lack of contradictions in the learned language structure.

Paradigmatic relations based on syntagmatic relations

Common to the three described approaches is an imprecise definition of what syntagmatic and paradigmatic relations are. This becomes apparent when trying to design algorithms for their automatic extraction (or to produce a corresponding calculus in Hjelmslevs case).

According to de Saussure's description, it is possible to observe any two words in a syntagmatic relation, which raises the question about the usefulness of such a notion. The distinction between langue and parole helps to differentiate between two facts: any two words *can be combined* into a (syntagmatic) expression and only certain words pairs *have been observed* to stand in a syntagmatic relation. However, given an infinite amount of experience that defines langue, then again any two words would stand in a syntagmatic relation with each other.

The problem of impreciseness appears to be even stronger with paradigmatic relations. Both de Saussure and Hjelmslev attempt a classification of various paradigmatic relations, but a proper description method or way of determining whether or not two words stand in a given paradigmatic relation with each other remains unclear. The general relation between the syntagma and the paradigma remains unclear as well. Currently, the only indication is that paradigmatic relations are somehow based on syntagmatic relations. This indication originates from the fact

[6] see (Rieger, 1991) for an in-depth discussion of this matter

that paradigmatic relations were introduced as equivalence or replacement classes in utterances within the syntagmatic paradigm.

Therefore a formal model is needed which is simple, yet sufficiently complex to describe paradigmatic relations based on syntagmatic relations. It should allow for a distinction of language levels, and a gradual refining and division (according to Hjelmslev's empirical principles) of the observed phenomena into syntagmatic and/or paradigmatic relations. The first step would be to extract the units of the given language level, then compute the syntagmatic relations and using these the paradigmatic relations. Then, assignments for element pairs could be refined and classified into more specific relations. Finally, continuous usage of such a system would provide an ongoing verification (and eventual re-learning) process.

1.4. Connectionism

Artificial Neural Networks (ANN) and the associated paradigm of connectionism represent an attempt to explain and model human intellectual abilities. This can be (and has been) used to model human-like language performance, among a large variety of other applications. Examples include NETtalk, which produces speech from text (Sejnowksi and Rosenberg, 1987), and the prediction of the past tense of English verbs (McClelland and Rumelhart, 1986). Neural networks are considered a viable method to extract language knowledge and there is in fact a vast amount of literature on this topic. However, for several reasons the entire approach of connectionism differs from the approach taken in this work:

1. ANNs learn from examples

2. The knowledge learned by ANNs is represented implicitly

3. Currently it is not possible to let ANNs learn complex correlations between patterns

The following description makes some strong simplifications and does not cover all exceptions and special cases. However, the primary differences between the approaches explicated further remain untouched by these. The basic functioning of any neural network depends on the fact that it is built from a set of nodes, which are interconnected either freely or according to a certain scheme. Each node may receive information (or energy). Each node can decide to send again energy to any or all of the nodes it is connected to. There are input nodes receiving activation from a source external to the network, as well as output nodes whose activation state is apparent to an external observer.

In the initial setup, no specific knowledge is represented in the network, be it in the decision algorithms of the nodes or in the connections between nodes. To give the network a purpose, it has to be trained first. The training consists of three steps, iterated n times. The first step is to activate the input nodes according to some form of input and then let the network iterate its activation propagation several times. The second step is to interpret the state of the output nodes. In the third step, the network is modified based on an interpretation of the state of the output nodes. If that state is unsatisfying, the existing connections between nodes are reconfigured. If that state is satisfying, the connections either are left as they were or are reinforced.

For any single piece of the description above there exists a variety of ways to implement it, such as allowing activation to flow in one direction only (feed-forward ANNs) or both directions (feedback networks). The success of a specific setup depends on specific purposes, but the overall picture remains the same.

The primary difference to the approach in this work is that ANNs learn principally by example. According to the preceding description, a training step is necessary to hinder the network from producing only random results. However, it is possible to expose an untrained ANN to input data and configure it in a way that it would recognize patterns and produce different output states for different patterns and similar output states for similar patterns. But in this case it would not differ in any way from yet another clustering algorithm. Except, perhaps, that with such a clustering algorithm it is even harder to make sure that it clusters according to predefined expectations. In this work it is supposed that it is possible to formulate a general learning mechanism which learns to extract specific, predefined types of information from input data, i.e. from raw text, without any training step.

Another difference of significance to the approach in this work is that if ANNs learn knowledge, they only do so implicitly (i.e. black-box). Given a perfectly trained (with respect to its measurable performance) ANN for a specific task (for example, deciding whether an adjective is negative or positive), it is virtually impossible to understand the reasons for its decisions from looking at its current configuration. Thus its decision could be based on complicated 'considerations' about the adjective's usage, but it also might base its decisions on completely unintuitive or even irrelevant properties and coincidentally produce perfect results in the specific evaluation (overtraining). The present approach though, is based on the premise that first and foremost, the phenomenon (for example the direction of the adjectives) must be understood by a human. This human is then able to formally specify that phenomenon. If that formal specification is correct, then its application to a specific language produces perfect results, otherwise it is incomplete or simply wrong.

The third major difference is the complexity of correlations between particular phenomena. It is well possible to expose and successfully train an ANN to recognize patterns in very detailed and complex input data. However, it is not feasible to apply ANNs to a multitude of different, though correlated, tasks. But the extraction of explicit language knowledge is exactly that - a multitude of different tasks (word sense induction, morpheme segmentation, POS tagging, etc.), which are correlated to each other. Although it is possible to design an algorithm or train an ANN for any specific task, better performance is achieved by accounting for some or all of the other tasks. For an explicitly designed learning mechanism this is not a problem, but it is for ANNs.

Despite the named differences, there are some interconnections. For one, the co-occurrences discussed later can be represented as networks, and activation spreading algorithms known from ANNs can be used to obtain explicit knowledge (Barth, 2004). As mentioned, ANNs can also be used as clustering algorithms, although currently the power of modern computers is insufficient to handle the large overhead of the ANNs, as compared to standard clustering algorithms. Thus, despite principal differences, the lessons learnt from connectionism were and will remain useful for other differing approaches, particularly such as the one in this work.

1.5. Sampling issues

Since all empirical models are based on data observations it is necessary to put thought into which data to use. Traditionally, language data is sampled in the form of a corpus of either printed (since the advent of mass-print), spoken (since the advent of audio storage devices) or electronically stored language (since the advent of computers). Clearly, using an audio data corpus with an algorithm that learns language structure directly from this audio data without any abstraction layer to higher forms of representation is not going to be successful. This is supported by the fact that in psycholinguistics strong evidence was found for a 'module' in the human brain that translates pure audio-input into phonemes, morphemes, etc., and that this module can clearly be differentiated from the other language processing modules, see Harley (1995) for an overview.

However, producing a corpus rich in syntactic, morphologic and semantic annotation, and eventually even correcting its contents (as happened with the Brown Corpus (Kucera and Francis, 1967)), will not be helpful either because this violates the empirical approach. The annotated data in this case can be used for evaluational purposes only[7]. Another reason for the low benefit of such fully annotated corpora is that they tend to be quite limited, due to the amount of work required for producing high-quality annotations.

However, the majority of current corpora are electronically available as transcriptions (or directly the original text) including very basic annotations such as word and sentence boundaries. Any further annotations, the specific format in which the electronic text is stored, the size or the kind of sampling used differ greatly among the various corpora as do the applications for which the corpora are used.

[7] Sometimes even this is impossible, because of a too large bias towards some theory underlying the annotation that does not fit real-world data. See Kilgarriff (1997) for an example.

Corpus size

It is possible to transform any format into almost any other by using a few simple programmed rules[8]. Therefore the specific format used in a corpus is nearly irrelevant. On the other hand, the size and content of a corpus are central issues because both factors significantly influence results of corpus based applications, see also Section 3.5.1.

The simple solution 'The more the better' seems to be enough in some cases (Banko and Brill, 2001; Brill, 2003). But doubling the corpus size does not typically halve error rates or double recall (coverage) of a learning algorithm. The relationship between size and performance is best described as being log-linear: In order to increase the performance of a learning algorithm by a fixed amount, the corpus size must be increased by an order of magnitude (Rapp, 2005a).

Most of the discussion concerning the proper corpus size (Sánchez and Cantos, 1997) centers around variations of the famous question of the 50 001st word (Carroll, 1967) and (Kucera and Francis, 1967): How large must a corpus be in order to encounter the 50 001st (different) word? This work takes a slightly different approach to this problem. The corpus needs to be sufficiently large to provide statistically reliable observations of all types of regularities, which are to be extracted by the introduced learning algorithms. At the same time, the corpus need not be large enough to contain observations of all possible appearances of the regularities. For example, to learn the various timeforms, it is necessary that for some verbs all time forms appear such as past, present and future. But it is not necessary that the same applies to all verbs.

The goal of correctly generalizing a set of regularities results in several parameters that become part of the function which determines the required minimum size of the corpus. One function is the complexity of the desired regularities. Another is the desired quantity of samples of the given regularity to be described (i.e. number of desired hyponym-hyperonym samples). Finally, the quality and adequacy of the corpus is an influencial parameter: If a fine-grained classification of plant species is desired by a taxonomy learning algorithm, then a simple newspaper corpus must be several orders of magnitude larger than a corpus of pharmacologic texts (and even then it might yield worse results).

Balance and representativity

The proper sampling of language data is an intricate problem. There is an implicit agreement among linguists that in general a corpus needs to be **balanced** and **representative** with respect to the language sampled. For example, according to WordNet (Miller, 1990; Fellbaum, 1998), the meaning of 'representative' is 'A single

[8] With the exception of undocumented or pre-computational formats where the annotation can be indiscernible from the language data itself like in Roget's original thesaurus.

item of information that is representative of a type'. Thus, as argued previously, it is by no means necessary to capture all possible language utterances or produce a perfect qualitative representativity (terminology according to (Oliva and Kveton, 2002)) to fulfill the representativity requirement. It is more important to capture all **types of phenomena** (or structural elements) expected to be learned from the corpus.

However, **balance** according to WordNet is a 'harmonious arrangement or relation of parts or elements within a whole'. This can be translated as follows: varying types of phenomena need to occur with a sufficient frequency and maintain a balance between each other. It can also be understood that the corpus needs to be quantitatively representative, according to Oliva and Kveton (2002). The specific considerations that influenced the construction of the currently available corpora differ strongly due to issues of further practical feasibility[9] and theoretical constraints.

A common interpretation of representativity and balance results in sampling a language across **text categories**. The ideal for corpora such as the Brown Corpus (Kucera and Francis, 1967)[10] or the British National Corpus (Burnard, 1995)[11] was to sample as many different yet generally accepted text categories as possible, including reportage, editorial, reviews, religion, etc. This automatically implies representativity with respect to different stylistic registers and appears to achieve the goal of representing varying forms of language usage. Additionally this seems to be a highly intuitive sampling method and usually results in concordances close to expectations. For example, in the BNC the word *house* indeed appears in contexts describing houses, buildings, living in houses or related. When examining a newspaper corpus, such a word occurs solely in contexts describing various kinds of catastrophes that could strike a house (i.e. as fire, flood or hurricanes), which at first seems unexpected.

Another notion of representativity closely resembling the one based on text categories is one across a taxonomy of topics or subject areas, as in the LIMAS corpus[12], (Rieger, 1979).

To achieve balance between various categories, a proper distribution along the 'importance' or 'size' of the corresponding categories has been used. However, this is controversial, because correct distributions can be drawn from the **perception** or **production** of texts, or both. The results can differ strongly between text types, such as between scientific text and daily newspaper text. For example, a scientist might read ten scientific texts while producing one. Compared to this,

[9] For example, it is not feasible to construct a large representative corpus of a language that is not spoken anymore

[10] The Brown corpus is freely available from, for example, http://clwww.essex.ac.uk/w3c/corpus_ling/content/corpora/list/private/brown/brown.html

[11] see http://www.natcorp.ox.ac.uk/ for more information

[12] The LIMAS corpus can be accessed on the internet: http://www.ikp.uni-bonn.de/Limas

newspaper texts would have a much worse ratio like one hundred perceived texts to one produced even for the most active journalists. The Czech National Corpus[13] is an example of a corpus to take the perception distribution of text categories into account, which is based on surveys estimating language perception, including such media as television.

Attempts at including spoken language in corpora reflect the awareness of the vast differences in language usage in **modes of language** used, such as speech vs. written text. The London-Lund corpus (Svartvik, 1990) was one of the first corpora of this kind and it included a psycholinguistically founded hierarchy of various modes of spoken language in which language samples could be classified into categories such as spoken, specific monologue and specific spontaneous vs. prepared monologue[14]. However, due to practical constraints, the percentage of speech in any modern large corpora will remain small, as observed in the BNC, containing only 10% of spoken text. Therefore the balance will not be considered.

Diachronic vs. synchronic sampling

Another possible distinction concerning representativity is between **diachronic and synchronic sampling**. Most corpora represent attempts to produce a synchronic sample of language usage by including texts from a very narrow time frame only. This has advantages such as mostly avoiding word meaning changes during the sampled time frame or coherence but also disadvantages such as possible word usage biases towards events specific for that time frame.

For example, the German usage of *Kohl* (cabbage) in the 90s would be strongly biased towards the chancellor *Helmut Kohl*. However, especially when performing diachronic research (e.g. for etymological research), it is important to sample across a certain historic period to measure changes in language usage (and possibly correlate them with historical events), using corpora such as the Helsinki Corpus of English Texts (Kytö, 1996). Diachronic sampling of language is also possible on a fine-grained scale, for example using a daily newspaper corpus to research trends, media visibility or other topics (Richter, 2005).

Impact of non-representativity

When designing corpora for algorithms that learn basic language structure (i.e., word classes, syntactic structure or semantic relations), it is less important to have a representative corpus with respect to text genres, categories or other such high-level properties. The pure size of the corpus, as shown in (Church and Mercer, 1994; Brill, 2003; Rapp, 2005a), is the largest factor influencing the results of

[13] The Czech National Corpus can be freely accessed and is located at http://ucnk.ff.cuni.cz/english/

[14] see http://khnt.hit.uib.no/icame/manuals/londlund/ for more details

such algorithms, because these basic structural elements occur independently of preferred text categories or whether a balance was reached.

Obviously, using vast quantities of similar data will benefit less than using the same amounts of varied data. Using a non-representative corpus can result in missing classifications, e.g. specific language units not classified anywhere into the found structure - although the structure itself might still have been found. A good example is the attempt to automatically derive a general taxonomy of topics from a large newspaper corpus. While a taxonomy as a structural element will certainly be found in such a corpus, its exact contents will only mirror the common newspaper taxonomy, dividing everything into politics, local news, sports, etc. Therefore this taxonomy would not comparable well to a taxonomy derived from a representative corpus.

It can be concluded from the variety of sampling methods presented that representativity is only of marginal importance to the goals of this work, as shown further in Chapters 3 and 6.

Completeness of a corpus

When discussing representativity it is also helpful to ask whether two corpora can be representative if they have not a single text or sentence in common? When using the definition of a representative sample from statistics, the answer is clearly yes, as long as both corpora are (statistically) representative with respect to the language sampled. A similar question is: when can a corpus be considered complete? If the sole purpose of the corpus is to cluster various word classes, then even one text could suffice. Additionally, if only a domain-specific ontology is to be extracted from the corpus, then a likewise domain-specific corpus is better suited than a representative corpus (of that language).

Typically, more goals are included, in which case the corpus is complete if and only if it is large enough a sample to enable algorithms (or humans if they do not know that language yet) to learn all the structural elements present in that language, possibly missing a part of the specific assignments to classifications as discussed above. For example, although one goal is the discovery of word classes, it is unnecessary to correctly assign the word *medicine*, as long as a sufficient amount of meaningful assignments for all word classes exists. When a corpus is complete therefore directly depends on the quality of the algorithms, and the complexity of the structural elements. These, among other factors result in an unknown lower boundary for the size of a corpus, because even the best algorithm or human would not be able to learn word classes from a single sentence only.

It is possible to provide an objective mechanism that helps to determine whether a corpus is representative with respect to the structure searched for. Similarly to the method of cross-validation, a corpus can be split into two (or more) parts. Then, an algorithm can learn a certain structural element, for example a taxonomy, from two independent parts. If the taxonomies learned from the two parts significantly differ, then it is highly probable that the corpus is not yet representative with respect to that structure. However, this can also be misleading, because the partition into two parts might cause the one part to contain all journalistic texts and the other part all literature texts, for example. Thus, an algorithm designed to classify genre types would produce very different classifications from the two parts.

On the other hand, according to (Church and Mercer, 1994), to detect whether an observed occurrence is only a quirk of the given corpus, other available corpora can be utilized. If the observed occurrence is frequently observed in one or only a few of the corpora, then it is very probably just a quirk of that corpus, such as the otherwise quite uncommon *Hear! Hear!* from the Canadian Hansards Corpus.

1.6. Summary

To recapitulate the assumptions used in the following chapter:

First, it is possible to divide the notion of meaning into three parts and the tripartite relation (or three binary relations) between them. One part, the language, can be examined independently of the other two parts, real world objects and the interpretant (or concepts, depending whose version of tripartite or binary relation is used). Using a language model based solely on the relations between signs, it is impossible to describe what a meaningful or true sentence in the meaning of formal semantics (Dowty, 1979) is - this would require a modeling of at least the (real or imaginary) world as well. Furthermore, it is impossible to grasp what 'meaning of a given utterance' is, because an assignment of which sign relates to which real world object is missing.

However, it is possible to tell whether a sentence is grammatically correct or seems semantically plausible (not plausible, but possible would be the utterance *A blue mistake*), whether or not a previously unseen word resembles a word of the specific language (such as *foryness* vs. *xzfyxsw*), or whether a given text is a proper text or a random collection of otherwise plausible sentences.

The scope of the model presented in the following chapter is therefore limited to language structure, without any cognition or world modeling, but this limitation still includes everything application areas such as Information Retrieval or Information Extraction can utilize.

Once descriptions and models of both the world and the interpretant can be added (such as knowledge representation or agent systems), complete meaning descriptions might become possible. Both fields, IR and Information Extraction (IE), currently are severely hampered by the nonexistence of an automatic language handling system requiring replacement by notoriously incomplete and heavily biased manual resources. A combination would therefore be mutually beneficial.

Second, the underlying structural principles of language appear to be composed of paradigmatic relations built upon a syntagmatic linearization. More abstractly, this can be formulated as the principle of linear concatenation as a special form of composition (syntagmatic) and non-linear abstraction (paradigmatic). The exact definitions of these principles are not yet sufficiently precise to enable algorithmic solutions based on them. Especially the difference between the theoretical possibility of almost any language unit combination and the sampling that can be undertaken to measure practical usage of language appears to be problematic. Furthermore, these principles seem to operate quite similarly on all levels of language, yet the specific instantiations of them can differ strongly. This speaks in favor of a learning mechanism able to learn specific appearances from an abstract understanding of the principles of composition and abstraction. Additionally, the dynamic character of language must be accounted for, which causes new language units to emerge, old ones to disappear from usage and existing ones to change their usage characteristics.

Finally, the results of this work are supposed to represent a simulation of an empirical inductive learning resembling the one proposed earlier by Rohwer & Freitag (2004) or Finch (1993). This requires samples of the languages whose structural elements are to be learned. The best, currently available form of such samples are (as large as possible) corpora of (electronically available) written language usage. The goal of the simulation is not to inductively learn one language by manually observing its usage and then producing a complete (and as error free as possible) description of that language. Instead, the goal is to produce a general system of learning algorithms that can simulate inductive learning and produce a description of (parts of) the language structure. Several specific learning algorithms are presented and it is demonstrated that the knowledge produced by individual algorithms can be combined to reduce the individual error rates and increase overall performance.

2. Structural Language Model

In this chapter, a structural(ist) and implementable model of language (SIML) is introduced. It is derived from a structuralist background as outlined in Chapter 1 and represents an extended and improved version of earlier work (Bordag and Heyer, 2006) (see also (Heyer, Quasthoff, and Wittig, 2005)). The SIML comprises structural elements and properties of language compatible with current and foreseeable capabilities of automatic and knowledge-free learning algorithms both from NLP and AI. Though the structural elements in this SIML include examples from fields like morphology, syntax or lexical semantics, it does not aspire to be a complete language model in the traditional sense. For example, it avoids mapping 'true' semantic meanings of utterances to real-world situations or truth values. The SIML also excludes purely introspection based features - or rather, ignores them until a **discovery procedure** (Harris, 1951) (or learning algorithm, in modern terms) becomes available for them. The generalizations about the considered structural elements are held as abstract as possible to avoid any language specific dependencies.

The SIML is not an entirely new model. Although it is not restricted to a pure vector space modeling, it does represent a mathematical understanding of language, quite similar to the one recently described in detail by Sahlgren (2006). The primary new contribution it attempts to make is the equality of language levels with respect to methods applicable to them. The claim is that **linguistic knowledge can be computed on any level using the same principles** (and thus with methods differing only in details). These principles are based partially on existing algorithms, as mentioned above, but also on the long line of quantitative linguistics research. Although any particular work is primarily concerned with only one particular language level (from the morphological level (Gerlach, 1982) up to the text level (Altmann and Krupa, 1964; Altmann, 1980; Köhler, 1983; Rothe, 1983; Hřebíček, 1989)), this line of research is a rich information source about the similarities between various language levels. However, not all insights from this field have become part of the model yet, such as the observation that "The relative number of sounds in the syllable decreases as the number of syllables in the word increases" (Menzerath, 1954).

An example for treating language levels equally is the description of syntactic word classes as a structural element of a given language based on the annotation of a dozen of words. Even if an algorithm could learn to classify all remaining words to the given classes, it would remain dependent on that particular level and language. Contrary to that, it is possible to hypothesize that on each language level different classes of elements exist, such as bound and free morphemes on the morpheme level, word classes on the word level, noun or verb phrases on the phrase level and others.

It can further be hypothesized that these classes are directly observable from the raw data through examining the usage patterns by providing a discovery procedure. This would enable the design of learning algorithms that produce for all possible languages a clustering and classification of language units on any language level. The result of these algorithms ideally represent what a linguist would have produced manually, i.e. particular class assignments for words. The possibility of such a system and preliminary thoughts on an implementation have been realized by several other authors, including Finch (1993) or Rohwer and Freitag (2004).

As stated, the SIML should be based on a minimal amount of maximally generalized principles. It cannot (yet) be implemented as a learning mechanism (or discovery procedure) to let it actually 'learn' language structure. Either specific implementations for particular types of language structure need to be developed, according to the empirical approach as described in Section 1.2, or implementations need to be provided as they might result from artificial intelligence research about self-organized acquisition of behavior (or knowledge) (Der, 2001). The language model presented here cannot yet be considered complete or exhausting. Contrarily, it remains open to changes and additions, and ideally it might be a step towards a more general learning mechanism not bound to language solely.

Algorithms derived from that model (or ones fitting into it) should comply with the most important premise: they should learn all (or as many as possible) aspects of a given language solely from a given sample of that language, according to the sampling described in Section 1.5. Methods not breeching this premise and being of Type 3 (see Section 3.1.4) include mathematics, in particular statistics. As such, this work is strongly influenced by a general understanding of approaches to language algorithms as described by Finch (1993), Manning and Schütze (1999), Jurafsky and Martin (2000) and Heyer, Quasthoff, and Wittig (2005).

The SIML can also be considered to be based on the distributional hypothesis as formulated by Harris (1968), and in fact, the ideas presented here can be viewed as an attempted continuation of Harris' work, see more in Section 3.1.1. Applications and algorithms described by Biemann, Bordag, and Quasthoff (2004), Biemann et al. (2004), Heyer, Quasthoff, and Wittig (2005) and Biemann (2006b) can be considered as directly influenced by the ideas presented here.

Another approach, similar to the one in this work, is Rieger (1989). The global context introduced later in this chapter corresponds to the syntagmatic α-abstraction, whereas the similarity operation corresponds to the paradigmatic δ-abstraction (or semantic topology). It is important to repeat (from Section 1.3) that the approach taken here is to view human cognition, the world and its objects[1] and language structure as separate topics, unlike Rieger's approach (Rieger, 1991). Nevertheless, these two abstraction layers as the base of further algorithms were initially introduced by Rieger. The restriction clearly allows algorithms to learn linguistic structure only from a large text sample. Given this, the proposed model might be viewed as fitting into Rieger's Semiotic Cognitive Information Processing (SCIP) System (Rieger, 1995) as one stratum, the language itself. The interconnections to the other strata remain unexamined in this work.

Since the language model is meant to describe most (ideally all) types of structural elements of language, it is necessary to state clearly what this includes. There are a few structural elements of language that are traditionally considered to fall under this description, including syntactic word classes, the syntactic structure of sentences and morphology. But these and other structural elements along with their derived research fields are considered to be mainly topics strongly differing from each other. In this work they are considered to be mostly the same mechanisms, only on different language levels.

Therefore, the first distinction described in the next Section 2.1 is the one of language levels. The remaining model is built upon that notion. There is a number of other topics at least implicitly covered by the SIML, which are traditionally excluded from the list of language structure elements. They are excluded because they are considered inseparable from the full notion of meaning, thus including either introspection or (a modeling of) real world objects, or both. These excluded topics comprise such important (for IR applications for example) topics as semantic word relations, word ambiguity, automatic clustering of morpheme classes, ontologies and more.

In addition to being excluded from the traditional understanding of language structure, these topics are hard to define properly. For example, the idea of word ambiguity intuitively includes a distinction between 'meanings' of a given word, available through introspection or knowledge about that language in general and usually given as definitions in natural language (i.e. a phrase or a sentence). But first, the most frequent distinction between word meanings is made between the different word classes of that word, i.e. *run* as either *to run* or *the run*. Second, the remaining distinct meanings (as given from introspection) of a word occur so infrequently that their existence is rarely verifiable, even in a large corpus, see also Kilgarriff (1997) for a discussion on this topic. Finally, algorithmic solutions able to differentiate the really observable different usages of a given word in a given corpus already exist (Neill, 2002; Rapp, 2004; Ferret, 2004; Bordag, 2006b). However, the resulting differentiations between meanings produced by these algorithms deviate strongly from those found in manually constructed dictionaries, implying that either the algorithms are wrong, the dictionaries are not based on actual data,

[1] Where humans use the language to refer to those objects

or a combination of both.

For example, when designing algorithms for computing semantic relations between words, the first step is to define the relations to be computed. While some semantic relations, such as cohyponymy, appear to be more viable than others (e.g. synonymy), it is the salience of the relation and the theoretical foundation of the method to calculate it that are of greatest influence. If the relation is made up arbitrarily, and no objective means exist to differentiate between words standing in this relation or not, then clearly, there will never be an algorithm able to calculate them more precisely than a random baseline algorithm. On the other hand, if a relation is of large impact on language use, in particular on the co-occurrence or non-co-occurrence of words; and if there are many observable hints on how to recognize such words, then algorithms can be constructed incorporating as many of these hints as possible to make reliable predictions.

Nevertheless, the majority of these algorithms will never make 100% correct predictions, meaning that other means for improving the results will be necessary. Without incorporating any further knowledge sources, the overall effectiveness of the approach can be improved by combining several algorithms to solve a problem and by utilizing classifier-bagging, for example. Since linguistic relations are often mutually exclusive (e.g., a word's synonym is never simultaneously its antonym), meta-rules that can be derived from the language model or those directly defined in it can effectively help to combine different algorithms and improve the overall accuracy of results.

2.1. Language levels

One of the first and fundamental findings in linguistics is that language is divided into several language levels with one or more related research fields in linguistics, such as phonology, morphology, syntax and semantics. The existence of these levels can be intuitively tested by producing several meaningful units on one level and attempt to combine them to a new unit - on a higher level. The test would then consist of assessing whether it is also possible to create a senseless unit on that higher level using sensible units from a lower level.

As an example, meaningful units on one level could be the common English morphemes *stuff*, *taste*, *food* and *less*. It is possible to create the non-word **tastefoodstuffless* - one level higher. However, using the same morphemes it would also be possible to create a correct phrase, such as *tasteless food stuff*, a unit two levels higher. As the *food stuff* especially shows, the division between language levels is not always clear. The **tastefoodstuffless* example could be used as a composite, a phrase or even as an answer to some previous question, therefore as a sentence (though likely to be considered ill-formed according to traditional syntax). This implies that when defining the language levels in the model to be presented, it is necessary to take such possibilities into account, while still allowing for an algorithmically graspable distinction between language levels.

Though the particular language levels can be considered well known, the definition given here specifies only that the levels exist, without restricting the definition to the existence of particular levels. Neither does it prescribe a strict vector space modeling of the levels as might be suggested by the example below. This allows to scale implementations of the model by need between finer-grained or coarser versions of levels. More importantly, there are principles operating on each level similarly which allow simple or atomic units (like morphemes) to be combined into complex units (such as word forms or phrases). Such combined units then represent the simple units for the next higher level. Based on empirical observations it becomes obvious that there must be another principle, namely that of abstraction through equivalence classes. These equivalence classes allow for similar utterances on one level to appear with only one part of that utterance changed for another. That implies that the two interchangeable parts could have something in common (such as syntactic class of word forms when interchanged on the sentence level). The two principles therefore are

1. **composition,** and
2. **abstraction** through equivalence classes.

Composition places atoms into a stream of atoms according to a number of rules. These streams of atoms can then represent complex units. If the hypothesized composition rules (be it explicit rules or only a kind of experience as described in Section 1.3) were obeyed, then the complex unit might be considered a proper simple unit on the next higher level. If not, then an interpreter might still have the possibility to understand the utterance, but will likely find it an incorrect usage of language, since it would not fit his previous experience of language.

Abstraction on the other hand, or *selection* (in the terminology of traditional structuralism), allows classifying sets of atoms into equivalence classes of atoms. Atoms within one class all have something in common while being distinguishable by something else against all other atoms. These findings seem trivial, but as shown later, constitute the cornerstone of a theoretical framework in which automatic calculation of semantic relations can be explained.

Definition

To formalize the notion of language levels, a certain level L_l with l={*phonemes, morphemes, words, sentences, texts, ...*} of a language L consists of two sets $L_l = (A_l, C_l)$. A_l is the set of all possible atoms on this level and C_l is a subset of all their possible combinations.

Since order matters and there are rules of composition, C_l is a set of all possible combinations of various lengths of the atoms. One possibility to express that is to take all possible concatenations of all atoms A_l^*, the Kleene closure. While this possibility is being used as default for most of the remaining work, this does not account for other types of composition, such as the Hebrew morphologic system. Alternatively, it is possible to entirely disregard the rules of composition for certain algorithms, in which case C_l is treated as a set of sets instead of tuples.

For example, in Chapter 3, the order of word forms within sentences is ignored, but the order of letters or morphemes within the words are preserved. Hence, C_{words} is a set of various sentences, where each sentence is a set of word forms. Hence, the two utterances *The guard dog barks!* and *The dog guard barks!* represent the same complex unit. However, $C_{morphemes}$ is a set of word forms, where each word form is a concatenated string of morphemes. Thus, *house-fire* and *fire-house* are two distinct word forms, despite consisting of the same morphemes.

It is possible for an atom of level L_i to be also an atom on a higher level L_{i+n} but not on a lower L_{i-m}, although it still has its own representation both on L_i and on L_{i+n}. For example, the atom a as a phoneme on the lowest (the phonological) level can simultaneously be an atom on the morphology and sentence level as well, as is the case of the interjection *A!* (with the exclamation mark being an additional atom). On the other hand, a composition rule on level L_i can implicitly modify the rules of composition on a lower level L_{i-n}. For example, if the word *yesterday* occurs when constructing a sentence, the likeliness of *did* occurring in that sentence is significantly higher than of *do*.

To elucidate the definitions given so far, some simplified examples can be constructed. The exemplary language to be described is the written natural English language L. The language levels comprise:

$$l = \{letter, morpheme, wordform, sentence, text\} \tag{2.1}$$

It is assumed that for each level a set of proper[2] complex units exists, that is a subset of the Kleene closure of its constituting atoms, i.e. the set of their possible combinations. In the following, the assumption is that for each level the set of proper complex units can be determined empirically. The framework itself does not provide those sets, but defines the principles of how the construction processes may take place, or can be inferred, based on empirical evidence. First, the lowest level $L_{letters}$ of L is defined:

$$L_{letters} = (A_{letters}, C_{letters}) \tag{2.2}$$

$A_{letters} = \{a, b, \ldots, z\}$ is the alphabet of the language L on the level *letters*.

It is possible to concatenate letters in order to produce strings of the form *abz*:

$$C_{letters} = \{a, b, ab, \ldots, zaz, \ldots\}. \tag{2.3}$$

[2] Proper for example in the sense as learned from experience

$C_{letters}$ can then be defined as a subset of all possible strings of arbitrary length: $C_{letters} \subseteq A^*_{letters}$. Thus, strings such as *aa* or *baaa* or *aaab* are complex units on the level of letters. However, not all letter combinations are actually observable in a corpus, therefore subset relation. The specific letter combinations that are present in English can be learned from observing a corpus of English language.

Second, the level of morphemes $L_{morphemes}$ of L is defined:

$$L_{morphemes} = (A_{morphemes}, C_{morphemes}) \tag{2.4}$$

Generally, the atoms of a given level are a subset of the complex units of the lower levels. In this case, however, there is only one lower level, so the definition is trivial: $A_{morphemes} \subseteq C_{letters}$. The subset relation describes that the string *boat* is a morph whereas the string *aaab* is not a morph (since that combination of letters has not been observed in the data). Not all complex units observed on the letters level are really morphemes, thus it must be a subset relation.

This definition actually defines only morphs as surface units of the morpheme level. However, the methods introduced below allow to find classifications of and relations between morphs which eventually allow a set of morphs to be viewed as a morpheme. A morpheme would then be a collection of morphs that were found to belong to the same class or share the same attribute. Therefore, labeling this level 'morph level' would be misleading about its expressiveness. The same applies to the word level.

The complex units $C_{morphemes}$ of this level can again be viewed as a subset of all possible compositions of the atoms of this level of all possible lengths (of the atoms): $C_{morphemes} \subseteq A^*_{morphemes}$ and is the set of morpheme combinations. For example, *boat* is a complex morphological unit of length 1, *boatboat* (by doubling the morpheme *boat*) is a complex morphological unit of length 2. In another example *boats* (by taking the morpheme *boat* and concatenating it with the morpheme *s*) is also a morphological unit of length 2. Hence, the definition of all levels up to the morpheme level can be summarized as follows:

$$\begin{array}{ll} A_{letters} & = \{a, b, \ldots, z\} \\ C_{letters} & \subseteq A^*_{letters} \\ A_{morphemes} & \subseteq C_{letters} \\ C_{morphemes} & \subseteq A^*_{morphemes} \end{array} \tag{2.5}$$

The third level L_{words} and all higher levels follow identically to the morpheme level, hence they are abbreviated. However, in practice these next levels hold greater importance than the lower ones, because these are the ones most easily observed:

$$L_{words} = (A_{words}, C_{words}) \tag{2.6}$$

The atoms of this level are the word forms as they occur in written English texts. They are a subset of all possible morpheme combinations of all possible lengthes, which excludes *boatboat*[3], but includes *boat* and *boats*:

$$A_{words} \subseteq C_{morphemes} \tag{2.7}$$

The complex units of this level are usually concatenated using characters such as space or comma as part of the alphabet. The atoms and the complex units on this level are typically (though not in Chinese or Japanese, for example) easily observable, because words are separated by spaces and sentences by full stops, exclamation or question marks. Since it is easy (but not trivial, due to composites and collocations) to observe word forms because they are separated from each other by spaces, it is a good starting point for algorithms that learn morpheme boundaries, see more in Chapter 5.

The fourth level, the level of sentences, $L_{sentences}$, again follows from its predecessor, the level of word forms:

$$L_{sentences} = (A_{sentences}, C_{sentences}) \tag{2.8}$$

The atoms on this level are all sentences as they have been encountered in written English texts. They are a subset of the set of all possible combinations of word forms to sentences of all possible lengths:

$$A_{sentences} \subseteq C_{words} \tag{2.9}$$

Similarly to the previous level, complex units of this level are concatenated using additional symbols (full stop, etc.).

This extremely simplified example is intended to demonstrate the concept of the language levels. Neither does it explain how the complex units can be observed in order to learn how to distinguish proper from improper ones, nor does it explain which kinds of rules can be learned from observing complex units of a given language level. The two basic principles were stated to be composition and abstraction. Proper composition can also be based on compliance of **syntagmatic** relations between the used atomic units, whereas abstraction can be based on compliance of **paradigmatic** relations between the used atomic units. The following paragraph describes these two notions.

2.2. Syntagmatic and paradigmatic relations

One of the most important distinctions made by de Saussure is the dichotomy between **syntagmatic** and associations, i.e. **paradigmatic** relations[4]. Syntagmatic

[3] Unless, of course, the corresponding corpus, from which language knowledge is to be obtained from also includes the present work as well.

[4] de Saussure does not use the term paradigmatic. Instead he introduces associations which include a broad spectrum of relations between words (such as whether they rhyme or not). The paradigmatic relations in this work represent a subset of these.

or paradigmatic relations in a language system relate two atoms that belong to the same level. Two atoms are syntagmatically related if they can be **composed** (or simply concatenated), that is appear together in some expression, for example *torch* and *shines*, complying (agreeing) in function and meaning. Two atoms are paradigmatically related if they are generally interchangeable and therefore appear in **similar contexts**, for example *torch* and *sun* which appear in similar contexts and have equivalent grammatical features. Syntagmatic and paradigmatic relations constitute fundamental semantic relationships.

Typical examples of syntagmatic relations on the word level include dependencies between nouns and verbs, compounds, and head-modifier constructions based on adjectives and nouns, or between nouns and nouns. Syntagmatic relations are often responsible for restricting selection from paradigmatic classes on a lower level. Paradigmatic relations vary, depending on the presumed measure of similarity. Paradigmatic relations on the word level range from semantic fields to well defined relations such as hyponymy, cohyponymy, hyperonymy, synonymy and antonymy.

It should be repeated that the notions of syntagmatic and paradigmatic relations do not pertain solely to the word level. By involving the notion of different language levels, it is one of the intentions of the proposed model to generalize the notion of syntagmatic and paradigmatic relations and to apply it to language levels beyond that of the word forms. In this way, algorithms developed for one language level may be more transferable to other levels. However, the same analyses yielding word classes on the word level might yield less varied information on lower levels, such as the distinction between vowels and conconants on the letter level. They maight also be not really feasible, such as a possible distinction between sentence types on the sentence level due to most sentences occurring only once.

The distinction between syntagmatic and paradigmatic relations follows from the primary assumption of structuralism: the value of a certain language element (i.e. atom) exists only due to the existence of other language elements in the same language. The basic relations between various language elements are equalities and inequalities - or better, similarities and dissimilarities. Dissimilarities in a language are expressed by opposition or contrast of elements.

> Nicht dass eines anders ist als das andere ist wesentlich, sondern dass es neben allen anderen und ihnen gegenber steht. Und der ganze Mechanismus der Sprache [...] beruht auf Gegenüberstellungen dieser Art [...]. (de Saussure (2001) p. 145)

To approach the main part of the model, the notion of local context $K_{lc}(a_i)$ (lc stands for *local context*) of a_i is defined for each language level:

Definition

The **local context** $K_{lc}(a_i)$ of a given atom a_i on a given level $a_i \in A_l$ is the set of all atoms a with which the atom a_i occurs together in a complex unit $c_n \in C_l$ on the same level:

$$K_{lc}(a_i) = \{a | a \in c_n \wedge c_n \in C_l \wedge a_i \in c_n \wedge a \neq a_i\} \tag{2.10}$$

The local context is obviously symmetric, because if a word X occurs to the left of another word Y, then the word Y occurs to the right of X. Therefore the following holds:

$$a_i \in K_c(a_j) \Longleftrightarrow a_j \in K_c(a_i) \tag{2.11}$$

Other possible definitions of local context might take the order of the atoms into account or be more specific about the complex units used. Again, the complex unit itself can be conceptualized as a word, a phrase, a sentence or just a fixed size window of words (or other atoms), which also depends on how the language levels are assumed and which level is in question. Generally, all these details can be considered variants of the given general definition.

Since an atom a_i occurs n times in an observation, it has a maximum of n possible contexts. For example, it is important that on the sentence level two sentences differing only in their word order could represent the same context, if the complex units are being treated as sets instead of tuples like in the bag-of-words approach commonly taken in Information Retrieval. Treating complex units as sets instead of tuples is a viable way if the ordering cannot be used and it represents a simplified view on language with a certain information loss.

There are various possibilities of instantiating this notion of local context. These possibilities might be used with different aims in mind. It is now possible to formalize the notion of an abstract syntagmatic relation $SYN(a_i, a_j)$ between two atoms, using the definition of a local context. The notion of syntagmatic relation as used in this context is very unspecific - the existence of such a relation between two atoms does not imply the existence of a linguistically agreed upon relation (between the two atoms) such as a head-modifier relation between an adjective and a noun, although this is likely to correlate.

Definition

A *syntagmatic relation* $SYN(a_i, a_j)$ (which is symmetrical) between two atoms $a_i \in A_k$ and $a_j \in A_l$ holds if a local context for one of the atoms exists in which the other appears:

$$SYN(a_i, a_j) \Longleftrightarrow (a_j \in K_{lc}(a_i)) \tag{2.12}$$

At this stage it is possible that each co-occurrence of a_i with a_j in a complex unit implies its own syntagmatic relation and that they all are truly different relations.

On the other hand, it is also possible that all co-occurrences of a_i and a_j imply the same syntagmatic relation. Additionally, the co-occurrence of a_i with a_j could be due to the same syntagmatic relation (or attribute, see Section 2.4) as the co-occurrence of a_i with another atom a_k. Further, due to differing methods of measuring co-occurrence or defining context, two atoms occurring near each other may not always be considered as co-occurring.

This requires a unification process that is able to decide whether the co-occurrence of the atom a_i with a_j is due to the same syntagmatic relation as another co-occurrence of a_i with a_j or with another atom a_k. This bottom-up unification step is defined in Section 2.4, while another top-down approach to this is introduced along with the definition of the global context below.

On the sentence level, where atoms are concatenations of words, it is possible for a word to co-occur with any other word - in the very least due to possible metalanguage usage such as *Here I use the words drink and atom in one sentence.* Assuming that the number of different sentences, and along with this the cardinality of C in general, is enumerable infinite, this would result in a syntagmatic relation between any two words or atoms. To accommodate this objection, it is possible to recall a comment of Wittgenstein in his Tractatus Logico Philosophicus:

> In order to recognize the symbol in the sign we must consider its *significant use.* (TPL 3.326) Wittgenstein:1921

Assuming that the 'significant use' of an atom is reflected in terms of frequency, there is an expectation value X that reflects the absolute number of joint occurrences that must have been observed in relation to the number of all local contexts for the absolute number to be significant. If the absolute number surpasses the expected value X, then significant use becomes something that can be learned as experience. Thus an observer, be it a human or a learning algorithm, would notice this frequent co-occurrence.

To determine this value X, parameters like frequency of atoms, their distribution, the size of the text corpus and more can be accounted for in a so-called co-occurrence measure, see Chapter 3. Commonly, X is the result of a function such as $X = sig(f(a), f(b), f(a,b), n)$ ($f(x)$ is the frequency of the atom x, $f(x,y)$ the co-occurrence frequency of a and b and n the corpus size). In effect, for any atom a_i belonging to the local contexts of another atom a_j (with $a_i, a_j \in A_l$) it can be decided whether or not $a_i \in K_{lc}(a_j)$ is a *significant* constituent $SIG_X(a_i, a_j)$ of these contexts, based on the number of such contexts and the expectation value X:

$$SIG_X\,(a_i, a_j) \Longleftrightarrow |\{a_i \in K_{lc}(a_j)\}| > X \tag{2.13}$$

The significance is symmetrical as well, thus the following holds:

$$SIG_X(a_i, a_j) \Longleftrightarrow SIG_X(a_j, a_i) \tag{2.14}$$

The absolute number of co-occurrences of a_i and a_j can be the result of different syntagmatic relations, and therefore have different surrounding patterns. Once the co-occurrence occurrences (for example sentences where both words occur) were split into different partitions by means of some method, the split counts of co-occurrences may still be significant.

Given that a number y of occurrences of an atom makes it a *significant* constituent of the context of another atom, the *statistical* syntagmatic relation can be defined:

Definition

A *statistical syntagmatic* relation $SYNS_y(a_i, a_j)$ between two atoms $a_i \in A_l, a_j \in A_k$ holds if and only if the absolute number of occurrences of a given atom (not necessarily all occurrences of that atom) a_i is a *significant* constituent of the contexts of an atom a_j:

$$SYNS_y(a_i, a_j) \Longleftrightarrow SIG_X(a_i, a_j) \tag{2.15}$$

Since significant co-occurrence counts can be split, this allows to define several syntagmatic relations between the two atoms a_i and a_j. Thus, if an atom such as *red* occurs 100 times in a corpus and the expected value X for the pair *red,cross* is $X = 30$, then it is possible that two or three syntagmatic relations exist between these two words. One might result from *red, cross* being a proper name, another might be describing a painting on a wall. However, the statistical syntagmatic relation below is represented only as $SYNS(a_i, a_j)$ omitting the y and its relation to X.

This statistical syntagmatic relation expresses the immediate experience learned from observing a corpus. It also corresponds to Rieger's $\alpha-$abstraction (Rieger, 1989), which is assumed between a_i and a_j if their co-occurrence correlates. While for the local context of two atoms it suffices to consider specific single instances of the complex units of a level, paradigmatic relations require comparing the contexts of atoms of one particular level. For this purpose, the notion of a **global context** is introduced:

Definition

The *global context* $K_G(a_i)$ of an atom a_i of a given level l is the set of all atoms a_j with which the atom a_i stands in a statistical syntagmatic relation $SYNS(a_i, a_j)$:

$$K_G(a_i) = \{a_j | SYNS(a_j, a_i)\} \tag{2.16}$$

This produces at least as many different global contexts as there are atoms. Because a pair of atoms can have two or more different statistical syntagmatic relations between them, this directly translates into the possibility of two or more different global contexts $K_{x,G}(a_i) = \{a_j|SYNS_x(a_j, a_i)\}$ and $K_{y,G}(a_i) = \{a_j|SYNS_y(a_j, a_i)\}$ of an atom a_i, thus enabling to express ambiguity as Chapter 4. In implementations, it is possible to compute the different underlying syntagmatic relations top-down, thus avoiding the implicitly needed unification step[5] of syntagmatic relations when proceeding bottom-up. By dividing a precomputed global context of a word by use of various clustering techniques, the resulting distinct global contexts can be found to represent different underlying syntagmatic relations. It is also possible to utilize differences between various significance measures and the resulting values or rankings, as described in Chapter 6.

Furthermore, it is noteworthy that the existence of several different global contexts for one atom compares well to de Saussure's notion of word centric meaning, where each word is the crossing of several paradigmatic associative chains, i.e. global contexts.

The goal was to enable paradigmatic comparisons of atoms (by comparing their contexts), and the contexts of an atom have been summarized by the notion of the global context. Hence, a comparison operator is needed that takes two global contexts as arguments and returns a similarity value. This can then be used to produce judgments whether they are 'equal' or not.

Definition

The context similarity operator $SIM(K_G(a_i), K_G(a_j))$ is defined as an operation that takes two global contexts $K_G(a_i)$ and $K_G(a_j)$ as arguments and returns a similarity value [0...1], where 1 means equality of the two global contexts.

A threshold t can be assumed which splits similarity values into two discrete values such as $[0, 1]$ with the meaning 'not equal' and 'equal'. This similarity operation results according to (Rieger, 1989) in a semantic hyperstructure (SHS), or semantic space, and corresponds to his $\delta-$abstraction.

Examples of comparison operator instances include known similarity measures such as the Tanimoto measure, the cosine (global contexts can be interpreted as vectors), Euclidian distance, and many more, see also Chapter 3. The question about which one is the 'best' measure cannot be answered easily, because of the difficulty to define a gold standard. It is also important to consider desired results - which paradigmatic relations are to be computed by an instance of a similarity measure. Details of this are discussed extensively in Chapter 3. However, there is no uniform usage of the notion of similarity in the literature. In many instances 'word similarity' is used to refer to a comparison of the global contexts of a given word, whereas sometimes it is also - misleadingly - used to refer the co-occurrence

[5] The unification step is necessary to find out that two seemingly different syntagmatic relations $SYN(a_i, a_j)$ and $SYN(a_k, a_l)$ are in fact the same syntagmatic relation for two different pairs of atoms.

measure of statistical syntagmatic relations, because similar words are also returned by such a computation, cf. (Terra and Clarke, 2003; Dagan, Lee, and Pereira, 1999; Brown et al., 1992; Turney, 2002). In a recent work (Sahlgren, 2006), it was shown in great detail that the type of relations returned by co-occurrence measures and by context comparison measures indeed differ.

An abstract definition of the paradigmatic relation can be given as follows:

Definition

If for two atoms a_i and a_j any of their global contexts $K_{x,G}(a_i)$ and $K_{y,G}(a_j)$ compare to each other with their similarity above a certain threshold t, then and only then a *paradigmatic relation* $PARA(a_i, a_j)$ holds between the two atoms.

$$PARA(a_i, a_j) \Longleftrightarrow SIM(K_G(a_i), K_G(a_j)) > t \tag{2.17}$$

This implies that if more than one pair of global contexts of the two words compare to each other there are several paradigmatic relations between the two atoms. However, only one pair of matching global contexts may also be due to two or more underlying paradigmatic relations. Additionally, the notion of the paradigmatic and syntagmatic relation as used in this work cannot be seen as representing exactly the linguistically (only informally) defined paradigmatic relations. Instead, it represents a simulation and the existence of a paradigmatic relation as defined here likely correlates with the existence of a traditional paradigmatic relation between the two given words (or atoms).

For example, when considering antonyms, it might appear mistaken at this point to stipulate that a paradigmatic relation may only hold if the global contexts of two atoms are similar to each other, since the meanings of two antonyms should be opposite. However, in the case of antonymy, opposition in meaning is expressed by one or just a few opposite values of features that generally are the same for the antonymous expressions, and is not based on a strong difference in contexts. For instance, the words *dim* and *bright* are opposites, but still occur in similar contexts.

This definition derives paradigmatic from syntagmatic relations. From the viewpoint of quantitative linguistics this is substantial, as it justifies the focus on statistical co-occurrence measures and similarity of contexts.

2.3. Linguistic categories

In addition to the principles of composition and abstraction, one of the most fundamental distinctions in linguistics is the notion of linguistic **categories**, distinguishing syntactic word classes on the word level, vocals and consonants on the phoneme level and more. Traditionally, categories are sets of distributional classes, and as such have been described and validated by subjecting samples of a natural language to substitution tests (Grewendorf, Hamm, and Sternefeld, 1989; Grewendorf, 1993).

The exact differentiation degree of linguistic categories varies greatly depending on the language level and the purpose of the categories. On the word level, a basic distinction can be drawn between four main categories of words: nouns (N), verbs (V), adjectives (A) and functional words (S). Though rudimentary, this distinction yields many possibilities with which to distinguish between paradigmatic and syntagmatic relations. For example, Lin (1998a) (and Lin (1998b)) uses categories on the word level in syntactic functions, such as *adj-of* and *subj-of*, to distinguish between semantic relations. Paradigmatic relations are always distribution classes with respect to one category. Hence, on the word level, relations between two nouns (NN) can be syntagmatic and paradigmatic, while relations between a noun and a verb (NV), or an adjective and a noun (AN), can be only syntagmatic.

Categories can be viewed as a generalized form of paradigmatic relations. Word classes, for example, can be expressed by way of equivalence expressions such as 'atom a_i is of the same word class like atom a_j'. However, the global contexts of two words like *sun* and *concept* would barely compare and therefore the paradigmatic relations of 'same-wordclass' would not emerge directly despite them sharing the same syntactic word class. Thus, categories represent generalizations from a number of observations: If a_i and a_j have the same word class as well as a_j and a_k, then it is possible that a_j and a_k have also the same word class. In other words, word class equality is transitive and symmetrical.

Generalizing to classes from paradigmatic relations is not always meaningful, especially in the case of synonymy. In this case, iterating the generalization is likely to lead to the useless statement that all words are synonyms of each other. Currently, the choice when to apply this generalization is mostly guided by intuition (i.e. it is known that word classes are indeed global classes, whereas synonym sets are only local groups for a few words). Further, the choice of methods (which significance measure, or co-occurrence windowing) in order to arrive specifically at word classes or morpheme classes also depends on introspection. Ideally, means must be introduced which automatically recognize the appropriateness of a generalization as well as the correct combination of methods.

Any final category classification $CLASS(a_i)$ can be introduced as a function that maps any atom a_i of a given level l into a set of values. For the word level, the set of values commonly used is drawn from a set of labels such as $\{A, N, V, S\}$. Generally, for any level the set of values can simply be a set of numbers:

$$CLASS(a_i) \longrightarrow \{1, 2, 3, 4, ...\} \tag{2.18}$$

Classes, and similarities in their function, are present on all levels. On the phoneme level, it is the classification of vowels and consonants, as it has been subject to several refinements, see (Trubetzkoy, 1939; Jakobson, 1956), and others.

It is possible to learn the distinction between vowels and consonants from written language in an unsupervized manner. However, further distinctions such as between *labial, plosive* or *fricative* are most likely not acquirable only from written language. This is clearly because these distinctions do not pertain to written language. On the morpheme level there are well known classifications of morphemes into derivational, inflectional and root morphemes. On the word level there are several classifications of lexical categories (for example (Schiller, Teufel, and Thielen, 1995)), whereas on the phrase level word form compositions are classified into noun phrases, verb phrases etc.

At this point, it also becomes clear that any classification on one level is at least partially dependant on the classifications on the lower levels. Finally, it is also possible to classify sentences into simple classes, such as into questions, assertions and exclamations, or into more complex classes, such as introduced by the rhetorical structure theory with nucleus sentences, explanatory sentences etc., see (Marcu, 2000). The higher the level, the more complex the interactions tend to be between the elements of category classes and the underlying classes.

Since categories are based on the most basic paradigmatic relations, other, more spurious paradigmatic relations can be computed by utilizing the less complex ones in a bootstrapping process - Chapter 6 represents a proof-of-concept for this. The global context of an atom (e. g. a word), itself being a set of atoms whose contexts resemble the context of the given atom, will usually contain a variety of classes. For example, filtering this set with reference to the category of a particular atom in focus will divide the set into syntagmatic and paradigmatic relations. Consequently, further definitions of paradigmatic relations can now be expressed as holding if and only if the two atoms belong to the same level, have similar contexts (also including those atoms not in the global context but still having similar contexts), and belong to the same category class.

Definition

An atom a_i stands in a *complex paradigmatic relation* $PARA_{cmp}(a_i, a_j)$ with the atom $a_i \in A_k$, $a_j \in A_l$, if and only if their global contexts $K_G(a_i)$ and $K_G(a_j)$ are similar to each other, and a_i belongs to the same category class as the atom a_j:

$$PARA_{cmp}(a_i, a_j) \Longleftrightarrow \begin{pmatrix} SIM(K_G(a_i), K_G(a_j)) > t \\ \wedge CLASS(a_i) = CLASS(a_j) \end{pmatrix} \tag{2.19}$$

The sentence *The X shines* is an example that helps to illustrate this definition. On the word level, word forms such as *lamp*, *sun* and *surface* would be possible substitutes for X. Alternatively, on the phrase level phrases such as *rising morning sun*, or *old and expensive desk lamp* might be appropriate. All of these atoms have in common that all of them *shine*. The implications of this finding for paradigmatic relations are detailed in the next paragraphs.

First, the problem of how a category class might be obtained automatically must be discussed (focussing on syntactic word classes only, for simplicity's sake), since it appears to contradict one of the initial aims of this work (describe a model that is at any point fully computable). Despite existing work on this matter (Clark, 2003; Freitag, 2004; Biemann, 2006b), the problem of obtaining word classes (at least) of sufficient quality is currently considered as mostly unsolvable, at least not without supervision (as well as any other category classification on other levels). However, when allowing to utilize a manually produced training set, there are several well known ways to obtain word classes (based on Hidden Markov Models and supervised learning) (Brill, 1992; Cutting et al., 1992; Brants, 2000).

A fully unsupervised word class tagger would require a different approach. First, for the present purposes it is not necessary that the $CLASS(a_i)$ - function uses a mapping onto a set of tags $\{A, N, V, S\}$, because a set of numbers would suffice. As is obvious from Definition 2.19, it is only important to know whether the category of two atoms equal. From the definition of the classes follows that all atoms of the same class share similarity pairwise in how they are used in complex forms, and at the same time contrast against other classes. This can be viewed as a clustering task, where the units to be clustered are atoms, and the features of these atoms are their global contexts (e.g. words from the word level), see also Biemann (2006b) for quite an exact implementation of this hypothesis.

2.4. Compliance and attributes

After exploring the atoms and their relationships, it is now necessary to examine the mechanics of complex units which are built from atoms. These are interconnected with the selection processes responsible for particular atoms to co-occur in complex units significantly. First, the observed complex units are assumed meaningful[6]. Note, that contrary to truth-functional semantics this model is not centered around the decision whether or not a sentence is true. Instead, this model provides means to compute whether it seems meaningful or meaningless (or easy for perception versus hard for perception, or plausible vs. implausible) with respect to prior experience or knowledge represented by a sufficiently large sample of texts.

The combinations of complex units from atoms on a certain level utilize implicit rules that obey what can be called *compliances* on the syntagmatic level, resembling the compliance in traditional grammar theories, such as Head-Driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1994; Cooper, 1996). Which particular atoms appear in a given combination depends on a selection process involv-

[6] Usually in implementations complex units with a frequency higher than a given threshold are assumed as meaningful, in order to discard various kinds of error sources that only add noise.

ing compliances on the paradigmatic level. Some compliances (sometimes termed agreements) might involve higher levels - for example, whether some morphemes are combined into *done* or *doing* on the word level depends on the selection at the sentence level. This can be due the presumedly existing attribute 'time' having been assigned a particular value in the sentence in which the lemma *do* is used.

2.4.1. Syntactic categories

To explore how semantic relations could be learned by algorithms, it is helpful to have a look at a lower level, namely the morphological. Semantic attributes, such as *direction* or *shiny*, can be seen as similar to morphologic categories such as declinational suffixes, person, gender or number on a lower level. The primary function of these categories is to explain morpho-syntactic compliance on a morphological and phrase/sentence level. In fact, de Saussure describes syntagmatic and paradigmatic relations through examples from morphology (de Saussure, 2001). The categories of person, gender, or number follow a syntagmatic pattern, which is combinable into tables. Thus, as examplified in Table 2.4.1 from German, the pronouns *ich, du, er* are syntagmatically related to verb endings in present tense singular with respect to the secondary category person.

person	*pronoun*	*verb*
1	ich	geh+e
2	du	geh+st
3	er	geh+t

Table 2.1.: An example of syntagmatic agreement on the morphological and phrase level for the person attribute.

One way (among many) to see this is by making use of a grammatical theory, such as the HPSG (Pollard and Sag, 1994; Cooper, 1996) or Lexical Functional Grammar (LFG) (Kaplan and Bresnan, 1982). According to HPSG there is a compliance of the attribute person (PER) on the values *1st*, *2nd* or *3rd*. However, this approach differs to the one given in this work: HPSG representations use feature structures to represent grammar principles, rules and lexical entries. Here, the primary goal is not to produce representations of sentences, but to be able to learn their possible attributes or features. Eventually, this makes it feasible to automatically produce sentence representations according to the automatically learned knowledge. However, these representations will not be immediately human readable, but they could be turned readable by assigning the learned attributes common names such as PER. One goal for such automatically learned attributes is to enable applications to handle language with an understanding that goes beyond simple strings and a strict vector space built from them.

Other differences to grammatical theories such as the HPSG include the fact that in the presented model there are no types and no principal difference between semantic and syntactic attributes. The only difference between semantic and syntactic attributes could be the fact that grammatical attributes always map into discrete values such as *1st* or *2nd*, whereas semantic attributes usually (but not always) map into a value weighted by an applicability strength (see below). This applicability value is to be understood abstractly - neither have normalization methods been specified, nor can they be simply taken as probabilities. The exact interpretation depends on specific realizations of the model.

In fact, it could be hypothesized, that grammatical attributes were former semantic attributes that through very frequent usage became canonized in a manner which allows for a fixed set of values with significantly less fuzzyness and ambiguity. The fact that even the exact number of tags in a tagset for syntactic categories is frequently subject to disagreement supports this hypothesis. In further support of this hypothesis, it is possible to compare the presence or absence of attributes across various languages, such as the famous example of Hopi, which has no time attribute. In such languages, the missing grammatical attributes are expressed via semantic attributes.

The existence of an attribute $p_x(a) = 0, 1$ can be defined as follows:

Definition

From syntagmatic co-occurrence regularities of an atom a_i with an atom a_j and the therefore existing statistic syntagmatic relation between them $SYNS(a_i, a_j)$, it is possible to hypothesize the existence of a trivial attribute $p_{ij}(a) \in P : p_{ij}(a) \longrightarrow \{0, 1\}$. This attribute is an element of the set of all possible attributes P, takes the two atoms a_i and a_j and maps both into the same value of 1 and 0 for all other atoms.

$$p_{ij}(a) = \left\{ \begin{array}{l} 1 : a|\,(a = a_i \lor a = a_j) \land SYNS(a_i, a_j) \\ 0 : otherwise \end{array} \right\} \tag{2.20}$$

If necessary, it is possible to write $p_{ij}(a)$ or $p_x(a)$ instead of just $p(a)$ in order to differentiate between various found attributes.

A syntactic compliance $CPSYN_p(a_i, a_j)$ of two atoms a_i and a_j is given if there is an attribute p yielding the same value for both atoms:

$$CPSYN_p(a_i, a_j) \Longleftrightarrow (p_x(a_i) = p_x(a_j) \neq 0) \tag{2.21}$$

The initial approach of implying compliance from syntagmatic relations would result in a multitude of different possible attributes for different co-occurrence pairs that all result in a single value:

$$\begin{array}{l} p_{ij}(a_i) = p_{ij}(a_j) = 1 \\ p_{kl}(a_k) = p_{kl}(a_l) = 1 \end{array} \tag{2.22}$$

Therefore it is important to provide a mechanism allowing for the unification of two seemingly different attributes p_{ij} and p_{kl} into a single p_z one that maps into two different values $p_z(a_i) = 1$ and $p_z(a_k) = 2$. At the same time it should not unify attributes which do not belong together. Generally, if the two atoms a_i and a_k stand in a paradigmatic relation $PARA(a_i, a_k)$, and the other two atoms a_j and a_l in another paradigmatic relation $PARA(a_j, a_l)$ then it is *possible* that the attributes p_x and p_y represent the same attribute with varying values. To verify the new hypothesis, further tests are needed such as measuring the inhibition between either a_i with a_l or a_k with a_j. Also, possible ambiguities causing the atoms to express several different attributes need to be accounted for.

$$p_z(a) = \left\{ \begin{array}{l} 1 : p_{ij}(a) = 1 \wedge PARA(a_i, a_k) \wedge PARA(a_j, a_l) \\ 2 : p_{kl}(a) = 1 \wedge PARA(a_i, a_k) \wedge PARA(a_j, a_l) \\ 0 : otherwise \end{array} \right\} \tag{2.23}$$

For example, due to the compliance restriction resulting from the attribute PER it is less likely to encounter a combination such as *Ich gehst* as opposed to the three correct combinations in a German corpus. However, using pure frequency counts might lead to mistaken conclusions, since such apparently wrong usages may still occur due to other reasons, such as metalanguage usage, complex syntactic constructions, mistakes or dialectical usage. Also, it is not common for each value assignment of an attribute to have its own unique representation, be it on the morphological or the sentence level. Especially in the isolative English, for example, most value assignments for attributes result only in a small set of distinct representations, and many result in the zero-representation.

In fact, in the above example, three things co-occur with each other: *Ich*, *geh*, *+e*. Thus it suffices to observe that the atom *geh* co-occurs with either *+e*, *+st* or *+t* on the morpheme level and that neither of the three suffixes co-occur together to know that they stand in a paradigmatic relation with each other (because co-occurring with *geh*) and in the same syntagmatic relation with *geh*. The next step would consist of generalizing to classes in order to know that instead of specifically *geh* these suffixes generally co-occur with all atoms of the same kind as *geh* and constitute an attribute. However, to satisfy the requirement of having a minimum amount of different attributes describing a maximum amount of dependencies and effects observed, it would be necessary to broaden the observation so as to include the morphemes and words *Ich*, *Du* and *Er* into the attribute.

Another unification operation is needed allowing for the unification of two seemingly different attributes p_{ij} and p_{ik}, for cases such as *Ich gehe*, *Ich arbeite* and *Ich renne*[7]. In other words, where contrary to the unification above one atom out of the two is the same for both attributes.

Thus, if two attributes p_{ij} and p_{ik} exist, they can be unified into one single p_z, which maps into the same value (contrary to the two different values above)

[7] *I go*, *I work* and *I run*

$p_z(a_i) = 1$ and $p_z(a_j) = 1$ and $p_z(a_k) = 1$ if and only if there is a paradigmatic relation between a_j and a_k.

$$p_z(a) = \left\{ \begin{array}{l} 1 : \ (p_{ij}(a) = 1 \vee p_{ik}(a) = 1) \wedge PARA(a_j, a_k) \\ 0 : \ otherwise \end{array} \right\} \tag{2.24}$$

This implicitly defines a bootstrapping process on a bipartite graph that iteratively finds all words for which a given attribute such as 'number' is applicable. For example, from *Er geht* and *Er rennt* such an algorithm would learn that *gehe* and *renne* share the same attribute. From *Sie geht* and *Sie rennt* it would learn that still the same attribute can be applied to all the four words *Er,Sie,rennt,geht*, while yielding the same value. A similar bootstrapping process has been employed to learn proper names automatically from a given corpus (Biemann et al., 2003). The unification mechanism introduced here can be understood as a highly generalized abstraction of that bootstrapping process.

As defined, for an algorithm to be able to learn such kinds of attributes, it is possible to observe two kinds of dependencies in a corpus:

1. **Syntagmatic:** language units representing compliance due to an assumed attribute, such as words or morphemes which either attract or inhibit co-occurrences. The example in Table 2.4.1 illustrates that the word *ich* attracts the morpheme *+e* in the corresponding verb and inhibits the morpheme *+st*.

2. **Paradigmatic:** language units representing the assumed attribute are not easily interchangeable, despite belonging to the same paradigmatic class. Furthermore, various representations of an attribute belonging to the same paradigmatic class are mutually exclusive, implying that co-occurrences of these, such as the direct co-occurrence of *Ich* with *Du*, are far less probable and mainly confined to special language usages.

The attributes can have interfering or overlapping influence on the construction of the complex unit in question. For example, both $PER(er) = 2$ and $PER(sie) = 2$ with the difference that the attribute *gender* is different: $GENDER(er) = 1$ and $GENDER(sie) = 2$. The amount of attributes ultimately acquired should (ideally) contain just the amount of attributes needed to explain all observable inhibitions and dependencies (hence be able to predict them). Furthermore, it is not important for the system to learn the names or labels of the attributes, as long as the results are to be used in other algorithms or applications. Thus, $NUMBER(er) = 1$ is equally useful as $CLASS1(er) = 1$.

Of course, to be able to learn attributes such as those provided in the examples, it is necessary to be able to distinguish the relevant units in the selection process. For the given examples of number and attribute this means that both word or possibly even phrase boundaries within a sentence, but also morpheme boundaries within word forms, must be known. The appearance of a given morpheme usually depends on more than one factor, therefore it is necessary to be able to distinguish between the usages by means of a disambiguation algorithm. Using these or similar methods should make it possible to explore the realizations of one attribute, such as the PER. Furthermore, such compliances function to restrict the possible values that a certain realization can assume. Therefore, co-occurrence counts will be biased towards the 'correct' usage of language, as opposed to wrong usages.

To summarize, in the model presented in this chapter only the existence of such attributes was described. No attemps were made to describe all possible such attributes. Additionally, an abstract way of learning such attributes was defined. The idea is that the algorithm, which learns the structure of a given language, should be able to distinguish the attributes by itself, through dividing the various influences and dependencies into classes and unifying the attributes where possible. As long as there are trivial attributes that map only into single values, the unification process is not completed. Ideally, all such attributes should be learned automatically by one abstract learning algorithm. Currently, however, in the field of deep lexical acquisition, it is considered to be an achievement to construct an algorithm that would reliably learn all possible representations of one particular attribute such as person or gender.

2.4.2. Semantic attributes

While on the phrase level syntactic compliance can be described by attributes with labels such as 'person' or 'number', it is often unclear how to grasp the notion of semantic compliance present on both the phrase and sentence level. Semantic compliance is observable in examples such as *I see the green frog* that stand versus examples such as *I run the thick frog.* While the former sentence sounds correct, the latter sounds utterly senseless in the absence of an appropriate context. Consequently, coherence of the latter is dependant upon the presence of an understood/proper context, whereas the former example remains almost always coherent. In both cases, the difference in correctness between the given sentences would be the background as given by a corpus: The first sentence is coherent if a normal corpus is to be assumed against which it is interpreted. The second sentence is also coherent if a corpus is assumed in which both *thick frogs* exist and that they can be *run.*

The ability to compute such coherence of sentences does depend upon an understanding of the world in which the corpus was built - it suffices to test the sentence against the corpus. It is therefore necessary to generalize the notion of attributes to include semantic attributes as well. The generalization can be performed by viewing the morpho-syntactic attributes as mapping not only into value sets, but rather, into distributions. Thus, the attribute person would map the atom *you* into the distribution $1 : 0.0, 2 : 1.0, 3 : 0.0$. It follows that the applicability of the first and third value is 0.0. The sole applicable value is the second. For the semantic attributes only one value would be applicable, but with fuzzy applicability weights or values. Further in the initial stage, each word form would give rise to its own semantic attribute. Hence the attribute $frog(green) = 0.6$, which is shorthand for the fact that the attribute 'frog' maps the atom 'green' into the sole value 1 with an applicability value of 0.6 (hence, this notation is equivalent to writing $PER(er) = 2$). This attribute would map the majority of other words like 'thick' into the sole value of 1 with an applicability value of 0.0.

The semantic attribute $q \in Q$, an element of all possible semantic attributes Q, could thus be expressed by paradigmatic relations, resembling the way syntactic attributes were expressed by syntagmatic relations. From the existence of a paradigmatic relation $PARA_w(a_i, a_k)$ (based on the similarity measure yielding a similarity value of $w > t$) between the atoms a_i and a_k it is possible to hypothesize the existence of a semantic attribute $q_k(a)$ that takes atoms and returns a value with an applicability weight of w greater than 0:

$$q_k(a) = \left\{ \begin{array}{l} w : PARA_w(a, a_k) \\ 0 : otherwise \end{array} \right\} \tag{2.25}$$

This semantic attribute may apply to only a single atoms pair, but may also apply to a large number of atoms. A semantic compliance $CPSEM_q(a_i, a_j)$ of two atoms a_i and a_j is given if there is a semantic attribute q such that it yields an applicability value of greater than 0.0 (or a given threshold t):

$$CPSEM_q(a_i, a_j) \Longleftrightarrow [q(a_i) > 0 \wedge q(a_j) > 0] \tag{2.26}$$

Thus, it is a way of saying that a_i and a_j comply (agree) semantically due to q (which is different from being only able to say that a_i and a_j stand in a paradigmatic relation $PARA_w(a_i, a_j)$. In order to obtain the initial semantic attributes for atoms of a given level l, it is possible to employ the co-occurrences on the same level within the complex units, for example. On the word level this means the sentence or neighbour co-occurrences of words, see Chapter 3. Thus, a word like *frog* could give rise to an attribute that maps either *green* and *jump* or, depending on the corpus used *dissect* and *Kermit* to applicability values other than 0. On the other hand, *green* and *frog* co-occur with several other words such as *living*, *wet*, *grass* and *water*, resulting in several semantic attributes giving rise to a compliance in the sentence *I see the green frog.*

A globally ambiguous word in a text will commonly be used with a meaning that complies with the meaning of that particular text. It is, of course, possible to use both (or more) meanings of an ambiguous word in a single text, but the meaning of a particular occurrence will usually be clear from its surrounding context. This explains the possibility to disambiguate words automatically by using context extracted from the surrounding text, as is evidenced by the extensive literature on this topic (Lesk, 1986; Sanderson, 1994) or (Banarjee and Pedersen, 2002).

As with word classes and grammatical attributes, it is possible to utilize a currently unspecified mechanism to generalize specific attributes such as ‘green’, ‘red’ and ‘blue’ into a more abstract single attribute, which a human observer might name ‘color’, while still retaining the initial attributes. Note, however, that the attribute ‘color’ would differ from the attribute that emerges from the word form *color*. Although it might be an obvious idea to unify those two, it is not necessarily a good one, because there is an intuitive difference between using a specific color and using the word form *color*. For example, while saying *I see the green frog* seems plausible, it either is less plausible to say *I see the color green frog* (or *I see the frog with the green color.*), or the meaning of these sentences actally differs from the first one slightly. The Latent Semantic Analysis (Deerwester et al., 1990; Dumais, 1995) could be a mechanism making such generalizations on a given corpus, however, this has to be investigated further. Additionally, the general mechanism of abstraction on the semantic level has yet to be specified.

Similarly as in morphology, the exact name and value of the generalized semantic categories may not be important. Furthermore, contrary to the sharp morpho-syntactic categories, semantic categories tend to be fuzzy, cf. (Rieger, 1991). This also implies that while inhibition and attraction for the morpho-syntactic attributes is clear-cut and therefore mostly easy to extract, semantic attributes may be more problematic (or require more complex algorithms).

Besides obvious semantic categories which a human observer would name *location* or *property*, for example, there are also less obvious ones such as *semantic orientation* (positive or negative) of words. Hatzivassiloglou and McKeown (1997) compute the positive or negative orientation of adjectives, exploiting the fact that conjoined adjectives typically have the same orientation in a small training set. Turney (2002) computes *orientation* by using the web search engine Altavista to obtain the occurrences of all words near positive or negative words (i.e. a training set as well). Though such algorithms use a manually created training set, and thus do not fit in the intended paradigm of fully unsupervised algorithms, they will gain importance once other algorithms become available that generate a small set of proposals for positive/negative algorithms. These first experiements also show a need for a generalized approach to the automatic extraction of semantic attributes.

The algorithms in Chapter 6 are highly simplified versions of parts of the hitherto modelled learning processes. They do not use explicit unification or explicit class representation. What they show however, is simply the concept of filtering the results of one method against other influences on the type of relations found. As a proof-of-concept these algorithms support some of the basic assumptions of the SIML. The construction of a full learning process according to the presented specifications is beyond the scope of a single work. However, attempting this would certainly reveal necessary corrections to the model.

2.4.3. Computing coherence

An example of how the previously defined notions can be applied, such that both the syntactic and semantic attributes are used, is to count the complying pairs of atoms (a_i, a_j) and the number of attributes p in which they comply for a given complex unit $c \in C_l$. This number can then be compared with the number of attributes which apply to single atoms of the complex unit (for the semantic attributes) or map to different values (for the syntactic attributes). This comparison can be used as a simulation of coherence, because the more complying syntactic attributes are present, the more grammatically correct will the sentence appear (in the strict sense that a single non-complying attribute leads to the sentence to be considered wrong). Additionally, the more semantic complying attributes are found compared to non-complying, the more familiar - or coherent with respect to the corpus used it appears.

Beginning with the given level, this kind of coherence can be computed for all lower levels of complex units. Thus, if a sentence is given, it is possible to separately compute the coherences of the phrases, the words and morphemes. Of course, the significance of coherence differs for each level. A morphologically incoherent word will be regarded as a non-word, while a syntactically incoherent but semantically coherent sentence will be seen as merely grammatically wrong, but still understandable. A highly simplified example of the coherence notion is a spell checking program, only producing binary decisions and only taking the word level into account. A less simplified example is a statistical POS tagger, which utilizes both the word level and a simplified version of the phrase level.

Returning to the example sentence *I see the green frog*, for the word *I* there are many sentence co-occurrences that can be translated into attributes such as $my(I) = 1.0$, $think(I) = 0.9$ and $see(I) = 0.6$. For the word *see*, besides the common attributes such as $why(see) = 0.7$ or $whether(see) = 0.3$, the abstract semantic category ‘concrete’ (vs. abstract) might play an important role in bringing together the sentence, because both for *see* and *frog* this category would map into a non-zero applicability value. However, while the basic semantic categories are drawn from an existing English corpus, they have only been hypothesized as currently no learning algorithm capable of such generalizations as described in the previous chapter exists, unless algorithms such as Latent Semantic Analysis (LSA) (Dumais, 1995) are accepted as producing such abstract categories.

Compliance also means that in a text mostly words complying with each other occur, since otherwise the text would become meaningless - or too difficult to read and understand. This might be expressed as coherence - the more compliance, the more coherent the text; but more compliance also implies less new information conveyed by a text of the same length. On the one extreme is a sentence already existing in the corpus, hence no new information. The opposing extreme is a sentence consisting of words not present in the corpus at all. Such a sentence is entirely new, but from the perspective of the corpus too different. Each word should have been introduced using phrases or expressions from the corpus. Thus, conveyance of new information must always go along with sufficient coherence in order to be understandable. Hence, the process of authoring a text always involves a trade-off between seemingly redundant compliant (or coherent) usages of words and the actual conveyance of new information.

The general coherence of a sentence (or rather complex unit) can then be abstractly defined through compliance on both syntactic and semantic attributes, as well as categories.

Definition

The syntactic coherence $COH_{syn}(c)$ of a complex unit $c =< a_1, a_2, ..., a_n >$ is defined as the number of complying syntactic attributes $p \in P$ versus the number of attributes that apply to one atom of the complex unit yielding a certain value without also applying to other atoms with the same value. Hence, for all i and j:

$$COH_{syn}(c) = \frac{|\{p|CPSYN_p(a_i, a_j)\}|}{|\{p|CPSYN_p(a_i, a_j)\}| + |\{p|p(a_i) \neq p(a_j))\}|} \tag{2.27}$$

A stronger version would account only for unmatching attributes in the divisor that yield values other then 0 for more than one word but still not match. Using this method directly enables to identify either missing attribute unifications or syntactic errors in the complex unit (if the attribute unifications are assumed as complete).

Definition

The semantic coherence $COH_{sem}(c)$ of a complex unit $c =< a_1, a_2, ..., a_n >$ is defined as the number of complying semantic attributes $q \in Q$ versus the number of attributes that apply to one atom of the complex unit yielding a value larger then 0.0 without applying to at least one other atom as well. For all i and j the semantic coherence is then

$$COH_{sem}(c) = \frac{|\{p|CPSEM_p(a_i, a_j)\}|}{|\{q|CPSEM_q(a_i, a_j)\}| + |\{q|q(a_i) \neq 0 \land q(a_j) = 0)\}|} \tag{2.28}$$

A specific normalization factor would have to be included in an implementation of this definition, which takes the large number of semantic attributes generated by high frequent words into account. Since the word *the* is very frequent and its global context would have at least a small similarity (unequal zero) to most other words, it would give rise to many primary semantic attributes with a small probability for a larger number of them to be satisfied in a given sentence. However, the great amount of semantic attributes with small similarity values strongly indicates a low specificness of the word, which makes the fact that most semantic attributes raised by this word are not compliant irrelevant. However, if for a very specific word the only three available semantic attributes comply in a sentence, this is significant information, compared to the possibility of them complying.

Definition
The combined syntactic and semantic coherence $COH(c)$ of a complex unit $c =< a_1, a_2, ..., a_n >$ can now be defined as the number of complying (syntactic and semantic) attributes p versus the number of attributes that apply to each atom of the complex unit without also applying to other atoms with an unspecified normalization factor γ.

$$COH(c) = \frac{\gamma COH_{syn}(c) + (1 - \gamma) \, COH_{sem}(c)}{2} \tag{2.29}$$

Instead of averaging the syntactic and semantic coherence factors, it would also be possible to add them or produce a weighted average. The exact implementation of a coherence measure can differ from the one presented, but the essential part of comparing complying with non-complying units will remain.

The coherence measure can be used in various applications such as grammar and spellchecking, but also in more complex ones, including text summarization and abstracting. For example, removing paragraphs or sentences with the highest semantic coherence would leave more interesting ones - those containing words from different topics or usages - probably sentences conveying entirely new information. From the selected sentences it would also be possible to select words which, when removed, would result in the least semantic coherence loss, thus shortening the sentences. It would then even be possible to select phrases and words, where replacing the former with the latter incurs a minimal coherence loss. This should result in replacing phrases with more abstract words, since they would still fit into the remaining sentences of a paragraph in order to keep the loss of coherence minimal. When not tolerating a loss of syntactic coherence, grammatically correct abstracts would be produced by such an envisioned system.

2.5. Semantic primitives and relations

In addition to semantic categories there are a number of seemingly problematic cases with the model as presented so far. The inherent fuzziness and lack of labels in the results[8] appears to inhibit a proper description of semantic primitives which either are present in the meaning of a word or not. However, the abstraction mechanism mentioned in Section 2.4.2, implemented as a clustering algorithm, provides

[8] Since the results of each step are abstract collections of features or distributions without labels.

abstract attributes that can be seen as equivalent to the semantic primitives except that these abstract attributes do not automatically receive names similar to those that can be found in manually created semantic primitive collections.

In order to clarify this analogy it is feasible to make an example for the word *leap*. Apart from the initial attributes like 'frog' and 'leg' that will emerge from an initial analysis of a corpus where *leap* frequently occurs, other abstract attributes could emerge from clustering. The would result in *leap* becoming part of one or more clusters resembling abstract attributes. The first one, for example, could be a cluster built from the words *leap, jump, run* and *go*. This cluster could thus be seen as representing the semantic primitive **move**. The second cluster could comprise the words *leap, big, huge, gigantic* and therefore represent the semantic primitive **large**. A third cluster could possibly be built from the words *leap, sudden, fast, jerky* and represent the primitive **sudden**. The 'meaning' of the word *leap* could then be represented either by the semantic primitives **large sudden move**, which is only interesting for humans, or by the three clusters representing the attributes, which is something applications can make use of.

The fuzzyness is another seemingly problematic feature of semantic primitives, because they are commonly defined to either apply (eventually positively versus negatively) or not, without fuzzyness. The meaning of the word *frog* therefore clearly possesses the primitive *+living*. However, as described in Section 1.3, this model aims at being able to analyze the *structure* of language, not the relation between parts of the structure and the real world. Also, whether *+living* is correct for the word *frog* depends on its usage - it might have been used as a name or a concept for something. In these cases, *+living* might still be slightly applicable, but not as clearly as before.

Further phenomena commonly encountered are linguistic relations such as antonymy, synonymy and hypernomy, which are not dependent upon the values of specific semantic categories. These abstract semantic relations are important for many NLP applications, and traditionally have been manually encoded in lexical-semantic structures such as WordNet (Miller, 1990; Fellbaum, 1998), GermaNet (Hamp and Feldweg, 1997; Kunze and Wagner, 1999), and EuroWordNet (Vossen, 1998). One important application of these resources is to infer other relations, cf. (Richardson, 1997).

In WordNet, the collection of semantic relations includes:

- *Hyperonymy* and *hyponymy* also sometimes called the is-a-relation. It holds between two 'concepts' (which are sets of words) whenever one has a more abstract meaning than the other.
- Two words are *meronyms*, whenever one denotes something that is part of the other word's denotation. In WordNet, meronyms are split into several types: part-of, member-of and substance. The differences between these types depend on the type of the denotation (countable, fluid, etc.).

- *Synonymy* means exchangeability within a sentence without changing the meaning of the sentence. In the logical tradition, true synonymy means exchangeability in all sentences of a language (i.e. the global context of a word). However, this is very rare, because it works against the economy of language (to use as few different tokens as possible). Here, synonymy means exchangeability in local contexts.

- *Antonymy* describes all kinds of oppositions.

- *Derivation* according to WordNet, covers all cases where one word is morphologically derived from another word; often this implies that the two words belong to different word classes. Derivation is usually viewed as a syntagmatic relation.

For some reason, *cohyponymy* has not been included in either WordNet or GermaNet. Deriving this relation from hyperonymy is possible, but sometimes yields errors, because hyperonymy has not been coded consistently. In GermaNet the authors have tried to resolve this by introducing artificial nodes of non-lexicalized concepts. However, this technique has not been used consistently enough to prevent all inconsistencies. Furthermore, new irregularities occur when artificial nodes are created.

In order to arrive at such relations using the framework presented in this chapter, it is necessary to first realize that they still resemble basic paradigmatic relations. However, using the method for finding them as outlined at the end of Section 2.2, the result would probably be a large variety of paradigmatic relations (each representing its own group of synonyms, for example). In theory then, a unification step is required which would detect that many of the found paradigmatic relations are actually the same relation for different atoms. On the other hand, current learning systems will be guided by introspection based hypotheses and thus be designed a priori in a way as to extract only the one or several desired relations. With this approach however, the possibility of entirely new paradigmatic relations will remain undetected. Without introspection it is possible to measure abstract properties such as symmetry, antisymmetry, transitivity etc. and take differences or similarities with respect to the results as unification cues.

Using the described framework, it is now possible to give algorithmic descriptions for their extraction from a given corpus. For example, the hyperonymy relation between two words holds, if several conditions co-occur. One condition is that both words must belong to the same syntactic word class. Also, the global contexts of both words must compare well to each other, because the one word is only more general than the other, thus it would appear in similar contexts. However, the co-occurrence frequency would be low (but not necessarily zero), because while it is uncommon to additionally mention the hyperonym of a word, it still happens in sentences like *We have that huge animal of an elephant in our zoo!*. More examples and specific algorithmic solutions of such abstract considerations are given in Chapter 6.

2.6. Conclusions

To recollect, several concepts are formalized in this chapter. From the trivial observation of language levels, the measuring of co-occurrences is introduced as representing local contexts. The difference between local and global contexts of language units is defined analogously to the difference between type and token. The global contexts, in unison with a statistical significance threshold, summarize the general usage of a given language unit and are subsequently used to simulate syntagmatic relations between pairs of language units. The set of all units standing in a syntagmatic relation with a specific given unit, describing its ‘meaning’ (in the sense of usage), can be used to effectively compare language units with each other ‘semantically’. These pseudo-semantic comparisons are then used to redefine a simulation of paradigmatic relations holding between two units if the comparison of their sets (of units in a syntagmatic relation) results in a high similarity.

By utilizing the introduced syntagmatic and paradigmatic relations, syntactic attributes and compliance (agreement) based on them are defined. A system of these attributes can be used to describe the syntactic structure of a complex language unit, such as a sentence. The introduced semantic attributes can further be used to describe the semantic structure of a sentence. Finally, the notion of coherence is exemplified as expressible with the attributes and used to measure how well a complex language unit fits the experience of a given language as previously learned from a raw corpus. By providing precise mathematical models for each concept introduced, the entire model can be instantiated into a framework or learning system.

The relatively simple instruments defined in this chapter allow for very complex descriptions of language structure while also allowing for a complete translation into algorithms. Since all instruments introduced in this chapter are based on co-occurrence observations, the entire model is suited to construct a fully automatic and unsupervised learning system capable of deriving structural descriptions from a raw text corpus of a given language, if the corpus is large enough. However, the knowledge of exactly which measure to apply at which step of the learning process is very complex and currently supposed to be introspection driven. Possibilities to avoid introspection at this point are discussed. Presently, such a complete system does not exist. One reason is that most pieces of this hypothesized system are unknown, and another is that most current algorithms are developed with only one particular goal with barely any interconnections to other algorithms. Once such a system exists, it will be significantly more language independent due to the high abstraction level of the employed methods.

One of the main points of this model - that the basic principles are the same for all language levels - was respected. All of the introduced formalisms are defined independently of the level in question.

While the examples given throughout the chapter suggest a purely co-occurrence driven approach, other approaches such as pattern-based learning fit into this model without problems. Patterns in that case are a slightly altered perception of what a global context of a given language unit, i.e. word, is. Thus, following a learning step, whose output is a set of typical patterns, it can be decided for each word, whether it commonly appears in conjunction with a particular pattern. Such a pattern, in turn, can be considered as part of the global contexts of that word.

The next chapters each summarize a group of different algorithms as described throughout the literature in the recent decades into a class of algorithms operating on the same language level and with similar goals. This grouping helps to establish their place in the model described in this chapter. It also helps to see where the shortcomings of current approaches are located, as seen from the perspective of the introduced model, and where improvements are possible. These improvements are described as in-depth as possible, while remaining within the breadth of this work, as well as being evaluated and compared with some of the existing algorithms.

3. The Word Level

3.1. Introduction

This chapter explores the possibilities of a practical implementation of the theoretical foundations introduced in the preceding chapter. One branch of natural language processing is concerned with the automatic extraction of lexical relations between words by means of statistical methods, usually measures of statistical co-occurrence. This partially fits to the distinction made between syntagmatic and paradigmatic relations. This research covers several apparently different topics, such as lexical acquisition, document clustering, computing semantical word similarity, word associations, synonyms, antonyms, idiosyncratic collocations, electronic thesaurus, semantic nets, etc. Throughout this chapter, they are referred to as **extraction of relations between words**, even though they are not extracted explicitly in most applications. Nevertheless, applications such as Information Retrieval, disambiguation algorithms, speech recognition, or spellcheckers (may) profit from information about contextual similarity between words, morphemes, word groups and even sentences. Although there appears to be a large variety of methods and goals (often resulting in a vague and imprecise terminology), after classifying the methods according to the model, the whole topic presents itself fairly coherently.

The reason for such an immense variety of algorithms whose main subjects are words lies in the accessibility - in most languages observing word occurence patterns within sentences is straightforward and requires nearly no additional preprocessing as compared to morphemes, for example. At the same time, using words as the research object is quite rewarding - results from very constrained algorithms can immediately be used in well known applications, usually increasing their performance. It is also crucial that the space of cognitive concepts lends itself to be mapped into a set of words where each one represents a concept. Thus, programs able to compute unspecified relations (commonly called associations) between these 'concepts' demonstrate an apparent 'understanding' of the world, resulting in limited success without having actually provided an adequate or useful modeling of meaning. Furthermore, the variety of phenomena to extract or handle for machine learning algorithms ranges from such vague topics as the extraction of idiomatic expressions to more specific topics such as the extraction of antonyms or tagging of syntactic word classes. This led to the fact that researchers from very different fields (including mathematics, linguistics, lexicography, media science or computer science) found themselves trying to produce corresponding algorithms.

The purposes of this chapter are:

- to outline current research in this area with respect to the model introduced in Chapter 2;
- to review the most commonly employed methodology foundations and to exemplify their similarities and differences (Sections 3.2 and 3.3);
- to propose a suitable evaluation method for comparing the various co-occurrence and similarity measures (based on earlier work (Bordag, Witschel, and Wittig, 2005)) and to reveal specific effects various parameters have on these methods (Section 3.4).

In addition to these three aims, an attempt is made to compare and organize the terms used by different authors.

3.1.1. Extraction of collocations

There are several historical motivations for computing relations between words. Firth (1957), besides stating that meaning and context should be central in linguistics, introduced the notion of collocation on the lexical level and defined it as the consistent co-occurrence of a word pair within a given (syntactical) context. Since the appearance of Firth's paper, the notion of collocation was further developed. Nowadays there is a dichotomy between grammatical and lexical collocations, although other possible divisions have been described by Smadja (1993). An informal definition of a grammatical collocation is given by Benson, Benson, and Ilson (1986): "A grammatical collocation is a phrase consisting of a dominant word (noun, adjective, verb) and a preposition or grammatical structure such as an infinitive or a clause." Examples include 'account for', 'adjacent to', 'an oath that' etc. On the other hand, lexical collocations consist of lexical elements with strong dependencies between each other and without the possibility of exchanging any of the elements. For example, saying 'to beat about the bush' is possible, but any other expression consisting of semantically similar words is considered wrong: 'to *hit about the bush' or 'to *kick about the bush'.

A second, mathematically motivated line of influence on today's computation of relations between words was established by Zellig Harris, who introduced the distributional hypothesis (Harris, 1968). He believed that linguistic analysis should be understood in terms of a statistical distribution of components at various hierarchical levels and constructed a practical conception on this topic. His own summary could very well have been the motto of this work:

> [T]he structure of language can be found only from the non-equiprobability of combination of parts. This means that the description of a language is the description of contributory departures from equiprobability, and the least

> statement of such contributions (constraints) that is adequate to describe the sentences and discourses of the language is the most revealing.
> (Harris, 1968)

However, Harris' and to some degree that of his students' attention (one of them being Noam Chomsky) was turned towards a more syntactic (formation rules) and logic (transformation rules) interpretation of meaning instead of lexical semantics (focussing on a broader understanding of relations between linguistic units). Nevertheless he believed that language is a system of many levels, in which items at each level are combined according to their local principles of combination. This does not necessarily exclude semantics.

Several decades later, these two directions of research (those of Firth and Harris) were picked up by Choueka, Klein, and Neuwitz (1983), Smadja (1989) and Church et al. (1991). The latter two had already developed an interpretation of meaning in linguistics from a computational point of view (Church et al., 1989). This new approach was partially derived from psycholinguistic research of word associations, and was combined with methods from information theory (mutual information) and computation (co-occurrences). Church applied this to simulate learning on a large corpus of text. Knowledge from the simulated learning about word associations was used to extract lexical and grammatical collocations. He also pointed out other possible applications, in particular the solution of polysemy.

It is clear that the work of Smadja, Church and others (to follow) is not a bare description of a method to semi-automatically compute lexical or grammatical collocations. Their usage of the term 'word association' indicates a broader meaning and indeed, in their examples of automatically computed, strongly associated word pairs standing in relations such as meronymy or hyperonymy are present. Smadja mentions them as examples of where Church's algorithm computed just 'pairs of words that frequently appear together' (Smadja, 1993). Lin (1998b) even considers 'doctors' and 'hospitals' as unrelated and thus wrongly computed as significant by Church and Hanks (1990), despite the meronymy relation between these words.

The much cited work of Dunning (1993) is important for two reasons. On the one hand, with the log-likelihood measure he introduces an improved mathematical foundation to this field of research. On the other hand and more importantly, he abstracts from the extraction of collocations in particular (only mentioning it) and calls the process 'statistical text analysis', which names the topic more precisely though more abstractly.

3.1.2. Computing semantic similarity

Since the early 1990s, the development of the statistical analysis of natural language has split into three directions. The first, already described as extraction of collocations, was initiated by Church and Smadja. It has been continued by Lehr

(1993) (see also Lehr (1996)), Evert and Krenn (2001), Seretan (2003) and most recently in Evert's dissertation (Evert, 2004). The main applications of this line of research are located in translation and language teaching, where it is important to know which expressions are common and which are not possible to avoid typical foreigners' mistakes.

The second line of development can be roughly named **extraction of word associations** and **computation of semantic similarity**. Initially mentioned by Church and also Schvaneveldt (1990), the concept is to (semi-)automatically extract pairs of 'somehow' related or similar words by observing their co-occurrence statistics. The resulting word pairs of significant co-occurrence are not solely idiosyncratic collocations. Many factors can cause two words to frequently co-occur, all of which could be subsumed as word associations. However, since this is a rather vague relation, it allows for a lot of interpretation.

In fact, in a given special context almost any two words might be considered associated with each other in some way. Nevertheless, the results obtained by algorithms from this field were useful, and thus have been utilized in many different applications, including word sense disambiguation (Agirre and Rigau, 1995; Yarowski, 1995; Pedersen and Bruce, 1997; Karov and Edelman, 1998; Pantel and Lin, 2000), word sense induction (or discrimination) (Schütze, 1998; Bordag, 2003; Purandare, 2004; Ferret, 2004; Bordag, 2006b; Biemann, 2006a), the computation of thesauri (Grefenstette, 1994; Lin, 1998a) and to a lesser extent in key word extraction (Matsumura, Ohsawa, and Ishizuka, 2003) (PAI) or (Witten et al., 1999) (Kea)), as well as text summarization (Salton et al., 1997; Mitra, Singhal, and Buckley, 1997). Manually acquired and automatically extended knowledge about word classes (using a trained part-of-speech tagger) has frequently been used to allow only pairs of words of the same syntactical word class to be taken into account. This significantly improves the perceived quality of the results, as seen in Grefenstette (1996). However, manually acquired knowledge, and therefore any algorithm based on such knowledge, does not fit the otherwise knowledge-free paradigm.

Additional to research on word associations, there is another related line of research which is concerned with measuring of semantic similarity based on a manually or automatically created thesaurus, including Grefenstette (1994; D'ejean et al. (2005) (for expanding thesauri) or Jiang and Conrath (1997) for work on measuring semantic similarity. Since a thesaurus can be viewed as a graph structure, it is possible to assume that the more distant two nodes are in this graph, the less similar the words represented by these nodes are. Strikingly, some algorithms based on pure co-occurrence data (Dagan, Marcus, and Markovitch, 1995) or hybrid approaches (Resnik, 1998), appear to yield results better than those based on manually created thesauri. But due to lack of comparable data, such statements remain unproven. There have been attempts at creating theoretical frameworks for such kinds of algorithms, yet they were either too narrow concentrating mainly on collocations (Lehr, 1996), or largely ignored and underestimated, such as Rieger (1991), perhaps because involving cognition. Yet these attempts show the need for such a framework and represent important work on the way to a proper understanding of the effects involved.

Unfortunately, the lack of an accepted and acknowledged model has already led to a growth in terminology and varying understanding of certain terms. For example, in the work by Terra and Clarke (2003) ‘word similarity measures’ are described, whereas other authors refer to the same methods as ‘word association measures’ (Rapp, 1996; Jiang and Conrath, 1997; Lin, 1998b). ‘Word association’ itself is used in either the psycholinguistic sense of association (Rapp, 2002) or in the statistical meaning of association as a synonym of correlation (Dunning, 1993). The notion of ‘context’ is scattered across a broad spectrum, ranging from n-gram models, where context is simply an n-gram, to windowing models, where context is defined as a number of words to the left and right of the observed word, to a notion of context which means the whole text within which the observed word occurs. Sometimes, the set of significant co-occurrences of a given word is labeled context, too.

Evaluating the results of semantic similarity algorithms has proven to be quite complicated. There is no easy way to define a gold standard, hence many different methods of indirect evaluation have been used. To begin with, psycholinguistic association experiments (priming experiments) (Burgess and Lund, 1997) have been used to generate human-based pairs of words associated with each other. Others have used the TOEFL synonym tests (Rapp, 1996; Landauer and Dumais, 1997; Jiang and Conrath, 1997; Terra and Clarke, 2003). The easiest and perhaps most comparable evaluation is one using large manually crafted knowledge sources such as Roget’s Thesaurus (Roget, 1946), WordNet (Miller, 1990; Fellbaum, 1998) or GermaNet for German (Hamp and Feldweg, 1997; Kunze and Wagner, 1999) as a gold standard. Unfortunately though, evaluations using these sources can be done in many different ways, crippling comparability. A standardized tool set or instance is needed.

3.1.3. Extraction of linguistic relations

The third line of development, the (semi-)automatic **extraction of particular linguistic relations** (or **thesaurus relations**) (Ruge, 1997), also known as automatic thesaurus construction (Shaikevich, 1985; Güntzer et al., 1989), must be distinguished from the other two lines of research, because it embeds them and introduces a different methodology (second order statistics, differentiating between syntagmatic and paradigmatic relations (Rapp, 2002), context comparisons (Biemann et al., 2004) etc.).

In the previous line of research, the term ‘word association’ was not well defined and has been used to denote various kinds of linguistic relations. This frequently included synonyms, sometimes plain word association (*play, soccer*) and sometimes other linguistic relations such as derivation, hyperonymy etc. In this third line of research, there is a constructive awareness for the different relations, and with it the means and need for an algorithmic differentiation between them.

Seen from this perspective, extraction of collocations or typical word usages (Heyer et al., 2001) is one possible task, in addition to the extraction of synonyms (Turney, 2001; Rapp, 2002; Baroni and Bisi, 2004), antonyms (Grefenstette, 1992), hyperonyms (Hearst, 1992), meronyms (Berland and Charniak, 1999) or even the qualitative direction of adjectives (negative vs. positive) (Hatzivassiloglou and McKeown, 1997; Turney, 2002) etc. Word sense distinction or induction, contrary to word sense disambiguation (Schütze, 1998; Neill, 2002; Tamir and Rapp, 2003; Bordag, 2003; Purandare, 2004; Ferret, 2004; Bordag, 2006b) fits to this area as well, since it can be viewed as describing relations between different words sharing the same word forms.

3.1.4. Discovering structure

There is no doubt that natural language is structured. The question is rather: which method to employ to discover and extract this structure? It is important to recall that in this work both syntax and semantics are considered as structure. At the first glance there are only a few directly observable parameters. Given a large sample of natural language in the form of a stream of texts, it is possible to observe the division into texts, paragraphs, sentences and finally (in most languages) the division into single word forms. It is also possible to encounter repeated words, paragraphs or even texts. It is possible to compute distributional statistics on the frequency of words, paragraphs and texts (Quantitative Linguistics). It is further possible to observe significantly co-occurring word forms within n-grams, sentences, paragraphs, texts or just within a fixed size window, and the distribution statistics of these co-occurrences. This corresponds to the syntagmatic dependency in the previous chapter. The point at which the structure of language beyond simple text-, paragraph- or word form boundaries is revealed is reached when certain word form pairs are observed more often than expected.

The expectation results from the mathematical unrelatedness, independence or unstructuredness hypothesis, which can be formulated as follows: If the word forms are unrelated or the language unstructured, any two word forms (or other units) co-occur with a probability of their multiplied relative frequency. If they do not co-occur with this predicted probability, the occurrence of the one word either inhibits or attracts the occurrence of the other, which means they are positively or negatively related to each other, implying a structure in the natural language.

This simplistic view is not without problems: first, there seems to be more structure than can be observed from the co-occurrence of word forms. For example, word forms are morphologically structured units. Some groups of word forms whose combined meaning deviates from the combination of the individual word forms. Sentences are not just sets of words: word order does matter (in most languages) and the same holds for paragraphs, texts and even for text streams. But is it really more than can be observed from word form co-occurrences, or is the observation of the co-occurrence merely the first step in a long chain of methods to unveil the structure of language automatically?

As mentioned in the previous chapter, it is possible to create a meaningful sentence containing any two word forms. For any case a context is imaginable within which this sentence would be meaningful. But language and its usage is redundant - additional explanations, descriptions or seemingly unnecessary (in the sense of delivering only the intended information) adjectives and other word forms are put into a sentence to convey the intended meaning in a highly redundant or modified way (increasing the beauty and readability of a sentence). For these reasons it is not a binary decision whether or not to put an additional word form into a sentence. Rather, it should be understood as a **continuum**, where some words are more necessary than others in order to convey the intended information. Viewed this way, sentences such as *He gave to him.* with a missing object, need not be treated especially, as long as the context for that particular sentence is very explicit about the object.

It can then be attempted to filter the specific (to a given context) information of a sentence (where two or more words are used together, which do not necessarily have something in common) against the redundancy information, which usually gives extra information about the world. As seen in the following example, there are words like *sweet*, *mellow* and *peaceful*, which are spatially close to each other in the sentence, and can be considered as being associated in the same manner as *vagueness* and *terror* in the middle of the sentence. However, *sweet* could not possibly be associated with *terror* in the same way, although both are observed in the same sentence.

> All was sweet and mellow and peaceful in the golden evening light, and yet as I looked at them my soul shared none of the peace of nature but quivered at the vagueness and the terror of that interview which every instant was bringing nearer.
>
> Doyle (1902)

This shows the need for a method that distinguishes between two usage types. The first type comprises the specific (relevant only to this sentence) co-occurrences of two or more words which appear together only to convey one particular piece of information. The second type comprises co-occurrences of two or more words that convey general world knowledge or increase readability or beauty of the sentence. The second type also reflects syntagmatic dependency. There are other related issues which all interfere with each other: idiomatic expressions (also called collocations), multi word lexemes, non-compositional compounds, and various semantic relations between words, such as hyperonymy, antonymy, meronymy, or synonymy. All of them, parts of the structure of natural language, result in the co-appearance of word forms in sentences. Alternatively, they can also inhibit each other's appearance.

Although to date the co-occurrence measure based methods do not attempt to distinguish between these relations (the main goal being somewhat more mathematically than linguistically motivated), they can be employed to describe a more complex algorithm that gradually reveals the structure of natural language. Several such attempts are made throughout the remaining chapters of this work.

3.2. Co-occurrence significance

This section gives a brief overview of a selection of co-occurrence significance measures. The selection is based on which of the many possible measures are either frequently encountered in related work or are mathematically well-founded. Given that most measures are either standard measures or have been described in detail by other researchers (Evert, 2004), the selected measures are described only briefly. However, emphasis is put on giving the measures in an explicit form with regards to the observable variables and on relations and motivations for the various measures.

As a first step in the automatic discovery of language structure through statistical significance tests, it is important to become aware of all the parameters that can play a role. First, let be assumed that the co-occurrences are measured on a corpus which is large, but of fixed size n (n can be the total number of running words, sentences, texts etc.). Irrespective of that, the formulae to follow can be transformed into the variant with a corpus of infinite size if the occurrence probability of the various items in comparison to the unknown size is known. These probabilities can be obtained by measuring a fixed part of size n of this infinite corpus, (Ciaramita and Baroni, 2006).

Secondly, for each element A (usually a word) the number of occurrences n_A within this corpus is known as the frequency $f(A)$. Frequency can be either simply the number of times A occurred: n_A, which is the default, or it can be more similar to the physical notion of frequency: occurrences of A per text (or some other frame, e.g. per million words (Church and Gale, 1995)) or in comparison to the corpus size n, where n can either be the number of running words or the number of sentences. The latter case is commonly used as an approximation of the probability $P(A)$ with which the unit A is expected to occur in the next sentence:

$$P(x) = \frac{f(A)}{n} \tag{3.1}$$

The third parameter determines whether two specific word forms count as co-occurring or not. They are considered as co-occurring if they are sufficiently close to each other, and not co-occurring if this is not the case. Although this explanation seems trivial, the exact interpretations can vary greatly, because the explicit distance parameter has not yet found its way into the standard formulae (except in the work by Holtsberg and Willners (2001)). Instead, it is often used to to distinguish between 'does co-occur' or 'does not co-occur', with varying values for that parameter (Schütze, 1992a; Schütze, 1995) to decide that. All following formulae measure co-occurrence in an unspecified window. Evaluating the different measures is performed by measuring co-occurrences in sentences.

The fourth parameter is the interpretation of the results obtained by measuring co-occurrence. In early works on collocation extraction (Church and Hanks, 1990; Smadja, 1989), the **globally** strongest co-occurrences (i.e. the ones with the highest significance value) were considered as the result. Lately, with interest shifting towards extracting particular semantic relations *for each word*, the **locally** for each input word strongest (again in the meaning of the highest significance value) co-occurrences (output words) are considered. These words, ranked by significance of co-occurrence, are called either word associations (Church and Hanks, 1990), set (Manning and Schütze, 1999) (though they also refer to them simply as the co-occurrences, implying the significance), significance list (Krenn and Evert, 2001), vector (Schütze, 1992b; Rapp, 2002) or context vector (Curran, 2003).

In this work the notions 'co-occurrence vector' and 'co-occurrence ranking' are used synonymously - yet they highlight different properties. For most applications and for all algorithms in the following three chapters only the x most significantly co-occurring words of a given input word are important. For the similarity computations introduced below in this chapter, the ranking is irrelevant, but the co-occurrence significance values in the vector are relevant. Therefore, this chapter primarily uses vector, whereas later ranking is used. The same applies to the difference between 'similarity vector' and 'similarity ranking'.

3.2.1. Basic measures

The most straightforward way to measure the significance of the number of co-occurrences of the words A and B is to take the number of co-occurrences n_{AB}. However, 'significance' in this case (and the next two) is not statistically motivated. This kind of measuring can be used as an evaluation baseline, where all other measures should perform better:

$$sig_{base}(A, B) = n_{AB} \tag{3.2}$$

The drawbacks of this measure are evident. For example, it disregards the frequency of both A and B. Thus, for a very frequent A it is not very interesting that it co-occurs with B twenty times. On the other hand, if the frequency of both A and B equals the number of their co-occurrence - then this is highly relevant information.

One improvement could be the Jaccard coefficient (Frakes and Baeza-Yates, 1992) (also called the Tanimoto distance (Tanimoto, 1958)), which in this special case has the following form:

$$sig_{tanimoto}(A, B) = \frac{n_{AB}}{n_A + n_B - n_{AB}} \tag{3.3}$$

This measure has been employed by Bensch and Savitch (1992) to create a lexical graph on which they then determine the minimal spanning tree - the methodology is similar to Schvaneveldt (1990). Grefenstette (1992) used it for measuring word similarity.

Another possibility, as used by Frakes and Baeza-Yates (1992) and Smadja, McKeown, and Hatzivassiloglou (1996) for retrieving collocations, would be to compute the Dice coefficient (Dice, 1945):

$$sig_{dice}(A, B) = \frac{2 \cdot n_{AB}}{n_A + n_B} \tag{3.4}$$

These two measures have several drawbacks that they share with the baseline. Although both account for the possibility of insignificance of co-occurrence of a frequent word with a non-frequent one, it is still informative to know (i.e. significant) that a word occurring only twenty times co-occurs also twenty times with a word occurring 2 000 times. In this case, both measures yield very small numbers which may appear insignificant. Another very important drawback is that they are not normalized against the corpus size. The rare (because accidential) co-occurrence of two low-frequency words might have a higher statistical significance than the frequent, because systematic co-occurrence of two high-frequency words.

However, in comparison to the following significance measures they have the advantage of giving normalized numbers in the range [0..1]. If these two measures are used to rank the co-occurrences according their significance, then they always produce identical rankings, because they are monotonic transformations of each other. However, they do produce different numerical values for the more significant and less significant co-occurrences, which is why similarity measures using these significance values may produce different rankings.

3.2.2. Assumption of statistical independence

Contrary to that there is a statistically founded way to find the proper measurement. The standard procedure in statistics is to formulate a null hypothesis. This null hypothesis defines probabilities (or expectation) for each possible outcome of an experiment, for example by using an assumed or guessed underlying distribution. Then the experiments are performed and their results are used to determine the significance of the observed deviation from the expected values. If the deviation according to the test is significant, then the null hypothesis can be rejected.

The significance of the observation is interpreted both in terms of by how large the deviation from the expected value was as well as how large the variance in the experiments was. Additionally it is important to know how many degrees of freedom are involved - that is how many variables are possibly interdependent.

The co-occurrence experiments can be modeled accordingly as a number of experiments with positive or negative outcomes: two words either co-occur or do not. Thus, there are two random discrete variables (the word forms). The null hypothesis is that they are statistically independent of each other. If they are indeed independent, then the occurrence of the one should not correlate with the occurrence of the other. It is possible to raise counts of four different conditions: either both words are present $f(AB)$, only one $f(\neg AB)$ or the other $f(A\neg B)$ is present, or none $f(\neg A\neg B)$.

Similarily to the excellent and more detailed (with respect to the underlying principles of statistics) overviews given by Evert (2004) and Tan, Kumar, and Srivastava (2002), the contingency (Table 3.2.2) can be expressed in terms of the raised counts, i.e. reprsenting all possible observations.

	A	$\neg A$	
B	$f(AB) = n_{AB}$	$f(\neg AB) = n_B - n_{AB}$	$f(B) = n_B$
$\neg B$	$f(A\neg B) = n_A - n_{AB}$	$f(\neg A\neg B) = n - n_A - n_B + n_{AB}$	$f(\neg B) = n - n_B$
	$f(A) = n_A$	$f(\neg A) = n - n_A$	s

Table 3.1.: Contingency table for co-occurrence of items within sentences.

If the occurrence of A is independent of the occurrence of B, then all following statements must be true: $p(A, B) = p(A) \cdot p(B)$, $p(\neg A, B) = p(\neg A) \cdot p(B)$, $p(A, \neg B) = p(A) \cdot p(\neg B)$ and $p(\neg A, \neg B) = p(\neg A) \cdot p(\neg B)$. Constituting the probabilities with the counts n, n_A, n_B and n_{AB} yields the following equivalence:

$$p(A, B) = p(A) \cdot p(B), \text{ thus } \frac{n_{AB}}{n} = \frac{n_A \cdot n_B}{n^2} \tag{3.5}$$

The other three statements are equivalent to the first one, in that they can be transformed into it.

Because language is structured and words are not independent of each other, testing whether the number of co-occurrences of a word pair is significant is usually omitted. In fact, any word A observed in a corpus always has a set of words co-occurring significantly with it. Instead of testing whether the frequency of observed co-occurrence counts is significant, a quantification is performed which gives a value stating the degree of significance. In some tests, such as the log-likelihood test (see below), this value can be used to determine the degree of certainity that it is significant. For example, the degree of 0.025 means that with a 2.5% probability the observed result is wrong and an observed deviation found to be significant is in truth insignificant.

The deviation-values are then used to rank the words co-occurring with A, according to their significance. The stronger the significance of some word B co-occurring with A, the safer it is to assume a correlation between A and B. This in itself is symmetrical, however for a more frequent word A, it is not uncommon to find several thousand strongly correlated words. For most applications and algorithms (including the ones in the following chapters), only the x top ranking co-occurring words are useful. This maximizes the probability that each is also linguistically related to the input word. However, once the significance values are used to construct a ranking, the ranking positions cease to be symmetrical. These rankings can also be used to represent a global (with respect to the corpus) 'meaning definition' of the word.

The following methods are various ways to measure the significance of the difference between observed, and the expected values, given the independence assumption.

3.2.3. Mutual Information

A fairly common method of measuring co-occurrence significance is to compute the mutual information (Church et al., 1989; Dagan, Marcus, and Markovitch, 1995; Lin, 1998a; Terra and Clarke, 2003) for word associations, as well as for computing synonyms (Turney, 2001; Baroni and Bisi, 2004), whereas Church et al. (1991) uses it directly for collocations. The idea is to measure the mutual information (MI) between two (assumedly) random variables, in this case words, by directly comparing the probability of observing A and B together (joint probability) with the probabilities of observing them independently:

$$sig_{MI}(A,B) \equiv \log_2 \frac{p(A,B)}{p(A) \cdot p(B)} = \log_2 \frac{n \cdot n_{AB}}{n_A \cdot n_B} \tag{3.6}$$

Another way to arrive at this formula is to look at the four statements which follow from the contingency table (Table 3.2.2), which represent the independence assumption. If one of the statements does not hold, it would be interesting to find out by how much it was missed. This is done by adding a factor x, which must equal 1 in the case of independence:

$$n_{AB} = \frac{n_A \cdot n_B}{n} \cdot x \tag{3.7}$$

which can then be transformed:

$$x = \frac{n \cdot n_{AB}}{n_A \cdot n_B} \tag{3.8}$$

Taking the logarithm of this gives the same formula as above.

This measure seems to be an improvement over the trivial measures, because it both detects the significance of (for example) A co-occurring with B 20 times, if

A's frequency is 2 000 and B's is 20, as well as 'normalizes' against the corpus size. But the normalization against corpus size only results in rewarding of low frequency words with higher significances. This problem becomes even more apparent in the case of perfect statistical dependence between both words, thus $p(A) = p(B) = p(A, B)$. In this case:

$$sig_{MI}(A, B) = \log_2 \frac{p(A)}{p(A) \cdot p(B)} = \log_2 \frac{1}{p(A)} = \log_2 \frac{n}{n_A} \tag{3.9}$$

This results in the arguably faulty conclusion that the mutual dependence of less frequent word pairs is more 'informative' or significant than the mutual dependence of less frequent word pairs. Furthermore, the single occurrence of two words being a co-occurrence $f(A) = f(B) = f(AB) = 1$ gives the highest possible score, since in this case the score returned is $\log_2 n$ which greatly overstates this co-occurrence, as pointed out by Dunning (1993). Besides, taking the logarithm in this case of application of mutual information is not really necessary, because contrary to the true significance measures introduced below, it only scales down the numbers monotonously. This then makes it obvious that this measure is similar to the Dice coefficient except for the multiplication with the corpus size.

Alternatively, it is possible to use the Pointwise Mutual Information (PMI) or a simplified form (omitting the n in the divisor) thereof, in a manner similar to the one introduced by Kilgarriff and Tugwell (2001), where the MI is multiplied by the log-frequency or plain frequency of the co-occurrence:

$$sig_{LMI}(A, B) = n_{AB} \log_2 \frac{p(A)}{p(A) \cdot p(B)} = n_{AB} \log_2 \frac{n \cdot n_{AB}}{n_A \cdot n_B} \tag{3.10}$$

Incidentally, this Lexicographer's Mutual Information (LMI) is nearly a reformulation of the Poisson approximation (further below).

3.2.4. Log-likelihood test

Briefly restating the main points of Dunning (1993), a proper statistical modeling and the use of suitable approximations is needed. He proposes that measuring co-occurrences should be modelled by statistical means by assuming each sentence (or other window) as an experiment in a row of repeated experiments with two outcomes: either both words A and B are contained in the sentence, or not. The properties of these experiments can be assumed as follows:

- The probability of the occurrence of the observed words does not change (which is true by definition, since the probability is determined by using a fixed size corpus).
- The experimental outcomes are not dependent on each other (or the dependence falls off rapidly with distance between the experiments, so that it can be disregarded).

- Both A and B occur in every sentence at most once. For high-frequency words this is not true and a common strategy to handle such cases is to ignore each further occurrence of a word in a sentence when determining the sentence co-occurrences.

In this case it is possible to describe the distribution of the outcomes using the binomial distribution. It describes the probability that a random variable, in this case the co-occurrences of A and B, is going to be observed exactly k times if there are n experiments and the independence assumption gives us the probability p of A and B co-occurring:

$$p(k) = p^k (1-p)^{n-k} \binom{n}{k} \tag{3.11}$$

where the mean is np and the variance $np(1-p)$. If $np(1-p) > 5$, the distribution of this variable approaches the normal distribution (Dunning, 1993). In the case of measuring co-occurrences, however, n is very large, whereas $p = p(A) \cdot p(B) = \frac{n_A \cdot n_B}{n^2}$ (from the independence assumption) is usually very small, thus:

$$np(1-p) = \frac{n_A \cdot n_B \cdot (n^2 - n_A \cdot n_B)}{n^3} \tag{3.12}$$

According to the Zipf distribution of word frequencies (Zipf, 1949), it is safe to say that except for the a highly frequent words of a corpus $np(1-p) < 5$ and that for far more than half of the words of a corpus it is even $np(1-p) << 5$.

Dunning proposes to use the generalized likelihood ratio test, which is the ratio λ between the maximum value of the likelihood function (sometimes called probability function) under the constraint of the null hypothesis to the maximum with that constraint relaxed. If the null hypothesis is true, then $-2\log\lambda$ (therefore this test is often called the log-likelihood measure) is asymptotically χ^2 distributed (Moore, 2004) with degrees of freedom equal to the difference in dimensionality of the hypotheses Θ and Θ_0. Since the degrees of freedom equals 1 in this case, when using a confidence level of 0.025 the significance threshold is 5.02. Any values above that are considered as significant with an error probability of 2.5% (or less).

A null hypothesis $H(\theta|x) : \theta \in \Theta_0$ is stated by saying that the parameter θ is in a specified subset Θ_0 of the parameter space Θ. The likelihood function $H(\theta) = H(\theta|x)$ is a function of the parameter θ with x held fixed at the value that was actually observed. The statistical model to be used is the binomial distribution. The general form of the likelihood ratio is then:

$$\lambda = \frac{\max_{\theta \in \Theta_0} H(\theta|x)}{\max_{\theta \in \Theta} H(\theta|x)} \tag{3.13}$$

Another way to formulate this (as originally done by Dunning (1993)) is to compare the hypothesis $H(\theta|x)$ with θ as a point in the parameter space Θ and x a point in the space of observations.

For co-occurrences, the single parameter of the statistical model based on the binomial distribution is p, whereas the experimental outcomes can be described by n (the number of experiments) and k (the number of positive outcomes). Since there are two binomial distributions to be compared, the one representing the null hypothesis and the one representing the observed data, we have $\theta_1 = p_1$ and $x_1 = k_1, n_1$ as well as $\theta_2 = p_2$ and $x_2 = k_2, n_2$, thus:

$$H\left(p_1, p_2; k_1, n_1, k_2, n_2\right) = p_1^{k_1}\left(1 - p_1\right)^{n_1 - k_1}\binom{n_1}{k_1} p_2^{k_2}\left(1 - p_2\right)^{n_2 - k_2}\binom{n_2}{k_2} \tag{3.14}$$

The hypothesis that the two distributions have the same underlying parameter is represented by $p_1 = p_2$. Thus, the likelihood ratio λ for this test is:

$$\lambda = \frac{\max_p H\left(p, p; k_1, n_1, k_2, n_2\right)}{\max_{p_1, p_2} H\left(p_1, p_2; k_1, n_1, k_2, n_2\right)} \tag{3.15}$$

The maxima of the likelihood functions are achieved with $p_1 = \frac{k_1}{n_1}$, $p_2 = \frac{k_2}{n_2}$ and $p = \frac{k_1 + k_2}{n_1 + n_2}$, with $L\left(p; k, n\right) = p^k (1 - p)^{n-k}$ this reduces the ratio to:

$$\lambda = \frac{max_p L\left(p; k_1, n_1\right) L\left(p; k_2, n_2\right)}{max_{p_1, p_2} L\left(p_1; k_1, n_1\right) L\left(p_2; k_2, n_2\right)} \tag{3.16}$$

Taking the logarithm of the likelihood ratio gives

$$-2 \log \lambda = 2 \left[\begin{array}{l} \log L\left(p_1; k_1, n_1\right) + \log L\left(p_2; k_2, n_2\right) \\ - \log L\left(p; k_1, n_1\right) - \log L\left(p; k_2, n_2\right) \end{array} \right] \tag{3.17}$$

From the contingency table follows that $k_1 = n_{AB}$, $n_1 = n_B$, $k_2 = n_A - n_{AB}$ and $n_2 = n - n_B$. Using these equations it is possible to give the final equation for the log-likelihood ratio:

$$-2 \log \lambda = 2 \left[\begin{array}{l} n \log n - n_A \log n_A - n_B \log n_B + n_{AB} \log n_{AB} \\ + \left(n - n_A - n_B + n_{AB}\right) \log\left(n - n_A - n_B + n_{AB}\right) \\ + \left(n_A - n_{AB}\right) \log\left(n_A - n_{AB}\right) \\ + \left(n_B - n_{AB}\right) \log\left(n_B - n_{AB}\right) \\ - \left(n - n_A\right) \log\left(n - n_A\right) - \left(n - n_B\right) \log\left(n - n_B\right) \end{array} \right] \tag{3.18}$$

Restating this equation in this explicit form avoids numerical problems with too small probabilities (Weeds and Weir, 2005) and results in run-time performance comparable to the Poisson approximations below. Since the log-likelihood ratio is always positive, it is possible to use the basic difference between observed and expected values (or rather, n_{AB} and $\frac{n_A \cdot n_B}{n}$ as in the Mutual Information definition above) in order to decide, whether a computed significance is positive or negative. This results in two different significance measures:

$$sig_{lg}(A, B) = -2 \log \lambda \tag{3.19}$$

and:

$$sig_{lg2}(A,B) = \left\{ \begin{array}{ll} -2\log\lambda & \text{if } n_{AB} < \frac{n_A \cdot n_B}{n} \\ 2\log\lambda & \text{else} \end{array} \right\} \tag{3.20}$$

With respect to the task of retrieving semantically or syntactically associated words, it is unclear which version is preferable. On the one hand, ignoring the distinction between significant co- and non-co-occurrence should rank words higher, which seem to inhibit each other. This could for example be the case for synonyms. On the other hand, it might be the case that high-frequency stop words in particular co-occur less often with each other than expected for no apparent reason. Therefore both variants are tested in the evaluation in Section 3.4.

However, making such a distinction between significant co- and non-co-occurrence should also imply measuring all words pairs which do not co-occur at all, yet could be expected to co-occur - that is where $n_{AB} < \frac{n_A \cdot n_B}{n}$. The primary two reasons for not including a deliberate analysis is the feasability and the desired ranking. Measuring all possible non-co-occurrences means to compare the frequency of every word with every other word. This could be restricted to only the more frequent words, because only their frequency is sufficient to achieve a smaller observed than expected value with any other word. The other reason is that the desired output of the co-occurrence algorithm on the whole is to rank the most significant co-occurrence highest. Hence, the most significant non-co-occurrences would be ranked lowest and not affect the evaluations given in Section 3.4.

The log-likelihood measure has been picked up and successfully employed by other researchers (such as Berland and Charniak (1999)) for computing the meronymy relation, as well as Rapp (2002) for the general computation of word associations. Krenn (2000) and later Evert and Krenn (2001) included this measure in their evaluation of various different lexical association measures, and found it to be one of the best measures.

3.2.5. Poisson distribution

Another approach at measuring significant co-occurrences has been taken by Quasthoff and Wolff (2002). Assuming that the number of random co-occurrences follows a Poisson distribution according to the independence assumption, they compute the logarithm of the probability of the given observation. Taking the (natural negative) logarithm turns a probability into a significance value, which indicates by how much the expected value was missed (analogously to the log-likelihood measure above). Formal proof that the Poisson distribution approximates this model is given by Holtsberg and Willners (2001). This approach relies directly on the fact that there will be words which co-occur significantly with a given word A and it is only important to find these words. This is solved by taking those with the highest significance value. Though possibly being less precise on some occasions than the log-likelihood method, the differences are assumed to be small enough to be disregarded.

The mean and variance of the Poisson distribution is $\lambda = np$. In this case p is the probability of words A and B to co-occur k times, which is $p(A) \cdot p(B) = \frac{n_A \cdot n_B}{n^2}$ under the independence assumption as given above, thus $\lambda = \frac{n_A \cdot n_B}{n}$. Taking the negative natural logarithm of the Poisson distribution and setting k to the observed value n_{AB} results in the significance of the deviation from the expected observation:

$$sig_{ps}(A, B) = -\ln\left(\frac{1}{k!}\lambda^k e^{-\lambda}\right) = \ln k! - k \ln \lambda + \lambda \tag{3.21}$$

Since in the case of $k > 10$ the expression $\sqrt{2\pi k}\left(\frac{k}{e}\right)^k$ approximates $k!$ well (Stirling's formula), it is possible to approximate the significance computation in the following way:

$$sig_{ps1}(A, B) \approx k(\ln k - \ln \lambda - 1) + \frac{1}{2}\ln 2\pi k + \lambda \text{ with } k > 10 \tag{3.22}$$

Another possible approximation of $\ln k!$ for large k is $\ln k! = k \ln k - k + 1$. For large k it is possible to further simplify to $\ln k! = k \ln k$. This yields an even simpler significance formula:

$$sig_{ps2}(A, B) \approx k(\ln k - \ln \lambda - 1) \;\text{ with } k >> 0 \tag{3.23}$$

As mentioned above, this last approximation is very similar to the LMI measure 3.10. This becomes apparent, when writing λ out as $\frac{n_A \cdot n_B}{n}$ and transforming the subtraction (note that $k = n_{AB}$):

$$sig_{ps2}(A, B) \approx n_{AB}\left(\ln\frac{n_{AB} \cdot n}{n_A \cdot n_B} - 1\right) \tag{3.24}$$

The Poisson measures are approximations of the log-likelihood and the LMI measure is nearly a reformulation of the Poisson measures. This makes it possible to use the same thresholds (and confidence levels) for all these measures. Thus, if any of these methods determines two words to co-occur with a significance of 6, for example, then the error probability is approximately 1%. Or, from a practical point of view, with an assumed confidence level of 0.01, the resulting universal cutoff value of 6 can be taken to determine whether an observed co-occurrence is significant or not.

Two approximations were used in the process of arriving at the formula for *ps1* and *ps2* each, as opposed to the 'proper' log-likelihood measure: using a Poisson distribution instead of a binomial distribution, then using approximation formulae in order to achieve representations which are easy to compute. In this case, the last step is especially problematic, because for small n_A or n_B small k will often be of interest. This leads to the question: could using so many approximations

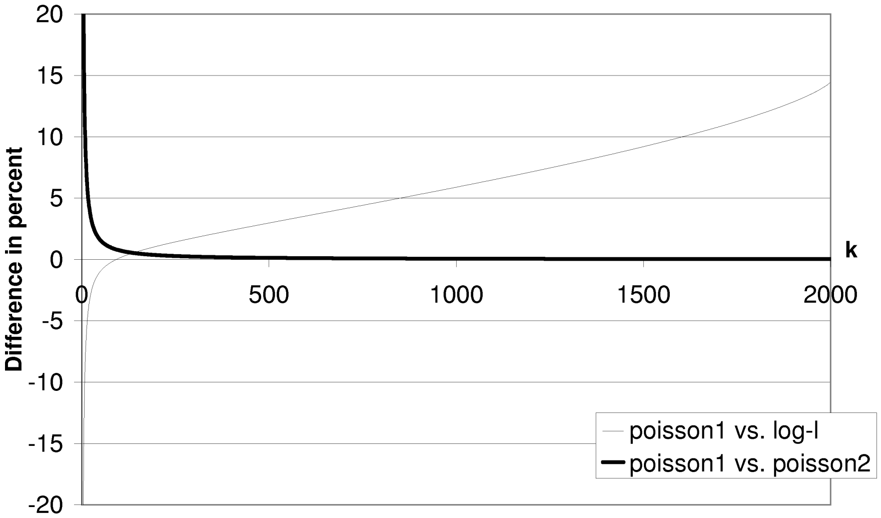


Figure 3.1.: Comparison of a Poisson distribution approximation *ps1* with the other approximation *ps2* as opposed to a comparison of a Poisson approximation with the log-likelihood measure *lg*.

be harmful in any way? A simple way to quantify the differences is to compare the numerical differences between the latter two approximations and to show that for larger k they indeed are asymptotically equal (corpus size 24 million sentences, $n_A = 2\,000$, $n_B = 4\,000$) and then compare one of the two with the log-likelihood ratio (same setting of corpus size and frequencies).

Figure 3.1 shows that the log-likelihood measure indeed differs from both Poisson approximations asymptotically for large k. However, the difference becomes significant only for a very large k, as compared to the frequency of a less frequent word, whereas Zipf's law indicates that most co-occurrences will be distributed in the low k's, as is illustrated in a sample measurement in Figure 3.2.

Therefore it is possible to formulate the hypothesis that the results obtained by the two measures should differ only slightly, especially if not the significance values are relevant, but instead the ranking of the results.

3.2.6. z-score and t-test

The z-score and the t-test are two further commonly used (cf. Evert (2004)) measures. Their explicit form, with respect to the four observable variables explained above is given, and they are included in the evaluations below.

The z-score is an asymptotic hypothesis test. It approximates the discrete binomial distribution with a continuous normal distribution. Its explicit form with respect to the present task of measuring the significance of co-occurrence observa-

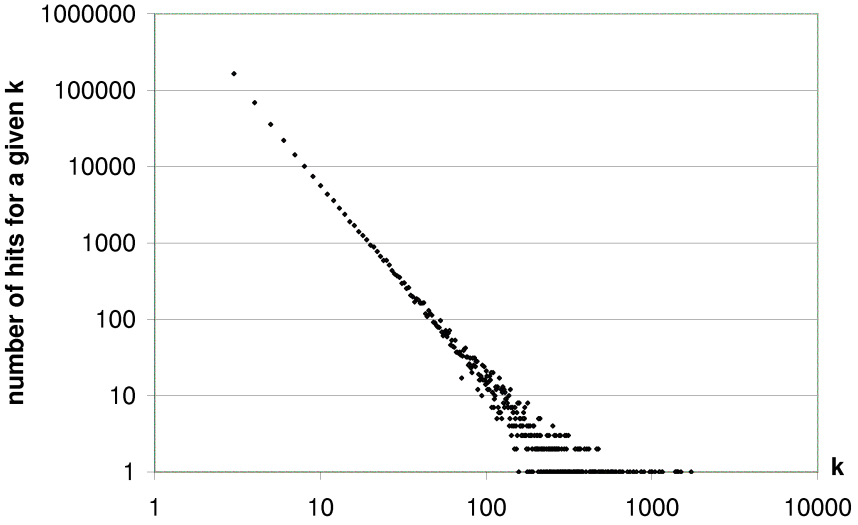


Figure 3.2.: Depicting the distribution of k for word pairs, where the frequency of the one word $f(A)$ is $3\,500 \leq f(A) \leq 4\,500$ and the frequency of the other word $1\,500 \leq f(B) \leq 2\,500$.

tions can be formulated as follows:

$$sig_{z-sc}(A,B) = \frac{n_{AB} - \frac{n_A \cdot n_B}{n^2}}{\sqrt{\frac{n_A \cdot n_B}{n^2}}} \tag{3.25}$$

Higher z-score values represent more evidence for a positive correlation between the two variables and thus stronger significance of co-occurrence.

The standard way to apply the t-test (Manning and Schütze, 1999) (also called t-score (Church et al., 1991) or Student's t-test) for measuring co-occurrence significance results in the following explicit form:

$$sig_{t-sc}(A,B) = \frac{n_{AB} - \frac{n_A \cdot n_B}{n^2}}{\sqrt{n_{AB}}} \tag{3.26}$$

Despite the different motivations that result in the z-score and the t-score respectively, in the current task the only difference between them is the divisor. In the one case (z-score), the difference between the expected and the observed value is being divided by the expected value $\frac{n_A \cdot n_B}{n}$. In the other case (t-score), by the observed value n_{AB}. For the z-score this can be translated into the rule that the highest significance values are achieved, if the difference between the observed and expected values is large, while the **expected value** itself is small. This is easily the case for frequent words, where the differences have much more room to vary, whereas the expected values still remain small. Hence, this measure might tend to overrate frequent words compared to infrequent words.

However, for t-score the rule is different. The highest significance values are achieved if the difference between the observed and expected values is large, while the **observed value** itself is small. This clearly emphasizes less frequent words. This also produces higher significances for high-frequency words which inhibit each other from occurring in sentences. Both these effects are responsible for the low performance of the t-score as reported below.

base	. , the of a and to in ' that
dice	fiction novels comedy adventure suspense romance novelist fantasy novel Cartland
tan	fiction novels comedy adventure suspense romance novelist fantasy novel Cartland
MI	Amiee Scotney Mastrovin Well-Tempered boogification Anouk Clavier Evallonia Fantastique underemphasized
LMI	a , of and the love in s as fiction
t-sc	. , the of and a to in ' that
z-sc	boogification Amiee Scotney Cartland Mastrovin Well-Tempered fiction Anouk Clavier anti-capitalism
lg	a , love of fiction and most the novels comedy
lg2	a , love of fiction and most the novels comedy
ps	love a fiction , of and most novels comedy novel
ps2	love fiction novels comedy novel adventure suspense most story Cartland

Table 3.2.: For each significance measure the 10 most significant co-occurrences of the word 'romantic' based on the BNC.

The examples in Table 3.2 support the hypotheses formulated for each introduced co-occurrence measure. The *base* ranks the co-occurrences almost entirely according to their frequency. The next two measures, *dice* and *tan*, produce identical and apparently very good rankings. *MI* ranks very rare words highest, whereas *LMI* contrarily emphasizes frequency of words, but less so that *t-sc*. The *z-sc* appears to rank rare words high in a manner similar but not identical to *MI*. Overall, the most balanced rankings are produced by *dice* (and *tan*), and the *ps2* measures with not too frequent or too rare words. The log-likelihood measures indeed appear to have a weakness for prefering some high-frequency words (as reported by Kilgarriff and Tugwell (2001)), while otherwise having a good overlap with *dice* and *ps2*.

3.3. Similarity measures

Having computed context representations of words, the next step is to compare them by comparing their contexts. The goal is to compute similarity relations between words. According to such a method, similar words are likely to be synonyms or cohyponyms, since these are relations of semantic similarity. Independently of how the context of a word was acquired, it can be represented as a vector (assuming a vector space model). Thus, for a word A the vector $\overrightarrow{A}$ is its context representation, obtained by using one of the methods mentioned above. It is not necessary to use a notation like that of Lin (1998a) or Curran (2003), since at this point there is no differentiation between various syntactic relations.

The vector $\overrightarrow{A}$ can also be represented as follows: $\overrightarrow{A} = (a_1, a_2, ..., a_n)$. In this case, n is the number of words in the vector space, and most a_i values will be zero-values. The value a_i is the significance value of the word A co-occurring with another word B, where the word B is represented in the i-th dimension. Thus, if the co-occurrence measure used is symmetrical, then at the position j (representing A) the value b_j will be the same as a_i.

There are several standard ways to compare two vectors (as described for example by Manning and Schütze (1999), Terra and Clarke (2003) or more recently Weeds, Weir, and McCarthy (2004)) and obtain an indication of how similar they are or how far away the two points represented by these vectors are from each other.

The most simple method to compare two context vectors is to count in how many dimensions they both have non-zero values, or, as Manning and Schütze (1999) term it, the matching coefficient (if the vectors are represented as sets):

$$sim_{base}(A,B)\left(\overrightarrow{A},\overrightarrow{B}\right) = \sum_{i=0}^{i=n} sgn\left(min\left(|a_i|,|b_i|\right)\right) \tag{3.27}$$

This measure is used as a baseline for the evaluation below. It has the property that it computes a similarity of 0 for utterly unrelated word pairs which do not share any words in their context representations. It does not take into account the length of the vectors, or the total number of non-zero entries in each. If two words B and C have identical amounts of matching co-occurrences, then they are ranked from highest to lowest according to their frequency.

Furthermore, this measure ignores the real significance values obtained in the previous step and only counts the non-zero values in a vector. Manning and Schütze (1999) term such vectors binary, and describe a number of measures applied to them.

3.3.1. Binary vectors

Definition

A binary vector $\overrightarrow{A}^{bin}$ is the result of a mapping from a real-valued vector $\overrightarrow{A}$ into the vector $\overrightarrow{A}^{bin}$ where:

$$\overrightarrow{A}^{bin} = \left(a_1^{bin}, a_2^{bin}, ..., a_n^{bin}\right) \text{ with } a_i^{bin} = sgn\left|a_i\right| \text{ and } i = 1, ..., n \tag{3.28}$$

Using this representation it is possible to adapt set operations for use on binary vectors:

Definition
The length $\left\|\overrightarrow{A}^{bin}\right\|$ of a binary vector $\overrightarrow{A}^{bin}$ is the number of non-zero values in that vector:

$$\left\|\overrightarrow{A}^{bin}\right\| = \sum_{i=0}^{i=n} a_i{}^{bin} \tag{3.29}$$

Definition
An intersection $\overrightarrow{A}^{bin} \cap \overrightarrow{B}^{bin}$ maps from two binary vectors $\overrightarrow{A}^{bin}$ and $\overrightarrow{B}^{bin}$ into one binary vector $\overrightarrow{C}^{bin} = \left(c_1^{bin}, c_2^{bin}, ..., c_n^{bin}\right)$ where:

$$c_i^{bin} = min\left(a_i^{bin}, b_i^{bin}\right) \text{ and } i = 1, ..., n \tag{3.30}$$

Definition
A union $\overrightarrow{A}^{bin} \cup \overrightarrow{B}^{bin}$ maps the two binary vectors $\overrightarrow{A}^{bin}$ and $\overrightarrow{B}^{bin}$ into one binary vector $\overrightarrow{C}^{bin} = \left(c_1^{bin}, c_2^{bin}, ..., c_n^{bin}\right)$ where:

$$c_i^{bin} = max\left(a_i^{bin}, b_i^{bin}\right) \text{ and } i = 1, ..., n \tag{3.31}$$

Using these adapted set operations it is now possible to give concise definitions (equivalent to the ones given by Manning and Schütze (1999)) for similarity measures on binary vectors. The definition of the previously given baseline measure can now be represented in the following way:

$$sim_{base}\left(\overrightarrow{A}, \overrightarrow{B}\right) = \left\|\overrightarrow{A}^{bin} \cap \overrightarrow{B}^{bin}\right\| \tag{3.32}$$

The overlap coefficient normalizes against the length of the vectors, since a shorter vector (having fewer non-zero values) can at most match the number of its non-zero values with those of the longer vector.

$$sim_{overlap}\left(\overrightarrow{A}, \overrightarrow{B}\right) = \frac{\left\|\overrightarrow{A}^{bin} \cap \overrightarrow{B}^{bin}\right\|}{\min\left(\left\|\overrightarrow{A}^{bin}\right\|, \left\|\overrightarrow{B}^{bin}\right\|\right)} \tag{3.33}$$

The drawback of this measure is that very short vectors will tend to be very 'similar' (or equal) to many other vectors. The Dice coefficient, applied to binary vectors, alleviates this problem by dividing by the total number of non-zero values:

$$sim_{dice}\left(\overrightarrow{A},\overrightarrow{B}\right) = \frac{2\left\|\overrightarrow{A}^{bin} \cap \overrightarrow{B}^{bin}\right\|}{\left\|\overrightarrow{A}^{bin}\right\| + \left\|\overrightarrow{B}^{bin}\right\|} \tag{3.34}$$

Another possibility is the Jaccard coefficient, which penalizes small overlaps, as opposed to large overlaps in contrast to the Dice coefficient:

$$sim_{jaccard}\left(\overrightarrow{A},\overrightarrow{B}\right) = \frac{\left\|\overrightarrow{A}^{bin} \cap \overrightarrow{B}^{bin}\right\|}{\left\|\overrightarrow{A}^{bin} \cup \overrightarrow{B}^{bin}\right\|} \tag{3.35}$$

When comparing the Dice coefficient with the Jaccard coefficient, the comparison can be reduced to the following inequation:

$$\left\|\overrightarrow{A}^{bin}\right\| + \left\|\overrightarrow{B}^{bin}\right\| - \left\|\overrightarrow{A}^{bin} \cap \overrightarrow{B}^{bin}\right\| \neq \frac{\left\|\overrightarrow{A}^{bin}\right\| + \left\|\overrightarrow{B}^{bin}\right\|}{\left\|\overrightarrow{A}^{bin} \cap \overrightarrow{B}^{bin}\right\|} \tag{3.36}$$

The only difference between these two measures is that in the one case the overlapping non-zero values are subtracted and in the other case are divided with. Since in the described vector space it always holds that $\left\|\overrightarrow{A}^{bin} \cap \overrightarrow{B}^{bin}\right\| \leq \left\|\overrightarrow{A}^{bin}\right\| + \left\|\overrightarrow{B}^{bin}\right\|$ both on the left and right side of the inequation, the effect is that of scaling the sum down (if there are overlapping non-zero values). Thus the Jaccard and the Dice coefficients will always yield the same similarity rankings for any word. Only the globally most similar word pairs will be different when using these two measures. Therefore the Jaccard measure is excluded from the evaluations below.

Another measure is the cosine measure between binary vectors:

$$sim_{cbin}\left(\overrightarrow{A},\overrightarrow{B}\right) = \frac{\left\|\overrightarrow{A}^{bin} \cap \overrightarrow{B}^{bin}\right\|}{\sqrt{\left\|\overrightarrow{A}^{bin}\right\| \cdot \left\|\overrightarrow{B}^{bin}\right\|}} \tag{3.37}$$

This measure is a simplification of the cosine (see below) on real-valued vectors. The difference is that the values are all set to 1. The difference to the Dice or Jaccard measure is that it normalizes against the length of the modified vectors. Using the same argument as when comparing the Dice and the Jaccard measure with each other it is possible to predict that this measure will produce very similar results, although not equal rankings.

3.3.2. Real-valued vectors

Ignoring the real significance values raises two potentially important problems:

- The ranking of the most significantly co-occurring words is ignored.
- In the computation of significant co-occurrences usually several kinds of thresholds are utilized. Either there is a cap on maximally allowed non-zero elements (due to implementation issues) or there is a cap on significance values, based on confidence levels. Both result in resetting varying amounts of non-zero values to zero in the final matrix.

Combining all these effects might result in too much information loss. Real-valued similarity measures therefore have to be compared with those operating on binary vectors, as done in Section 3.4.

One possibility is to compute the cosine of the angle between the vectors:

$$sim_{cos}\left(\overrightarrow{A},\overrightarrow{B}\right)=\frac{\overrightarrow{A}\cdot\overrightarrow{B}}{\left\|\overrightarrow{A}\right\|\cdot\left\|\overrightarrow{B}\right\|} \tag{3.38}$$

This measure also has the property that it computes a value of 0 (completely dissimilar) for any word pair not sharing any co-occurrences. This is one of the few (and the only one presented here) real-valued measures which share this property with those applied to binary vectors.

Another possibility is to compute the distance between the two points represented by A and B and to interpret the results inversely - increased distance between points means decreased similarity. However, there is an infinite number of possibilites to compute the distance between two points in a vector space. Hence the selection will be restricted to the 1-norm (L1) and 2-norm (L2), because they are the two most commonly employed:

$$dist_{L1}\left(\overrightarrow{A},\overrightarrow{B}\right)=\sum_{i=0}^{i=n}|a_i-b_i| \tag{3.39}$$

This measure is sometimes referred to as the City-Block measure or Manhattan metric, because it measures (in a two-dimensional space) how many blocks a car must pass until it gets from point A to B.

The 2-norm distance is the most intuitive distance, because it is the generalization of the n-dimensional space distance, which would be obtained by using a ruler to measure the distance between A and B:

$$dist_{L2}\left(\overrightarrow{A},\overrightarrow{B}\right)=\sqrt{\sum_{i=0}^{i=n}(a_i-b_i)^2} \tag{3.40}$$

Another perspective is to consider the task as a comparison of conditional distributions of the two words A and B (Pereira, Tishby, and Lee, 1993). The conditional distribution $p = P(*|A)$ for any other word to appear, given the appearance of A, is compared to the corresponding distribution $q = P(*|B)$ for B. The basis for any such comparison is the **Kullbach-Leibler** divergence (or relative entropy or information gain (Cover and Thomas, 1991)):

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \tag{3.41}$$

Using the divergence as a distance measure is problematic. One minor reason is that it is not symmetrical, but it is easy to take the sum of both divergences $D(p||q) + D(q||p)$, or the mean. A major problem is that it is undefined in the case of $q(x) = 0$ and $p(x) \neq 0$ - for many if not most word pairs this is the case. As one of a number of solutions such as back-off smoothing or the α-skew divergence, Lee (1997) (and later Lee (1999)) proposes to use the total divergence to the mean, or with other words, the **Jensen-Shannon** (JS) divergence. The JS divergence compares both distributions to the mean of the two distributions:

$$JS(p,q) = D\left(p||\frac{p+q}{2}\right) + D\left(q||\frac{p+q}{2}\right) \tag{3.42}$$

This modification is symmetrical, and in the case of $p(x) = 0$ or $q(x) = 0$ is simply 0. It also results in 0 for words not sharing co-occurrences and has a maximum value of $2\log 2$ (Lee, 1999). Therefore, the JS divergence is chosen as a representative of the distributional similarity measures for the evaluations below. Nevertheless, any measure comparing distributions is defined only on probability distributions, which in this case are obtainable only from direct frequency counts (the baseline co-occurrence measure). Of course, it is possible to apply the same transformation (divide each value in the vector by the sum of all values) to any vector, but in such cases the underlying mathematical model becomes meaningless. It can therefore be expected, that the Jensen-Shannon divergence performs best on the baseline co-occurrence measure.

Computing the similarity between all words does not necessarily require comparing each word with each other. Obviously, only words that are co-occurring with the co-occurrences of the input word are candidates to share any co-occurrences with it. Due to the power-law distribution of word frequency, for most words this candidate list is short. In any case, but especially for the remaining very frequent words, it is possible to use only the most significant co-occurrences to find new candidates.

This results in an algorithm with two approximation parameters which can be set such that the complexity of the entire algorithm is linear, instead of quadratic. In the experiments in this chapter the upper bound of words (each word is compared with) was 10 000. Additionally, the maximal amount of non-zero entries in the two vectors to be compared, ordered by decreasing significance, was restricted to 200. This affects only 2% of all comparisons, but decreases run-time significantly.

base	tone friendly realistic really Yet attitude coffee theatre drama musical
dice	suspense romance Cartland Haggard adventure novels fantasy realism romances private-eye
MI	Amiee novels graceful vibrant MGM tributes une Godard Femme plays
LMI	character story apparently desire outsider since powerful challenge manner audience
t-sc	life lost produced success story performance popular scene religious audience
z-sc	novel plays musical vibrant gothic Jeez stories excitement hero farce
lg	adventure desire entertaining romance life story writer talent realistic protagonist
lg2	desire life story writer talent realistic entertaining romance protagonist man
ps1	romance picaresque story desire novel dramatic writer re-creates hackneyed style
ps2	romance suspense fiction fantasy heroine Haggard private-eye adventure tales sentimental

Table 3.3.: For each significance measure the 10 most similar words according to the similarity measure *dice* of the word ‘romantic’ based on the BNC.

However it also skews the binary vector similarity measures: Since especially the baseline counts only matching non-zero values it should produce identical results irrespective of the underlying co-occurrence measure. Yet, due to the restrictions and the correspondingly differing selections of non-zero entries, the performance varies in the evaluations below.

Table 3.3 shows an example using the *dice* similarity measure to compare words based on various co-occurrence significance measures. The qualitative differences between the various measures are far less obvious in comparison to the differences between the plain co-occurrence measures in Table 3.2. Nevertheless, for example *base* and *t-sc* still produce rankings that consist of words that are not specific for ‘romantic’. However, the rankings of *dice* or *ps2* contain words that can be considered to be much more similar to ‘romantic’.

Table 3.4, on the other hand, compares rankings produced by various similarity measures based on a single co-occurrence measure (*ps2*). The difference between the plain co-occurrence ranking ‘*only*’ and the similarity rankings can be interpreted such that the former contains more syntagmatically related words. However, this distinction is not as clear-cut as for example the qualitative differences between the various co-occurrence measures.

only	love fiction novels comedy novel adventure suspense most story Cartland
base	romance suspense fiction fantasy heroine Haggard adventure tales sentimental romances
dice	romance suspense fiction fantasy heroine Haggard private-eye adventure tales sentimental
cbin	romance suspense fiction fantasy heroine Haggard private-eye adventure tales sentimental
cos	unrequited romance re-creates passionate Brooke-Rose all-consuming lust passion courtly suspense
L1	obsessions mimetic Desire inventive fairy-tale fable Sylphide time-honoured fictions portrays
L2	regaled Peniel incurably far-fetched poignant gaiety Hornblower Atwood evocative delusion
JS	Dame poems pleasures lovers fascinated Booker Fielding quest heroine distorted

Table 3.4.: Based on a single co-occurrence measure (*ps2*) the rankings of 10 most similar words to ‘romantic’ are shown for each similarity measure based on the BNC. For comparison, the ranking of the co-occurrence measure is included as ‘only’.

To summarize, it is obvious that numerous different possibilities exist first to compute co-occurrence based context vectors and second to compare such vectors. The majority of similarity measures does not depend on how the vectors were created. Others, such as the distributional similarity measures, are closely interlinked to the way the vectors are acquired and interpreted (as probabilities in this case). In any case, there are two steps: obtaining syntagmatic dependencies and then comparing global contexts.

3.4. Evaluation

According to the SIML (introduced in the previous chapter), the exact implementation of the co-occurrence significance or similarity measure does not significantly influence the type of relations extracted. However, using improper means for distinguishing between significant and insignificant co-occurrence, or measuring similarity, should strongly influence the overall quality of results. To quantify these and other related statements, the main purpose of the evaluation in this section is to:

- measure the quality of rankings obtained by using the various introduced measures
- measure the influence of corpus size

- measure the influence of word frequency
- estimate the influence of gold standard bias
- provide empirical evidence for the theoretical hypotheses concerning differences between syntagmatic and paradigmatic relations

The evaluation must therefore ensure that a sufficiently large corpus is used to prevent a possible falsification of the obtained results by running it on a different corpus. Additionally, the evaluation must be repeated on at least two different languages to provide evidence for the language independency of the underlying model. Finally, there should be several different evaluation instances (i.e. annotators, gold standards or applications) such that subjective biases can be estimated.

Throughout the related work, there is a rich variety of possibilities to evaluate algorithms that compute association strength or similarity between words. After considering several popular evaluation methods, a conclusion is drawn that for the purposes of this chapter the gold standard evaluation using a semantic net is the most suitable way to evaluate the algorithms. An overview of other possible evaluation methods with reasons of their rejection[1] particularly with regards to the above-mentioned requirements is given below.

3.4.1. Psycholinguistic experiments

Association or priming paradigms (Burgess and Lund, 1997) can be used to evaluate results of the algorithms by comparing them with data obtained from human subjects in psycholinguistic experiments. Suitable are association or priming experiments, where subjects are asked to rapidly name semantically close words in response to a stimulus word. The list of most frequently named words can then be compared with automatically obtained lists. There is a vast array of possibilities to design such experiments, but two considerations lead to a rejection of such a method in the present setting:

1. The experiments as such are very costly to do if they should be applied to large evaluations instead of small samples, as done usually. Therefore, it is probable that the evaluation results will not be representative, especially if used to evaluate a great variety of measures, as is the case in this chapter.
2. Alternatively, data from previously performed similar experiments can be used (such as Deese (1959)). But due to the cost factor their small size provides only sparse data, compared to the possibilities of the gold standard method below. They are also by no means less subjective or susceptible to

[1] However, a rejection in this context does not imply that the method concerned is useless. In fact, a test such as the TOEFL might be very well suited if similar material for other languages could also be acquired, and if relations other than synonymy were included.

regional or time biases. Additionally, they are usually restricted only to one type of relation and only one language.

3.4.2. Vocabulary tests

A vocabulary test usually comprises a question with a multiple-choice answer. If both are electronically available, the test can be used quite straightforwardly to evaluate word similarity computation methods. The test of English as a foreign language (TOEFL) has been made electronically available, and the part testing synonyms comprises 80 test items. This evaluation method has been used by many authors, such as Rapp (1996), Landauer and Dumais (1997), Jiang and Conrath (1997), Turney (2001) and Terra and Clarke (2003). The reasons for not using this kind of test in the present work are:

1. Testing against only 80 items poses the problem of whether the results are representative. In such a case overtraining (by fitting thresholds) can occur very fast.

2. This test tests only synonymy. Since the intention is twofold: not revealing the quality of the measures, but also whether there are preferences for different types of relations, measuring only one of the many possible linguistic relations does not suffice.

3. Finally, this test is only available for one language, with no equivalent (with respect to the key words) in other languages.

3.4.3. Application-based evaluation

Application-based evaluation is an indirect method of evaluating results of a knowledge extraction algorithm by putting the extracted knowledge into use and observing the performance of the application using this knowledge. As described by Curran (2003), scientific applications can be smoothing language models (Dagan and Church, 1994), word sense disambiguation (Dagan, Lee, and Pereira, 1997) or Information Retrieval (Grefenstette, 1992). It would also be possible to use any kind of (non-scientific) software application for this purpose by replacing words (in that application) with the computed similar words and observing the user's behaviour. This kind of evaluation is not used in this chapter, because:

1. It would be relatively easy to use this method for testing synonymy. Any other linguistic relation would, however, be somewhat difficult to implement, and therefore it would again be hard to show what the extraction algorithms actually extract.

2. The evaluation results would always be influenced by other factors and other error sources, and also be indirect.

3. Such an evaluation further presupposes an existing framework or application in which it might be used, as well as a set of users employing this application. Neither an application nor the users were available.

3.4.4. Artificial synonyms

One of the most interesting approaches to evaluating automatic extraction algorithms is to use artificial items. The idea for testing synonymy is to randomly replace a part of the occurrences of a word with a pseudoword while leaving the other part unchanged. The two words (the pseudoword, and the original word) should then be perfect artificial synonyms. This enables to measure how often the pseudowords are extracted as synonyms of the retained words.

Inspired by the artificial pseudowords introduced for word sense disambiguation evaluation as in Gale, Church, and Yarowsky (1992) and Schütze (1998), this method has been successfully used in various studies, including the one by Grefenstette (1994). Problems such as the unnatural pureness of ambiguity (in the disambiguation task) with this kind of evaluation have been investigated by Gaustad (2001) and Nakov and Hearst (2003.). However, the two reasons for rejecting any such evaluation method, were of another kind:

1. Though producing artificial synonyms is simple, it is hard to create artificial antonyms, meronyms or other linguistically related words. Consequently, it would again be difficult to measure preferences of various measures for different linguistic relations.

2. This type of evaluation does not allow to directly compare the performance of co-occurrence significance measures with similarity measures.

3.4.5. Gold standards

The finally chosen evaluation method is known as evaluation using ‘gold standards’. The idea is to select a lexical knowledge source, thesaurus or dictionary, and test the results of the algorithms against this knowledge source. There are, however, several different approaches to this.

The global approach was applied by Grefenstette (1994) in his SEXTANT system. A number of word pairs, which the algorithm computed as most similar, are compared against the equivalent of synonym sets in Roget’s Thesaurus (Roget, 1946). Additionally, the probability of two randomly selected words to collide in a synonym set in Roget’s Thesaurus is computed. Because the number of correctly computed pairs is much higher than the number prediced by random collision, this shows the quality of his algorithm. This kind of interpretation of results by taking the globally most similar word pairs and evaluating them is basically the same approach as used by Church et al. (1989) for the extraction of collocations.

However, this approach leaves many questions open. First, it does not become clear how well the evaluated algorithms perform for a larger variety of words. Such small sample sets also cannot be taken as representative for the entire language. But even when taking larger samples, another problem is that the quality of the globally most similar words is highly dependent on problems with the particular measure taken. For example, the ability of the corresponding measure to cope with extremely low or high frequency of a given word (as is the case with mutual information based algorithms). Also, the most of the globally most significant or similar word pairs are less interesting, because most applications, such as clustering, use the local rankings of words.

In the local similarity comparison approach, similarity between two words as acquired from a thesaurus such as WordNet is compared to similarity between the same two words as computed by an algorithm. Similarity from the thesaurus is obtained by using a distance measure based on the hierarchy of the thesaurus modified by various weighting factors. This allows to obtain a similarity ranking between an input word and all other words contained in the thesaurus. Clearly, the advantage with this approach is that a similarity value produced by an algorithm can be judged in any case, even if the two words computed as similar are highly unrelated. However, the disadvantage is that it again becomes hardly possible to measure preferences for various relation types, since they are all used to compute the one 'gold standard' similarity value.

The local direct retrieval approach, also called the Information Retrieval approach (Curran, 2003), is to choose a set of words and observe which other words the algorithm (to be evaluated) computes as similar. This can be viewed as an Information Retrieval task: The input word is the search query, and each word computed as similar is either correct or wrong. Verifying the correctness of a word can be done with an electronically available thesarus. This allows to directly measure precision P and recall R, mean average precision MAP and any other common Information Retrieval evaluation value. Such evaluations are then based on how many correct words were among the first x similar words, and how many of those were synonyms, antonyms, hyperonyms etc., according to the knowledge source. This notion of precision and recall differs from the ones used in Weeds and Weir (2005), where it is used to define the overlap between various similarity and co-occurrence vectors.

The lower a correct word is in a ranking, the less valuable the correct hit is considered. Curran (2003) proposes to use the mean average precision, i.e. inverse ranks of matching synonyms (according to WordNet) to give a representative score for a computed ranking of similar words for the input word. Subsequently averaging these rankings over all input words yields a precise score of the quality of a certain algorithm. Curran manually selected up to 300 nouns for a detailed analysis of the algorithms described. He primarily used the synonymy relation, since he was interested in finding out how well the algorithms computed similar words.

A modified local direct retrieval approach. Equivalently to earlier work (Bordag, Witschel, and Wittig, 2005), avoiding the preference of one relation (synonymy) over others, the local direct retrieval approach is modified in the evaluations below. Each relation in the gold standard defines its own retrieval task with a set of correct words a given input word stands in relation with. Each task is named after the relation, such as *antonym* or *synset*. Additionally, all words related with the input word are counted as correct in a separate retrieval task, called *total*.

Given that such thesauri exist in sufficient size for several languages (some languages having several different ones), all conditions formulated at the beginning of this section are met. The minimal variety of two languages is given by applying the approach to English and German. Using the BNC for English and a similar sized newspaper corpus for German does not provide absolute security that the obtained results cannot be falsified. However, the evaluation of corpus size influence below indicates that at least it is extremely improbable. Additionally, statistical significance tests show that even small variations in the obtained results are statistically significant. Hence, the requirement of sufficient corpus size can be assumed as given.

Finally, using GermaNet and the Annotation Project as gold standards for German and WordNet for English reveals the biases within any of the three gold standards. In fact, the evaluations below show that only the performance is affected by the choice of language, corpus size and gold standard. Neither is preference for relations or relation types, nor the relative performance of measures affected by these three parameters.

3.4.6. Experimental design

Given the modified local direct retrieval approach, the experimental setup consists of a set of evaluation measures, a set of languages represented by corpora, a set of gold standards and a set of specific experiments.

Evaluation measures

The above-mentioned notions of precision, recall and mean average precision are defined as follows:

Precision $P(w)$ for an input word w is defined as the number of correct words found, divided by x, the total number of words computed as similar[2]. A correct word is one standing in a relation with the input word, according to the knowledge source. That is, if 5 synonyms are to be found, $x = 50$ and the algorithm finds 2, then the overall precision is $\frac{2}{50}$ for this word.

Recall $R(w)$ for one word w is defined as the number of correct words found within the x most similar words divided by the number of words standing in a relation, according to the knowledge source. That is, if there are 5 synonyms to be found, $x = 50$ and the algorithm finds 2 synonyms, then overall recall is $2/5$ for this word.

These precision and recall values can be averaged over all input words in order to obtain global precision P and recall R scores which represent the overall performance of the algorithm. Since for recall finding 2 out of 4 synonyms for one input word is 'worth more' than finding 2 out of 10 for another word, one algorithm can have higher recall and lower precision than another.

Often, precision and recall are combined into a single so-called **F-value**, which is the harmonic mean of precision and recall:

$$F = \frac{2 \cdot P \cdot R}{P + R} \tag{3.43}$$

Alternatively, it is possible to compute the **mean average precision** (MAP) as the mean over all input words of the average inverse rank of each correct hit. If r_i is the rank of a relevant word $c_i \in C$ of all relevant words C_w (for a given input word w), then the average inverse precision for that word $AP(w)$ is the sum of the inverse ranks divided by the minimum of the number of elements in C_w and the number of words Q_w the algorithm retrieved:

$$AP(w) = \frac{\sum_{c_i} \frac{1}{r_i}}{min\left(|C_w|, |Q_w|\right)} \tag{3.44}$$

The MAP is then the mean over all input words $w \in W$:

$$MAP = \frac{\sum_{w \in W} AP(w)}{|W|} \tag{3.45}$$

This method emphasizes returning relevant words higher ranked, but also generally returning more relevant words. Hence it combines both precision and recall, but unlike the F-value it also includes the quality of the rankings. For example, there are two words A and B for which an algorithm computes 10 'similar' words. For A, two out of six relevant words are found at the positions 3 and 5, whereas for B only one out of six relevant words is found, but at the first position. In the first case this results in $AP(A) = \frac{\frac{1}{3}+\frac{1}{5}+0}{min(6,10)} = 0.09$, whereas in the second case this results in $AP(B) = \frac{\frac{1}{1}+0+0}{min(6,10)} = 0.17$. MAP often results in very low scores, but

[2] x usually has values like $x = 5, 10, 25, 50$ or even 200

was found to be a reliable method for distinguishing algorithms that produce very similar results in IR.

It is necessary to take the minimum of the number of returned words and the number of relevant words to always ensure the possibility of reaching the maximum score of 1.0, even if the result set is smaller than the set of relevant words. However, the difficulty here is that the measured results can be artificially improved by simply restricting the rankings to smaller sizes. In the example above, for A and B restricting output sizes to the three top-ranked resulting words 'improves' the $AP(A) = 0.11$ and $AP(B) = 0.33$. In a similar manner, it may also artificially worsen the results. Restricting to a ranking of the top two ranked words results in $AP(A) = 0$ and $AP(B) = 0.5$. Generally, the artificial improvement effect occurs much more frequently than the worsening effect. This is also the reason why in IR, when this measure is used, the answer sets are always fixed at 1 000 results, even if this means pretending that there are that many relevant documents. For the same reason, in the experiments in this chapter all answer sets were restricted to the arbitrarily chosen number of 100 items.

For the algorithms in this chapter such a threshold is unproblematic, because for any input word it is possible to guarantee at least 100 most relevant output words. For the algorithms in Chapter 6 this is impossible. There, algorithms are compared to several baselines. Some of these algorithms inherently cannot guarantee to produce a fixed amount of output words (or any output words at all). Therefore the notion of **size-adjusted baseline** is used. Given the output of an algorithm, a baseline is size-adjusted such that for every single input word the ranking is cut to the exact size of the ranking produced by the algorithm for that word. This allows a fair comparison of the ranking of the algorithm with the ranking of the baseline.

The size-related problem, and the fact that MAP combines aspects of precision, recall and ranking in a single number leads to a more difficult absolute interpretability of the figures. However, it is very useful to find relative preferences for rankings. Therefore, in this chapter mostly the plain precision and recall values are used as evaluation measures, whereas in Chapter 6 mostly MAP is used.

3.4.7. Corpora and gold standard

The evaluation itself is performed on two corpora: one for English and one for German. The first corpus is the raw text from the British National Corpus (BNC). The second one is a small part of the 'Projekt Deutscher Wortschatz' (Quasthoff, 1998) corpus that currently contains approximately 35 million German sentences. To make the quality of the results roughly comparable to results obtained from the BNC, which contains 100 million running words, the part of the German corpus for the evaluation was chosen to be of the same size. This results in randomly selecting about 5.95 million sentences from the main corpus, see Appendix A. However, for the corpus size influence experiment below, several subcorpora were created with 1 million sentences increase steps (drawing non-randomly, in chronologic order of the sentences) up until size of 16 million sentences.

Unlike the representantive BNC, the German corpus consists primarily of newspaper texts from the most popular German daily newspapers. Since this evaluation does not try to promote absolute values, but rather relative statements such as '*algorithm A* performs significantly better than *algorithm B*', the actual quality of the used corpus is not of decisive significance, as long as it remains possible to compare the results to the knowledge sources.

Three gold standards are used, one for English and two for German. For English it is WordNet (Miller, 1990; Fellbaum, 1998), for German GermaNet (Hamp and Feldweg, 1997; Kunze and Wagner, 1999) and the Annotation Project (see Appendix B). GermaNet is wider known and is supposed to resemble WordNet, both in contents as well as in structure. The Annotation Project has not yet been made public, hence it is mostly unknown. It was created using the entire corpus, parts of which were drawn for the present evaluations, so it is more probable to be of the same domain and genre. GermaNet, on the other hand, was mostly created without using corpus-based methods.

Not every word in the corpus is contained in the gold standards and conversely, not every word in the gold standard is contained in the corpora. Additionally, some words contained both in the corpus and the gold standard have an insufficient frequency to allow any conclusions about its usage. Therefore, the following restrictions were put in place:

- For both corpora, only the 100 000 most frequent words (100K rule) were used as input words
- This set was further restricted to words present in the corresponding knowledge source. For WordNet that leaves 35 966 words, for GermaNet 21 686 and for the Annotation Project 40 857.
- The output word sets were restricted to words present in the corresponding knowledge source, and from the remaining only the top 100 ranked words were considered for the evaluation. Note that this in unison with the previous rule excludes all inflected word forms from two of the three evaluations, because they are contained in the raw corpora, but not in the knowledge sources (WordNet and GermaNet).

These restrictions have the effect that not all annotated word pairs in the knowledge sources can be found. Table 3.5 shows the resulting amounts of valid words for each knowledge source. The table also provides the remaining number of words after the frequency filtering (100K overlap) and for a small set of relations the resulting amount of word pairs annotated with that relation.

	Annot	GermaNet	WordNet
total words	75 728	52 620	146 212
100K overlap	45 343	21 686	35 990
100K word pairs	342 180	701 208	2 130 452
100K avg. rel. words	7.5	33.3	59.9
distinct relations	58	13	26
cohyponyms	119 245	608 550	1 799 727
hyper(o)nyms	67 782	103 038	209 333
synonyms	31 478	21 281	83 691
n adj typ. property	16 690	n.a.	n.a.
n v typ. obj. of	14 295	n.a.	n.a.
part/consists of	13 991	10 098	17 259

Table 3.5.: Statistics of the Annotation Project compared to GermaNet. Counts for relations restricted to the 100K rule. All relations are assumed to be directed, meaning that counts for symmetrical relations are doubled.

GermaNet and WordNet are comparable in that GermaNet is about half the size of WordNet not only with regards to the amount of words, but also with regards to the annotated relations. In the Annotation Project, the total number of different word forms is larger than in GermaNet (which contains only lemmas), but smaller than WordNet. However, the density of annotated relations between these words is much sparser.

Nevertheless, for each knowledge source the large amount of 'attempts'[3] makes the results statistically robust.

All evaluation measures such as MAP can either be computed for each relation separately, or for a selection of relations at once (for example all hierarchical paradigmatic relations), or for all relations. The SIML in the previous chapter introduces a basic difference between syntagmatic and paradigmatic relations. Therefore, it would make sense to also collect the various specific relations found in the knowledge sources into these two (or more) relation types. However, both WordNet and Germanet lack syntagmatic relations, because they were not considered 'semantic' and the main purpose of them is to encode semantic relations. In the Annotation Project, on the other hand, such relations are encoded. Hence, for WordNet and GermaNet only one additional relation is introduced: the *total* relation which subsumes all other relations. This allows to measure the overall performance of an algorithm in that for any given input word any annotated word is counted as relevant. For the Annotation project, the following classes of relations are introduced additionally to the *total* relation:

- **Syntagmatic**: multi word expresions, typical object/instrument, ...

[3] i.e., the performance of each algorithm is tested on several dozens of thousands of different words

- **Symmetric paradigmatic**: cohyponyms, antonyms, synonyms, ...
- **Hierarchic paradigmatic**: hyperonyms, meronyms, ...
- **Derivates**: adjectives from verbs, adjectives from nouns, ...
- **Other**: CEO of, title of, cause, ...

These classes are also used in Chapter 6. However, in this chapter *derivates* and *other* are omitted, because of their low relevance and low total number of encoded word pairs.

3.5. Experiments

3.5.1. Influence of corpus size

Experiment 1 is devoted to determine the extent to which corpus size influences rankings obtained from co-occurrence significance measurement results. Due to time constraints no attempts were made to further determine the correlation between co-occurrence or similarity measure or type of relation and corpus size. For the latter, the other experiments below provide sufficient evidence to rule out any possible interaction. However, it would be interesting to test the former correlation. But that would imply running all following experiments on all corpus sizes, where each larger corpus requires nearly linearly more time and the 88 runs (11 co-occurrence measure combinations times 8 similarity measures[4]) for the two BNC sized corpora already take approximately 3 weeks on one 3GHz personal computer.

The largest overlap between the corpus and the gold standard was found between the Annotation Project and the German corpus (40 857 words), therefore this combination was used. The experimental corpora were created by drawing sentences in one million steps from the larger German newspaper corpus 'Der Deutsche Wortschatz'. The sentences were drawn in chronologic order, according to their date of publication. The smallest corpus contains one million sentences and the largest corpus consists of 16 million sentences. MAP was used as the evaluation measure - precision and recall yield identical observations in this case.

Figure 3.3 shows the MAP of the (again, arbitrarily chosen) *lg2* co-occurrence significance measure on corpora with growing sizes, measured with the *total* relation. Additionally, measurements for the *syntagmatic, paradigmatic* and *hierarchical paradigmatic* are provided to indicate, whether preferences for certain types of relations exist.

As the figure shows, increasing corpus size certainly improves the similarity between rankings computed by the algorithm and relations annotated in the gold

[4] Because most similarity measures share many steps, this was again slightly optimized by computing all similarity measures for one underlying co-occurrence measure in one pass

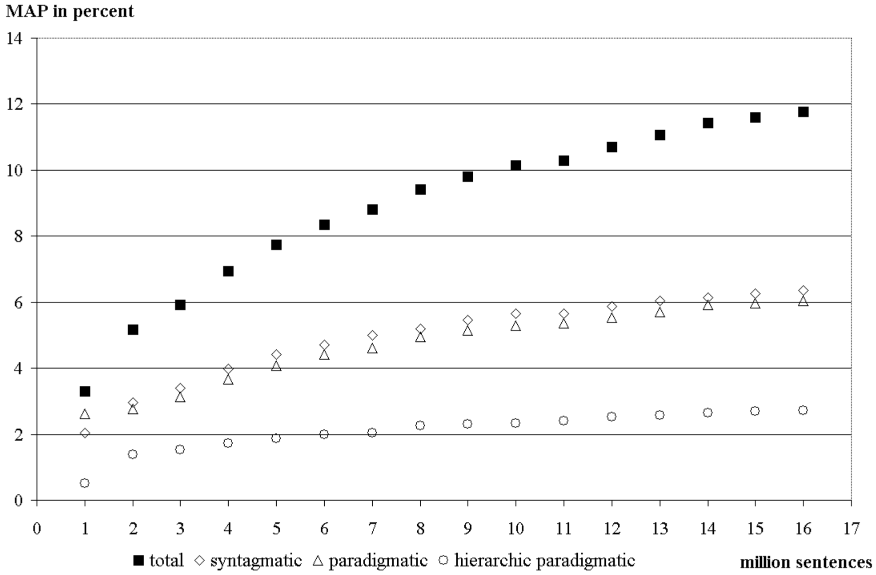


Figure 3.3.: Influence of corpus size on quality of most significant co-occurrences for 100 000 most frequent words using the Annotation Project as gold standard.

standard. The increase is logarithmic - depicting the same figure scaled logarithmically for the corpus sizes results in a line. There is no clearly observable correlation between corpus size and preferences for types of relations.

One interesting observation is that the BNC-size (roughly 5 million sentences) is 'not quite enough'. Doubling this 10 million might improve algorithms based on co-occurrence observations by approximately 30%. However, the graph also shows that surpassing the 20 million sentences limit should yield only small improvements, unless the increase is by an order of magnitude. However, the graph also shows that trying to observe significant co-occurrences is sensible only on corpora with a certain minimum size not much smaller than that of the BNC.

3.5.2. Influence of measures

One of the main goals of the evaluation in this chapter is to gauge the relative performance of the various co-occurrence and similarity measures. Hence, Experiment 2 tests and compares all measure combinations. For both corpora, all significant co-occurrences according to the 11 significance measures described above are extracted. The threshold for significance was set such that for each of the 100 000 most frequent words at least 100 co-occurrences contained in the corresponding gold standard were extracted. Then, on each data set all 8 similarity measures were computed, comparing all (potentially similar) words with each other. Again, it was ensured that each of the 100 000 most frequent input words (or, rather those also in the corresponding gold standard) have at least 100 most similar (output) words.

	base	dice	tan	MI	frMI	t-sc	z-sc	lg	lg2	ps1	ps2
only	0.52	1.79	1.79	1.39	1.48	0.42	1.75	1.67	1.67	1.75	1.77
base	1.11	1.85	1.85	1.62	1.80	1.02	1.68	2.10	2.10	2.10	2.58
over	1.10	1.76	1.75	1.53	1.23	0.98	1.34	1.37	1.32	1.56	1.86
dice	1.12	1.89	1.89	1.63	1.83	1.04	1.72	2.15	2.15	2.13	2.57
cos	0.68	1.88	1.88	1.70	2.00	1.15	1.57	1.89	2.13	2.16	2.33
cbin	1.12	1.88	1.88	1.62	1.83	1.05	1.69	2.13	2.13	2.10	2.50
L1	1.21	0.92	0.88	1.16	1.32	1.23	0.86	1.13	1.19	1.15	0.94
L2	1.17	0.97	0.97	1.09	1.50	1.22	0.86	1.37	1.52	1.33	1.30
JS	1.23	0.88	0.70	0.90	1.15	1.16	0.78	1.06	1.06	0.84	0.88

Table 3.6.: MAP in % for BNC measured on WordNet (using the *total* relation) for all measure combinations.

The result is a matrix with 11 co-occurrence measures times 8 similarity measures with an additional row *only* for the co-occurrence measures themselves. Table 3.6 shows the MAP values obtained by using the BNC with WordNet as the gold standard. The first general observation is that all values are extremely low, with the highest being only at 2.58%. Compared to that, common values for MAP in IR range within the 20% to 30% interval, depending on the difficulty of the task.

To properly judge the significance of variations within a group of means the Scheffé test (ANOVA post-hoc test) is used. The following discussion of the various differences and similarities is based on several such tests (all with $\alpha = 0.05$), further detailed in Section 3.5.6. In short, such a test was run for each row and column and additionally for all means at once. The results do not differ with respect to language, knowledge source, or evaluation measure used. Generally speaking, any difference smaller than 7% of the absolute values is insignificant. For example, in Table 3.6 the measure combination *ps1_base* (MAP=2.10) achieves only 81% of the quality of *ps2_base* (MAP=2.58). This difference of 19% was found to be significant, whereas the difference to *ps2_cos* (MAP=2.13), which is less than 2%, was found to be insignificant.

Tables 3.7 and 3.8 show the same experiment for the German BNC sized subcorpus measured against GermaNet and the Annotation Project. The figures are nearly doubled in the first case and almost quadrupled in the other case over the experiments using BNC and WordNet. This is despite the fact that the German subcorpus is of the same size as the BNC, but of a worse quality in that it is not representative or balanced. Additionally, WordNet contains nearly twice as many words with annotations as compared to GermaNet.

The possible reasons for this surprising finding are manifold. One might be the differing sizes of the knowledge source - apparently the larger the knowledge

	base	dice	tan	MI	frMI	t-sc	z-sc	lg	lg2	ps1	ps2
only	0.85	2.26	2.26	1.78	1.81	0.71	2.15	2.02	2.02	2.10	2.13
base	2.07	3.33	3.34	2.80	3.16	1.89	2.71	4.04	4.07	3.80	4.66
over	1.91	3.25	3.26	2.71	2.08	1.79	2.44	2.50	2.49	2.85	3.17
dice	2.11	3.38	3.38	2.81	3.22	1.93	2.79	4.12	4.15	3.82	4.58
cos	2.08	3.35	3.35	2.92	3.55	2.25	2.62	3.47	3.67	3.48	3.93
cbin	2.12	3.37	3.37	2.80	3.22	1.94	2.74	3.97	3.99	3.64	4.34
L1	2.23	1.78	1.81	2.19	2.09	2.31	1.80	2.03	2.09	1.98	1.66
L2	2.13	1.94	1.90	2.09	2.82	2.27	1.74	2.59	2.71	2.35	2.30
JS	2.19	1.92	1.91	1.85	2.28	2.12	1.73	1.92	1.98	1.68	1.58

Table 3.7.: MAP in % for the German subcorpus measured on GermaNet (using the *total* relation) for all measure combinations.

source, the more an algorithm ‘misses’ according to MAP, which includes recall. This intuition is supported by the fact that the precision values (independent of recall) in Tables 3.10, 3.11 and 3.12 do not differ nearly as much.

Another reason might be the flectivity of German, compared to English. Whereas the English corpus consists only of roughly $660K$ different word forms, the German corpus contains nearly four million different word forms, see Appendix A. Hence, any particular word form has a much larger chance of being less ambiguous in German.

The core observation though is that the various measures produce rankings that differ strongly in their quality. The relative differences between the measures remain the same (with one exception), despite using three entirely different knowledge sources and two different languages. This, with the Scheffé test, ascertains that the differences are not due to chance. The exception is that in the case of using the Annotation Project the co-occurrence significance rankings (*only*) outperform the rankings produced by following similarity measures, which is discussed below in Section 3.5.7. However, in theory (and according to the SIML), it is meaningless to compare co-occurrence rankings with similarity rankings for pure performance, because they represent principially different types of relations. For example, a gold standard consisting solely of a dictionary of idiomatic expressions would fit much better to co-occurrence data. Then, no matter how good a similarity comparator works, co-occurrence rankings will always achieve a much higher overlap.

The true reason for the observed preference is the annotation process used in the Annotation Project. The relevant part of the process consists of showing the annotator a word and several most significant co-occurrences of it to annotate. The annotation program (see Appendix B) then allows to easily select relations between the presented word and it’s co-occurrences or other potentially related words. The other potentially related words were produced by metarules the annotators could define, such as transitivity for relations (for example cohyponymy). Nevertheless,

	base	dice	tan	MI	frMI	t-sc	z-sc	lg	lg2	ps1	ps2
only	3.68	11.47	11.47	8.89	9.78	3.10	11.40	11.23	11.24	11.92	12.12
base	5.02	8.84	8.84	5.74	8.15	4.61	6.40	9.91	9.97	9.72	11.47
over	4.56	8.20	8.20	5.38	5.21	4.32	5.12	5.52	5.53	6.64	6.91
dice	5.14	8.93	8.93	5.75	8.29	4.73	6.49	9.99	10.06	9.65	11.02
cos	4.28	8.70	8.67	5.95	7.94	5.12	6.48	7.73	8.24	8.48	8.71
cbin	5.16	8.88	8.88	5.72	8.27	4.75	6.33	9.67	9.73	9.26	10.52
L1	4.18	4.00	3.81	3.93	3.88	4.44	2.90	3.29	3.41	3.25	2.67
L2	3.98	4.00	4.02	3.70	4.70	4.38	2.96	4.18	4.39	3.92	3.82
JS	4.05	3.65	3.65	3.08	3.60	4.10	2.93	2.78	2.85	2.35	2.23

Table 3.8.: MAP in % for the German subcorpus measured on the Annotation Project (using the *total* relation) for all measure combinations.

approximately two thirds of all automatically proposed words were co-occurrences, and only 10% automatically proposed words were ‘proposed’ by similarity measures. This is due to the software for the similarity computations becoming complete only towards the end of the Annotation Project. Hence, it is not surprising to retrieve this bias using the Annotation Project as a knowledge source.

3.5.3. Upper bounds

Given the extremely low MAP values (and only marginally higher precision values), it is necessary to assess the maximal measurable performance. For one, the chosen evaluation method has no means to distinguish between true negatives and false negatives. That is, the word pair *apple - strawberry* not being in the knowledge source does not preclude the existence of any relations between them. Undoubtedly, neither of the knowledge sources comes close to contain all relations between all words, as is demonstrated in Table 3.9. Hence, an unknown amount of word pairs which do stand in a relation are extracted by the algorithms, but counted as wrong, hence false negatives. One possibility to at least estimate the amount of false negatives is to measure the ‘inter-annotator agreement’ between two knowledge sources as described in Section 6.2. The approximate conclusion to be drawn from comparing GermaNet and the Annotation Project is that the maximal achievable performance is about 60% MAP.

This comparison is problematic, because it provides no information regarding the number of unannotated word pairs, because it only compares to incomplete sets of annotations. For example, it could very well be the case that both knowledge sources contain only 10% of all word pairs not standing in one of the relations that are annotated. Hence, if an algorithm correctly exctracts all 100% of all word pairs, but does so at the cost of additionally extracting a similar amount of wrong word pairs in the process, then the true performance (in MAP) of that algorithm

input	output words	P
apple	apples(X) strawberry pudding pear(C) fruit peach(C) pears(X) lime pies plum(C)	3/8
Europe	Japan countries(X) Britain(M) America continent(H) Europeans(X) European nations(X) Germany(M) Western	3/6
Pluto	Uranus(C) Jupiter Neptune Saturn planets(X) Venus Aries planet Mars(C) Moon	2/9
PhD	MPhil(X) postgraduate MLitt graduate undergraduate MSc students(X) undergraduates(X) university degrees(X)	0/6
nail	nails(X) nailed(X) plasterboard remover polish screws(X) out-of-bounds(X) hammer bullet fixing(H)	1/6
total		25.3%

Table 3.9.: Precision measured for several example words and their 10 most similar words according to *ps2_dice*. (H = hyperonym/hyponym, C = cohyponym, M = meronym) Instead of removing words not in WordNet (and filling up the ranking to contain 10 words again), they are marked with X in this example.

is about 50% (assuming that the relevant words are evenly distributed among the rankings). Yet, measured against the small knowledge source only a performance of at most 5% is measured.

Consequently, the most important question is: how much of the total possible information is contained in WordNet, GermaNet and the Annotation Project? However, it is nearly impossible to try to gauge that. For example, if only the 100 000 most frequent words from the BNC are taken, then only 35 966 are also contained in WordNet. Using this as a simple estimate that WordNet therefore contains only 35% of all possible knowledge, then the hypothesized algorithm would have a measurable performance of 17%. Of course, such simple estimates are very imprecise, because out of those 100 000 most frequent words many are inflected word forms and thus do not count.

On the other hand, it is possible for the knowledge source to contain relations either not directly observable in the corpus the algorithm is applied to or which occur only very rarely. This would especially affect MAP based measurements. For example, if out of 60 relevant output words for an input word A only 30 have a sufficient frequency to allow statistic observations about their occurrences, then the maximal measurable MAP is only 50% for any algorithm relying on co-occurrences. Section 3.5.8 below provides additional insights into how the various measures cope with low frequent input words.

All this leads to one conclusion. An algorithm with a measurable precision of 10% (of the 5 most similar words) does not find only one relevant word for every second input word. The 10% are true positives, but the remaining 90% are certainly not all true negatives. Table 3.9 demonstrates this fact very clearly. Hence, the reported performance values cannot be taken as absolute quality measures of the measures or algorithms. But they are useful as a method to gauge the relative performance of one algorithm or measure over another.

3.5.4. Performance of co-occurrence measures

To properly compare the quality of co-occurrence measures it is possible to first examine the results obtained from similarity measures based on co-occurrence rankings.

Computing word similarity based on the co-occurrence significances of the second Poisson approximation $sig(A,B)_{ps2}$ significantly outperforms the other similarity measure combinations in all three experimental setups. The first approximation ($\ln k! = \ln \sqrt{2\pi k} \left(\frac{k}{e}\right)^k$) is very precise, but it performs statistically insignificantly worse than log-likelihood. In contrast, the simplified form of the second approximation ($\ln k! = k \ln k$) introduces a systematic error which results in an increasing positive discrepancy for larger k (i.e. n_{AB}). This can be paraphrased in that the second Poisson measure systematically overrates larger co-occurrence frequencies over small ones. The practical effect can be explained with the following example.

Assuming that word A has two potential co-occurrences B and C, where B co-occurs with A only slightly less often than C, but the frequency of B is much lower than C. The systematic error then causes C to have a greater chance of being ranked higher than B, despite the common effect of the significance measures: the smaller the single word frequency of a word pair and the higher the co-occurrence frequency, the more significant the observation. This systematic error is clearly the reason for the significant deviation of its performance compared to the precise approximation and the original log-likelihood measures. The reason this deviation is positive might be related to the fact that the less frequent a word, the less probable it is contained in the knowledge source used for the evaluations.

However, the plain co-occurrence rankings are not as straightforward. First, there is a group of measures found to not differ significantly from each other by the Scheffé test (Table 3.13): the dice and tanimoto measures, z-score, both log-likelihood variants as well as the Poisson approximations. Apparently the MAP values do not suffice to reveal the underlying differences, because using the results of the same measures for similarity computations results in four different groups of measures. There, the worst group with respect to the performance of the corresponding measures consists of the *t-sc* and the *base*. The best group contains exclusively *ps2*, significantly differing from the second-best group which contains the remaining log-likelihood based measures (*lg*, *lg2* and *ps1*). The remaining two groups appear to be less conclusive, drawing a thin line between the mutual information variants together with the *z-sc* on the one hand and the *tan* and *dice* on the other hand.

With respect to co-occurrence measures, it can be concluded that the gold standards evaluation using the MAP values is not sufficiently varied to detect the decisive differences. However, using an average frequency comparison of extracted words makes it apparent that Dice and Tanimoto tend to rank low-frequency words higher than the more balanced log-likelihood based measures. The following example illustrates the differences. The input word A ($n_A = 1\,000$) in a one million sentences corpus co-occurs 100 times with B ($n_B = 500$) and 150 times with C ($n_C = 2\,000$). According to the dice coefficient $sig(A,B)_{dice} = 0.133$ and $sig(A,C)_{dice} = 0.1$, which means that the less frequent B is more significant than C. Contrary to that, the second Poisson approximation ($sig(A,B)_{ps2} = 429$ and $sig(A,C)_{ps2} = 497$) results in an inversed ranking. The reason this adversely affects similarity computations is sparsity. The less frequent a word in a co-occurrence vector, the less probable it produces a match with a co-occurrence vector of another word.

	base	dice	tan	MI	frMI	t-sc	z-sc	lg	lg2	ps1	ps2
only	1.95	8.35	8.36	6.35	6.82	1.47	8.08	7.79	7.79	8.13	8.26
base	5.37	7.95	7.95	6.09	7.77	4.99	7.14	8.79	8.78	8.65	10.29
over	5.37	7.56	7.58	5.90	5.27	4.80	5.89	5.71	5.50	6.27	7.84
dice	5.43	8.12	8.14	6.12	7.89	5.06	7.35	8.98	8.98	8.82	10.37
cos	3.29	8.09	8.05	6.30	8.49	5.63	6.90	7.96	8.80	8.82	9.16
cbin	5.44	8.12	8.14	6.10	7.93	5.09	7.29	8.96	8.94	8.77	10.17
L1	5.88	3.74	3.53	4.41	6.23	5.97	3.67	5.30	5.53	5.32	4.23
L2	5.70	3.84	3.83	3.93	7.07	5.93	3.52	6.36	7.05	6.17	6.03
JS	5.68	3.80	2.82	3.37	4.81	5.49	3.52	4.21	4.16	3.36	3.54

Table 3.10.: Precision in % for the 5 most significant or similar words for the BNC measured on WordNet (using the *total* relation) for all measure combinations.

The sole difference between the two log-likelihood variants *lg* and *lg2* is that the second method differentiates between significant inhibition and attraction. This difference does not affect the measured quality of the co-occurrence rankings, and has only an insignificant effect on the subsequent similarity rankings. This is because only relatively few words have a sufficiently high frequency. While approximately 4.7% of all observed co-occurrence pairs were found to be negative, only for few word pairs the negative significance is was high enough (before setting it to negative) to interfere with the 100 most significant co-occurrences.

As expected, the overrating of infrequent words renders the rankings of the mutual information measure nearly useless. While the lexicographer's modification helps, it does not suffice to reach the quality of the log-likelihood based measures (including the Poisson approximations).

As could have been expected, the worst measure is clearly the t-score *t-sc*, be-

	base	dice	tan	MI	frMI	t-sc	z-sc	lg	lg2	ps1	ps2
only	3.47	8.59	8.59	7.12	6.90	2.84	8.48	7.71	7.71	8.06	8.14
base	7.44	10.98	10.99	8.78	10.40	6.92	9.18	12.79	12.87	12.11	14.63
over	7.25	10.88	10.91	8.77	8.06	6.89	8.67	9.21	9.17	9.71	11.18
dice	7.55	11.24	11.26	8.92	10.59	7.03	9.57	13.17	13.24	12.45	14.70
cos	7.99	11.18	11.17	9.26	11.86	8.33	9.05	11.27	11.87	11.36	12.33
cbin	7.59	11.25	11.27	8.92	10.68	7.08	9.57	13.15	13.21	12.36	14.28
L1	8.64	5.92	5.92	7.02	11.61	9.12	6.54	8.68	8.90	7.85	6.74
L2	8.22	6.46	6.34	6.61	10.64	8.75	6.15	9.74	10.23	8.81	8.65
JS	7.98	6.82	6.83	5.88	7.69	7.92	6.26	6.49	6.53	5.46	5.19

Table 3.11.: Precision in % for the 5 most significant or similar words for the BNC measured on WordNet (using the *total* relation) for all measure combinations.

cause it is not really applicable to the task of observing significant co-occurrences. The fact that the *t-sc* does not even outperform the baseline supports this conclusion. This is due to a simultaneous strong preference for small co-occurrence frequencies and high-frequency words: only then is the difference between observed and expected large while the observed value is small. Hence, the *t-sc* ranks high-frequency, but rarely co-occurring words highest, which cannot be really useful.

The reason this measure was used frequently throughout related work is that often the globally most significant co-occurrences are desired, instead of local co-occurrences. An analysis of the applicability of the discussed measures to globally most significant co-occurrences is not in the scope of this chapter. However, for most applications relying on word similarities for their clustering or other comparison-based operations the locally most significant rankings are of interest.

Table 3.10 additionally provides the precision measured on the BNC with WordNet for only the 5 most significant co-occurrences or most similar words. This provides more intuitive numbers about the performance of the algorithms and corresponds more closely to the automatic thesaurus creation task. Table 3.12 shows the same precision for the German subcorpus and the Annotation Project. The values can be interpreted so that for example the second Poisson approximation extracts one relevant word (true positive) for two out of three input words.

As mentioned above, the fact that the difference between the precision values is not as large as between the MAP values between German and English further supports the hypothesis that the low MAP numbers in English result of the large size of WordNet, as opposed to the smaller size of GermaNet. Generally, for any word WordNet contains twice as many relevant words than GermaNet does. Yet the algorithm is the same for both languages and therefore extracts approximately the same amount of correct words. For MAP this means then that the equivalent counts of found true positives are divided by a twice as large number.

	base	dice	tan	MI	frMI	t-sc	z-sc	lg	lg2	ps1	ps2
only	2.85	14.33	14.33	10.13	11.01	2.16	13.91	13.16	13.16	14.00	14.58
base	5.78	11.62	11.61	7.79	9.41	5.39	8.56	11.65	11.71	11.55	13.60
over	5.14	10.73	10.72	7.44	5.54	4.92	7.19	6.03	6.00	7.29	8.28
dice	5.88	11.76	11.76	7.78	9.54	5.49	8.69	11.81	11.87	11.54	13.28
cos	5.25	11.67	11.62	8.06	9.22	5.97	8.78	8.88	9.44	10.00	10.26
cbin	5.90	11.70	11.70	7.75	9.54	5.51	8.51	11.55	11.60	11.14	12.76
L1	5.36	5.01	4.67	5.68	8.53	5.99	4.03	4.58	4.80	4.42	3.73
L2	4.98	5.15	5.14	5.31	6.70	5.65	4.06	6.02	6.38	5.82	5.94
JS	4.93	4.71	4.66	4.70	3.91	5.04	4.16	2.92	2.94	2.42	2.58

Table 3.12.: Precision in % for the 5 most significant or similar words for the German subcorpus measured on the Annotation Project (using the *total* relation) for all measure combinations.

Finally, the baseline (ranking according to plain co-occurrence frequency) is consistently outperformed by every measure except the t-score throughout all three experiments. Hence, 'Yes, we can do better than frequency!', to answer the question formulated by Krenn and Evert (2001) on a similar topic.

3.5.5. Performance of similarity measures

Comparing the similarity measures results in several surprising observations. First, it appears to be impossible to outperform the baseline. Second, Jensen-Shannon divergence is not the best measure, which seemingly contradicts the results of other researchers.

The **baseline** disregards the computed significance values and only counts matching non-zero elements in the two vectors to be compared. The other measures either additionally take the amount of non-zero elements into account or weight the non-zero elements according to their relative values. Obviously, all this additional information either degradates the results (*over*, for example) or does not seem to improve them.

A manual examination of several examples reveals that the reason is a combination of data sparsity and Zipfs Law (Zipf, 1949). Given an input word A and a set of words whose co-occurrence vectors have at least one matching element with the co-occurrence vector of A, the amount of matches is power-law distributed, as depicted in Figure 3.4 for the example word 'village'. The amount of non-zero values in the vectors is power-law distributed as well. That means that most words have one match, fewer two matches, etc. This also means that particularly the words with the highest contextual similarity to A have initially rapidly falling amounts of matches (with a long tail). The first might have 200 matches, the next only 167, then the next only 150 etc.

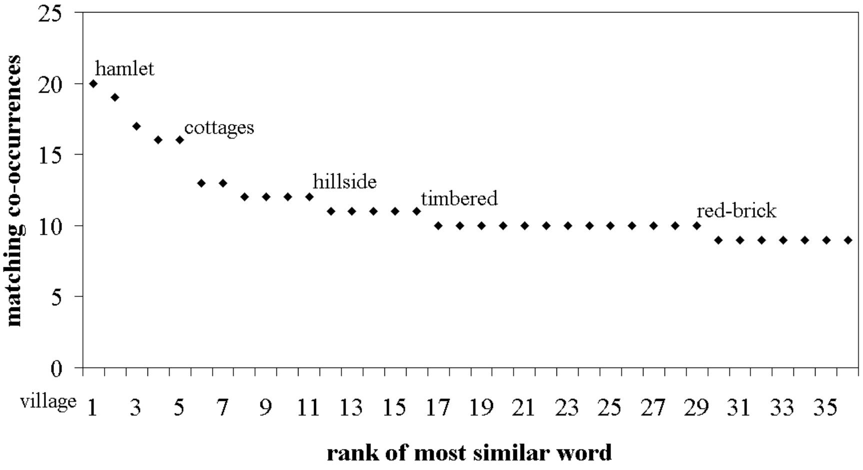


Figure 3.4.: Amounts of matching non-zero entries in co-occurrence rankings for most similar words of 'village'.

Following these considerations, the individual performance of each similarity measure can be easily explained. The **overlap** factor yields worse results, because it gives a higher weight to statistically non-representative words over representantive ones. If a word B has only two significant co-occurrences and another word A has 100, then chances are good that the two of B both match with two of the 100 of A. This vector then has a maximum overlap factor of 1.0. However, another word C with 100 co-occurrences and 80 of them being mutually non-zero in the vectors of A and C, produces an overlap similarity of only 0.8. Clearly, this is not preferable.

The binary version of **dice** resebles the overlap factor except that in the divisor the lengthes of both vectors are combined. This alleviates the problem described above and produces rankings very similar to the baseline. At the same times it is less dependent on the frequency sort order for words with the same amount of matches. This also explains its similar performance with the baseline.

It is interesting that the real-valued similarity measures result in consistently worse rankings. This appears to result from a certain information loss associated with weighting a few words against many others. The significance values produced by any co-occurrence measure in this evaluation are power-law distributed as well. This means that there are a few very large and many small values in the vectors. Whenever a measure such as cosine then attempts to take that into account, all small non-zero entries in the vector become nearly irrelevant in that their distinguishing power is being minimized.

For example, the comparison of two vectors with 60 matches is then reduced to comparing only the 5 to 10 matches with the largest values. Obviously, the gain from taking the weights into account is by far outweighted by the aggravation of the data sparseness this introduces. This reasoning is further supported by the fact that the cosine performs almost as good as the baseline, if instead of the values in the vectors only their presence is used (i.e. binary cosine), but a lot worse if the real values of the entries are used instead.

The poor performance of the **city block metric** (L1) and the **euclidian distance** (L2) indicates that these measures are inappropriate to the task of comparing words due to the reasons explained above. Because they directly depend on the dimensions used to be truly orthogonal, the poor performance additionally indicates the inappropriateness of the vector space model that is used to model the context vectors. It would be interesting to see, whether it is possible to increase their performance by modifying the vector space. Such modifications might be based on additional knowledge about word ambiguity or morphological relatedness (as obtainable from the algorithms in the following two chapters).

As mentioned previously, the **Jensen-Shannon** (JS) divergence is applicable only to frequency counts, because only they can be directly transformed into probabilities. Hence, in the result Tables from 3.6 to 3.12 only the results for JS applied on the baseline co-occurrence measure are meaningful.

Out of all measures applied to the baseline co-occurrence data, the JS is indeed among the best measures, although not consistently. However, using a proper significance measure (log-likelihood) and then computing similarity using the *dice* measure consistently produces approximately twice as good similarity rankings throughout all three experiments. As could be expected, applying JS to vectors containing significance values rather than frequency counts produces rather awkward results. The combination of the baseline co-occurrence measure with JS, reproduces the findings in related work where JS (or the related alpha-skew) was found to be one of the best-performing measure. At the same time, this shows that an entirely different approach easily outperforms this measure. At this point however, it can be expected that only a more appropriate word space model instead of the straight forward vector space, might yield further improvements for the simple co-occurrence and similarity measure based paradigm.

Finally, it remains interesting to see whether a similarty measure can be defined which consistently outperforms the baseline in this task - despite the data sparseness. One possibility might be a rank-based similarity measure. Another could be second order similarity, i.e. comparing words based not only on their context vectors, but also on those of the words contained in the initial context vector (similarly to the iterated co-occurrences (Biemann, Bordag, and Quasthoff, 2004)).

3.5.6. Correlation analysis

In addition to using the Scheffé test for comparing the various measure combinations it is possible to compute a pairwise Pearson's correlation coefficient. There are three types of possible sources of interactions to be tested: interactions between the co-occurrence measures (Table 3.13), the influence of co-occurrence measures on similarity (Table 3.15) and the interactions between similarity measures (Table 3.13). The Pearson coefficient offers a slightly different perspective on the interactions between the measures, because it is less affected by the actual resulting MAP values.

Using the MAP value (again on the *total* relation) for each word, a separate set of 35 966 data points (words both among the 100 000 most frequent words and contained in WordNet) was used as a representation of each measure combination. Similarly to the Scheffé test, Table 3.14 shows the correlation coefficients between all co-occurrence measures, Table 3.16 between the co-occurrence rankings when used to compute word similarity and Table 3.18 compares the similarity measures based on one co-occurrence ranking.

	1	2	3
t-sc	0.42		
base	0.52		
MI		1.39	
LMI		1.48	
lg			1.67
lg2			1.67
z-sc			1.75
ps1			1.75
ps2			1.77
dice			1.79
tan			1.79
sig	0.664	0.807	0.247

Table 3.13.: Scheffé test for MAP values of co-occurrence measures (based on the BNC and WordNet). $\alpha = 0.05$

A low correlation coefficient such as 0.53 between *base* and *dice* in Table 3.14 means that the co-occurrence rankings produced by these two measures differ strongly, but are somewhat similar. A high correlation coefficient such as 0.95 between *dice* and *z-sc* means that the rankings are nearly identical with respect to the used knowledge source. Aside from a few exceptions such as *dice* and *MI* with 0.9, a coefficient of 0.8 or higher correlates with the fact that the two measures do not differ significantly.

Obviously, several measures, such as *lg, lg2* and *ps* correlate strongly. For some measure pairs the correlation is by definition, such as between the two log-likelihood variants. However, it is interesting that *t-sc* and *base* are almost identical. Apparently, for most cases the expected value is much smaller than the observed value from which it is being subtracted. Hence, if $\frac{n_A \cdot n\_B}{n^2} << n_{AB}$ the equation for *t-sc* can be simplified to $sig(A,B)_{t-sc} = \frac{n_{AB}}{\sqrt{n_{AB}}}$, which would produce rankings identical

	base	dice	tan	MI	frMI	t-sc	z-sc	lg	lg2	ps1	ps2
base		0.53	0.53	0.52	0.77	0.95	0.59	0.72	0.72	0.68	0.70
dice	0.53		**1.00**	0.90	0.77	0.51	**0.95**	**0.80**	**0.80**	**0.83**	**0.80**
tan	0.53	**1.00**		0.90	0.77	0.51	**0.95**	**0.80**	**0.80**	**0.83**	**0.80**
MI	0.52	0.90	0.90		0.71	0.50	0.90	0.72	0.72	0.76	0.71
LMI	0.77	0.77	0.77	0.71		0.72	0.82	0.97	0.97	0.94	0.95
t-sc	0.95	0.51	0.51	0.50	0.72		0.55	0.68	0.68	0.63	0.66
z-sc	0.59	**0.95**	**0.95**	0.90	0.82	0.55		**0.84**	**0.84**	**0.87**	**0.85**
lg	0.72	**0.80**	**0.80**	0.72	0.97	0.68	**0.84**		**1.00**	**0.97**	**0.98**
lg2	0.72	**0.80**	**0.80**	0.72	0.97	0.68	**0.84**	**1.00**		**0.97**	**0.98**
ps1	0.68	**0.83**	**0.83**	0.76	0.94	0.63	**0.87**	**0.97**	**0.97**		**0.98**
ps2	0.70	**0.80**	**0.80**	0.71	0.95	0.66	**0.85**	**0.98**	**0.98**	**0.98**	

Table 3.14.: Pearson correlation coefficients between all pairs of co-ocurrence significance measures. The group containing the best-performing measures that do not differ significantly is highlighted.

to the baseline.

Also relevant is the strong correlation between *z-sc* and *t-sc*, although according to the Scheffé test their performance was found to differ significantly. This is again due to the observed value typically being much larger than the expected value. The correspondingly simplified form of the *z-sc* significance measure takes the form $sig(A,B)_{z-sc} = \frac{n \cdot n_{AB}}{\sqrt{n_A \cdot n_B}}$, which essentially does not differ from $sig(A,B)_{MI} = \frac{n \cdot n_{AB}}{n_A \cdot n_B}$ in that it produces identical rankings. Hence, the significant difference in their performance relates to such cases where the difference between observed and expected value is not large, which is more probable for rare words.

As expected from the discussion in Section 3.2.3, the *LMI* measure correlates strongly with the log-likelihood measures, yet due to the lower performance the MAP values differ significantly.

Table 3.16 indicates that when using co-occurrence measures to compute word similarity, the correlations between many measures become weaker. The large group of measures not differing significantly splits into several smaller groups, also reflected by generally lower correlation figures. It is interesting, that according to the Scheffé test, *ps2* differs significantly from the other log-likelihood measures, but the correlation coefficient between *ps2* and *lg* is nearly the same as between *ps1* and *lg*.

Table 3.18 shows that the three best-performing measures *base*, *dice* and *cbin* produce nearly identical rankings. The conclusion is that it is irrelevant which of the three to use, whereas the other measures perform significantly worse. Incidentally, these three measures do not make use of the underlying significance values. This supports the hypothesis (formulated above) that the significance values themselves are irrelevant, but rather their interpretation in terms of the ranking they define.

	1	2	3	4	5
t-sc	1.02				
base	1.11				
MI		1.62			
z-sc		1.68	1.68		
LMI		1.80	1.80		
tan			1.85		
dice			1.85		
lg				2.10	
lg2				2.10	
ps				2.10	
ps2					2.58
sig	0.996	0.222	0.244	1.0	1.0

Table 3.15.: Scheffé test for MAP of similarity based on various co-occurrence measures (based on the BNC and WordNet). $\alpha = 0.05$

	base	dice	tan	MI	frMI	t-sc	z-sc	lg	lg2	ps1	ps2
base		0.41	0.41	0.43	0.40	0.44	0.48	0.43	0.42	0.44	0.43
dice	0.41		0.99	0.53	0.58	0.38	0.70	0.63	0.64	0.72	0.70
tan	0.41	0.99		0.53	0.58	0.38	0.70	0.63	0.64	0.72	0.70
MI	0.43	0.53	0.53	0.35	0.29	0.56	0.38	0.39	0.46	0.43	
LMI	0.40	0.58	0.58	0.35		0.55	0.61	0.78	0.79	0.71	0.70
t-sx	0.44	0.38	0.38	0.29	0.55		0.43	0.52	0.51	0.48	0.48
z-sc	0.48	0.70	0.70	0.56	0.61	0.43		0.66	0.66	0.70	0.66
lg	0.43	0.63	0.63	0.38	0.78	0.52	0.66		**0.96**	**0.80**	0.79
lg2	0.42	0.64	0.64	0.39	0.79	0.51	0.66	**0.96**		**0.80**	0.79
ps	0.44	0.72	0.72	0.46	0.71	0.48	0.70	**0.80**	**0.80**		0.80
ps2	0.43	0.70	0.70	0.43	0.70	0.48	0.66	0.79	0.79	0.80	

Table 3.16.: Pearson correlation coefficient between all pairs of co-occurrence significance measures applied to compute word similarity using the baseline similarity measure. The group containing the measures performing second-best not differing significantly from each other is highlighted (the group containing the best measure consists only of the one measure *ps2_base*).

3.5.7. Preferences for linguistic relations

According to the SIML, significant differences between the extracted relation types should be observable only when comparing a co-occurrence significance measure with a similarity measure. Within the set of co-occurrence measures the preferences should remain mostly stable, equally as within the similarity measures. Additionally, the co-occurrence measures should rank syntagmatic relations higher, whereas

	1	2	3	4	5
L1	0.94				
L2		1.30			
only			1.77		
over			1.86		
cos				2.33	
cbin				2.50	2.50
dice					2.57
base					2.58
sig	1.0	1.0	0.581	0.050	0.865

Table 3.17.: Scheffé test for MAP of various similarity measures based on *ps2* co-occurrence measure (based on the BNC and WordNet). $\alpha = 0.05$

	only	base	over	dice	cbin	cos	L1	L2
only		0.39	0.36	0.39	0.39	0.33	0.20	0.19
base	0.39		0.82	**0.98**	**0.97**	0.68	0.31	0.36
over	0.36	0.82		0.84	0.86	0.65	0.49	0.42
dice	0.39	**0.98**	0.84		**0.99**	0.69	0.36	0.40
cbin	0.39	**0.97**	0.86	**0.99**		0.68	0.38	0.40
cos	0.33	0.68	0.65	0.69	0.68		0.46	0.56
L1	0.20	0.31	0.49	0.36	0.38	0.46		0.76
L2	0.19	0.36	0.42	0.40	0.40	0.56	0.76	

Table 3.18.: Pearson correlation coefficient between most similarity measures using the *ps2* co-occurrence measure. The group containing the second-best performing measures that do not differ significantly is highlighted (the group containing the best measure consists only of the one measure *ps2_base*).

paradigmatic relations should be ranked higher by similarity measures. However, of the three knowledge sources used, only one, the Annotation Project, also contains syntagmatic relations. This leads to a third set of experiments where the measure combinations are evaluated using the German subcorpus and the Annotation Project as a knowledge source.

Instead of simply counting each relevant word as correct, separate evaluations for each relation type can be given. As described in Section 3.4.7, all relations in the Annotation Project can be classified into a small set of relation types, such as syntagmatic and paradigmatic relations. Of the resulting relations, only the syntagmatic and paradigmatic relations are relevant. The paradigmatic relations are further separated into symmetrical and hierarchical paradigmatic relations, because otherwise certain interactions would remain undetected (see below).

Table 3.19 shows that for all co-occurrence measures the relation between the

.	syn	para	hier	syn/para	syn/hier	para/hier
base	2.02	1.63	0.96	1.24	2.10	1.69
dice	5.90	6.96	2.38	0.85	2.48	2.92
tan	5.90	6.97	2.38	0.85	2.48	2.92
MI	4.75	4.91	1.98	0.97	2.40	2.48
frMI	5.31	5.02	2.39	1.06	2.23	2.11
t-sc	1.73	1.34	0.82	1.29	2.11	1.63
z-sc	6.20	6.36	2.46	0.98	2.52	2.58
lg	6.08	5.96	2.71	1.02	2.24	2.20
lg2	6.08	5.96	2.71	1.02	2.24	2.20
ps	6.44	6.41	2.79	1.01	2.31	2.30
ps2	6.57	6.54	2.92	1.01	2.25	2.24

Table 3.19.: MAP in % for the German subcorpus measured on the Annotation Project for syntagmatic, paradigmatic and hierarchical paradigmatic relations for all co-occurrence significance measures.

extraction of syntagmatic and paradigmatic relations measured by dividing their MAP values varies within a small interval - from 0.85 to 1.29. For the better measures (*dice*, *tan*, *lg*, *ps2*), this interval is even smaller (from 0.85 to 1.02). The same applies to the relation between syntagmatic and hierarchical paradigmatic relations. However, the same cannot be said about the difference between the symmetrical and hierarchical paradigmatic relations. Both *base* and *t-sc* deviate strongly from the other measures, which extract symmetrical paradigmatic relations roughly two to three times better than hierarchical relations. This deviation could be due to their generally poor performance, which increases noise, rather than due to an underlying effect.

Contrastingly, Table 3.20 again shows the relations between ranking syntagmatic, symmetrical and hierarchical paradigmatic relations, except for the similarity measures (based on various co-occurrence measures). Due to the large number of combinations only a selection is shown. The first half of the table shows the measures that produced the most extreme ratios. The second half of the table shows the values for all similarity measures computed using the baseline co-occurrence measure, as well as the second Poisson approximation.

The most important observation in Table 3.20 is that the ratios between syntagmatic and symmetrical paradigmatic relations differ strongly from the ratios observed in Table 3.19. Simultaneously the syn/para ratios vary only little within the interval of 0.15 to 0.43. These findings support the predictions made by the SIML: significant co-occurrences express a syntagmatic relationship, whereas similarity comparisons express paradigmatic relations. Some issues remain, including that co-occurrence measure such as *dice* rank paradigmatic relations higher than syntagmatic (or at least roughly as high as is the case of the log-likelihood vari-

.	syn	para	hier	syn/para	syn/hier	para/hier
frMI-cos	0.91	6.20	4.00	**0.15**	**0.23**	1.55
dice-L1	1.16	2.71	1.82	**0.43**	0.64	1.49
MI-L2	0.86	2.77	0.82	0.31	**1.05**	3.38
frMI-base	1.36	5.82	4.18	0.23	0.33	**1.39**
MI-dice	1.22	4.57	1.29	0.27	0.94	**3.53**
base-base	0.85	3.95	2.21	0.22	0.39	1.79
base-over	0.80	3.55	1.98	0.22	0.40	1.80
base-dice	0.87	4.09	2.22	0.21	0.39	1.84
base-jacc	0.87	4.09	2.22	0.21	0.39	1.84
base-cos	0.66	3.64	1.83	0.18	0.36	1.99
base-cbin	0.87	4.11	2.22	0.21	0.39	1.85
base-L1	0.90	3.77	1.17	0.24	0.77	3.23
base-L2	0.88	3.53	1.12	0.25	0.79	3.16
base-JS	0.86	3.57	1.18	0.24	0.73	3.03
ps2-base	2.08	8.52	4.90	0.24	0.42	1.74
ps2-over	1.35	5.08	2.73	0.26	0.49	1.86
ps2-dice	1.94	8.42	4.55	0.23	0.43	1.85
ps2-jacc	1.94	8.42	4.55	0.23	0.43	1.85
ps2-cos	1.36	6.91	3.47	0.20	0.39	1.99
ps2-cbin	1.86	8.05	4.30	0.23	0.43	1.87
ps2-L1	0.56	2.31	0.90	0.24	0.62	2.57
ps2-L2	0.69	3.52	1.31	0.20	0.53	2.69
ps2-JS	0.56	1.88	0.64	0.30	0.87	2.94

Table 3.20.: A selection of measure combinations with their relative preference for relation types measured on syntagmatic, paradigmatic and hierarchical paradigmatic relations. MAP in % for the German subcorpus measured on the Annotation Project.

ants), but these are further discussed in Chapter 6. Currently, it suffices that the observations do not falsify the predictions.

Despite this, the additionally provided ratios between syntagmatic relations and hierarchical paradigmatic relations seem to falsify the predictions, because of a rather large interval, ranging from 0.23 to 1.05. This appears to indicate that combining *MI* with *L2* ranks syntagmatic relations significantly higher than hierarchical paradigmatic relations. This difference also is very strong, compared to other measure combinations such as *LMI-cos*, which contrarily rank hierarchical paradigmatic relations higher. However, the reason for this is a variation in preferences between symmetrical and hierarchical paradigmatic relations, as the ratios between these demonstrate.

In fact, it is possible to observe two sets of similarity measures. One set consists of *L1*, *L2* and *JS*, consistently ranking symmetrical paradigmatic relations over three times as high as all other measures. The other set consists of the remaining measures, which show only little variation in this respect, and consistently rank symmetrical paradigmatic relations less than twice as high as hierarchical ones. Again, this coincides with the general poor performance of the first set of measures. Nevertheless, the measures *L1*, *L2* and *JS* appear to share a property causing both poor performance and a stronger preference for symmetrical paradigmatic relations. While this property remains unknown, a possibility is that the word pairs standing in symmetrical paradigmatic relations (mostly cohyponyms) are easier to observe, because they have more similar frequencies and more symmetrical contexts.

To summarize, the experiments in this section show that irrespective of the co-occurrence significance measure, syntagmatic relations are clearly respresented by co-occurrence observations. Similarly, paradigmatic relations are represented by comparisons of global contexts, irrespective of how these contexts are acquired and how they are compared. Only the overall quality of the results is affected by the choice of a particular measure combination.

3.5.8. Correlation of performance and word frequency

Experiment 4 is devoted to gauge how the various measures cope with data sparseness. One claim frequently made when introducing a new statistical significance test is that it allows for more precise significance estimates, even for few occurrences (Dunning, 1993). The evaluation method used in this chapter allows for empirical data to be provided that shows whether such claims are true. In fact, the experiments in Section 3.5.4 already show that the group of log-likelihood related co-occurrence measures, when used for similarity computations, perform consistently better than any other measures. However, these results do not show, whether or not there is a correlation between the frequency of a particular word and the ability of a given measure combination to compute its most similar words or determine most significant co-occurrences.

To show the existence of such a correlation, the following experiment was conducted. First, the set of 100 000 most frequent words was split into 100 slices, each containing 1 000, excluding those not annotated in the Annotation Project. Then the average performance measured on the *total* relation using precision was obtained for each slice. Because the slices containing less frequent words have a large variance in results, the performance measured for each slice was averaged with the values of the two previous and two following slices. These smoothed values are shown in Figure 3.5, for a selection of 6 measure combinations. The selection includes only the co-occurrence measures *base*, *dice* and *ps2*. This is because *dice* and *ps2* perform equally well when compared directly and the other, unrelated measures such as *MI*, perform considerably worse. For each of these co-occurrence measures, their direct performance, as well as their performance when used for similarity computations measured as precision for 5 most relevant words are shown.

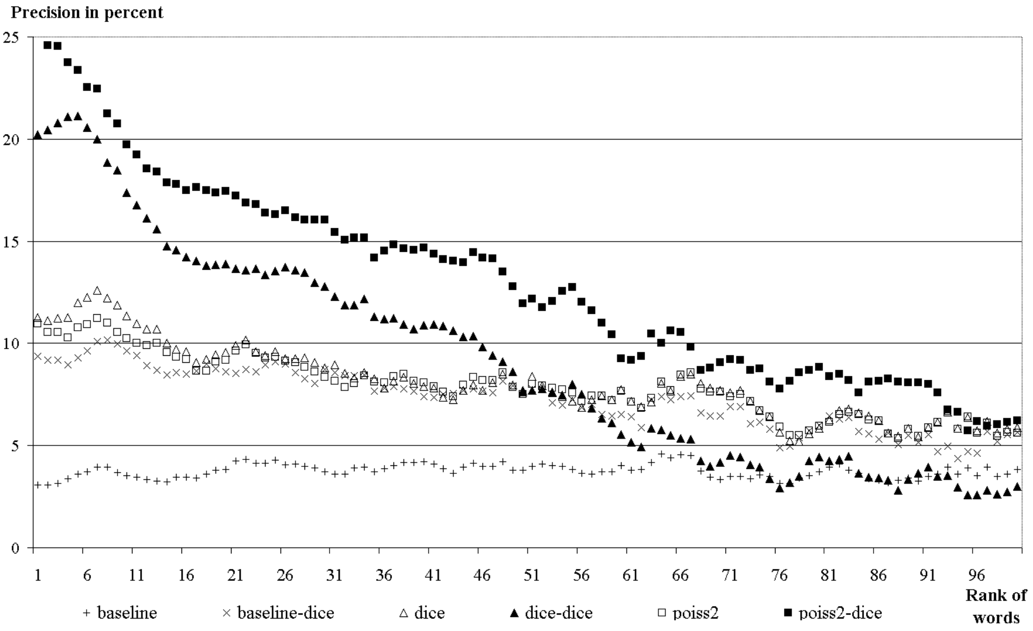


Figure 3.5.: Influence of word frequency on performance of co-occurrence measure and similarity measures. Measured for 100 000 most frequent words from the German subcorpus using the Annotation Project as gold standard and precision for 5 most relevant words. Each data point is performance for 1 000 words, averaged over the previous and following two data points for smoothing.

As expected, Figure 3.5 shows an almost completely overlapping curve for the co-occurrence measures *dice* and *ps2*, and a much lower curve for the *base*. It is interesting that the baseline performance does not correlate with word frequency, whereas both *dice* and *ps2* perform significantly better on more frequent words. However, the decisive difference between *dice* and *ps2* is that when used to compute similarity, the performance degradation for lesser frequent words differs strongly. In fact, computing similar words using *dice* co-occurrences produces worse results than *base*, even though both measures perform almost equally well for frequent words. The *ps2* significance measure though, does not show any such weaknesses and it's performance degradates slowly according to the falling frequency of the words up to the point where observations are too scarce to enable the observation of significant differences.

Due to the correlation analysis in Section 3.5.6 it is to be expected that the other log-likelihood measures (*lg*, *lg2*, *ps1*) fare equally well, whereas *tan* should have the same weakness.

3.6. Conclusions

The results reported in this chapter are both of practical and theoretical relevance. The practical relevance is for any type of application that depends on simulated knowledge of word associations or word similarity. The evaluations clearly show which measures are suited best for such a simulation and the discussions show the underlying reasons for why they are suited best. The evaluation method itself has been thoroughly tested and it's weaknesses and strengthes were highlighted. The proposed evaluation method enables easy comparison of new measures and algorithms. It can, for example, be used to compare other measures using different corpora, thus alleviating the problem of different authors using different corpora to compare new measures with known ones. It suffices to evaluate the new measure alongside one or more of the known algorithms. The absolute numbers will vary, but not the relations between the performances of the algorithms.

The theoretical implications are manifold. For the SIML the most important question is whether indeed, as predicted, co-occurrence observations simulate syntagmatic relatedness of words, whereas similarity computations based on co-occurrences simulate paradigmatic relatedness of words. It was possible to observe a clear and statistically significant difference with respect to the extracted syntagmatic or paradigmatic relations between co-occurrence and similarity measures. However, both methods also extract a significant amount of 'wrong' relations, which is further discussed in Chapter 6.

A related topic is the proper approach to the extraction of collocations, idiomatic expressions or multi word units. An approach that can be called the **global** approach (contrary to the **local** approach in this work) is probably best summarized by Evert (2004). It uses a co-occurrence significance measure to rank all possible word pairs extracted from a corpus. This assumes that the significance values of two different pairs of words, for example $sig(A, B) = 50$ and $sig(C, D) = 10$ are comparable in that $50 > 10$ and hence $sig(A, B)$ is more significant than $sig(C, D)$. However, given that the values produced by most co-occurrence measures scale up corresponding to the frequency of the two words, this assumption may be faulty. This would explain why the same measures perform differently in Everts work. Several possibilities to use the local approach to extract such collocations are discussed further in Chapter 6.

Another theoretical question regards the ability of any particular significance measure to determine the significance of an observation even for rare events. It could be shown that mathematically well-founded measures such as the log-likelihood measures or Poisson approximations show a stable performance even for less-frequency words, whereas an initially well-performing measure (such as *dice*) degradated extremely for less frequent words.

Finally, this chapter provides a solid foundation for the remaining chapters of this work. It provides a sound and practical method to simulate knowledge about each word by only using information available from a raw corpus without any manual pre-processing. Co-occurrence rankings are used to find word ambiguity in the following chapter. Word similarity is used to obtain morphological information in Chapter 5. Chapter 6 finally discusses further methods to distinguish between syntagmatic and paradigmatic relations. As such, all these methods can be considered unsupervised and knowledge-free, because at any point only the most general assumptions about language are made, without ever adding corrections to the results or using knowledge about the particular language.

4. Lexical Ambiguity

In the previous chapter, the extraction of basic syntagmatic and paradigmatic relations was introduced. The implicit assumption was that each word has a unique meaning, resulting in a global context of words which are largely related to each other. However, for many instances the set of co-occurring words that represent the global context contain clearly divisible word groups. In such a group the words are related to each other, but not to words of the other groups. For example the word *feather* co-occurs not only with the words *flock, birds, eagle* but with *cap, hat, headdress* as well.

These observable partitions can be explained through polysemy or ambiguity. Ambiguity means that a word can have several distinct meanings. Apart from easily observable lexical senses there are several other types of polysemy which significantly impact the correct parsing of a sentence. The following is a sample list comparing some instances of polysemy based upon potential meanings or usages of a given word:

1. **Lexical senses:** A word can have several entirely unrelated meanings stemming from either metaphoric or idiomatic usage. For example, one meaning of the word *space* refers to the three-dimensional expanse within which everything is located, whereas a second possible meaning (not listed in WordNet (Miller, 1990; Fellbaum, 1998)) is the amount of data blocks on a hard drive.

2. **Syntactic senses:** Some languages allow a specific word to be used as different parts of speech. For example *walk* is used as a noun in *to take a walk*, but as a verb in *to walk somewhere.* Syntactical differences in meaning need not be automatic indicators for lexical differences in meaning.

3. **Idiomatic senses:** Several words can be used in one fixed idiomatic expression where the parts of the expression can be completely unrelated to each other and hence also with to input word. As seen in the expression *to beat about the bush* the word *beat* co-occurs with *bush* although it is highly improbable that these two words words would co-occur ouside of this expression.

4. **Metaphoric senses:** When a word is used metaphorically in a certain context, it provides for atypical co-occurrences. For example it is possible to say *You are my sun!* to emphasize or magnify the meaning of an utterance. Some metaphors are in common usage and often resemble idiomatic expressions, but new metaphors can be invented at any given time. In the words of

> Kilgarriff (1997): “Metaphor is, among other things, a process where words spawn additional meanings”.

Alternatively other, more detailed classifications of polysemy are also possible (Levin and Hovav, 1991; Levin, 1993; Levin, Song, and Atkins., 1997). For this chapter the given classification is sufficient to explain the phenomenon as well as make good predictions as to which types of polysemy are easy to find and which require a more elaborated method.

Does polysemy exist?

In attempting to devise an algorithm that automatically finds different senses of a word from a corpus, it is important that many of the senses found in a given gold standard need not be present in the corpus. However, the reverse is also possible, where distinctions between senses are clearly manifest based on corpus evidence but not so clear in the gold standard (Kilgarriff, 1997). A good example is the clear lexical distinction between *space (and astronauts)* and *office space*. In both cases the WordNet meaning #1 of the noun space is to be considered applicable. However, when considering the example *joint* (i.e., either of a living being or of a robot), it is difficult to determine if the word has one sense or two, regardless of plentiful corpus evidence in favor of a clear distinction.

All such considerations have lead Kilgarriff (1997) to formulate the hypothesis, that senses do exist, but only relative to the task the corpus from which they are drawn is to be applied to. This hypothesis may be too strong - it seems more appropriate to state that senses exist solely within a given corpus and the task is to find methods that automatically distinguish the four (or more) types of polysemy. Regarding unclear cases such as the above-mentioned *joint*: such decisions might be modified to be indeed purpose-oriented, since for some applications it could be a potentially useful distinction, whereas for others it could be harmful.

Consequently, the question of whether ambiguity can be automatically detected must be anwered positively. Much more intricate is the question of how to extract all types of polysemy. A great variety of syntagmatic and paradigmatic relations exist, and each can play a role in adding another meaning to a word. These again can be computed (or rather simulated) by various representations of local and global context. For example, once neighborhood co-occurrences are computed, it is possible to apply an unsupervised part-of-speech clustering algorithm and tagger (Schütze, 1995; Biemann, 2006b) to detect classes of words. Obviously, all words appearing in more than one resulting class can be assumed as syntactically ambiguous. Word senses found this way would differ from lexical senses detected by clustering sentence co-occurrences, as described further in this chapter. This yields a first distinction between lexical and syntactical senses of words. In order to obtain idiomatic senses, it would be possible to employ any of the existing algorithms to find linguistic collocations (Evert and Krenn, 2001). Once a word is found to be part of a linguistic collocation such as *beat about the* **bush** it can be assumed to have such a sense, in addition to all other previously detected senses of that word.

Thus, the underlying hypothesis of this chapter is that different word senses exist in any corpus and when using proper methods any type of ambiguity can be found - if the frequency of that sense is sufficient. Furthermore, it appears that the quality of the specific algorithm determines which frequency level is sufficient. Ideally even a low frequency should suffice, but currently this remains unfeasible.

In terms of the SIML in Chapter 2, polysemy is given when two or more different global contexts share the same word form. Stated differently, two word forms, w_i and w_j appear identical, but have two distinct global contexts $K_G(w_i)$ and $K_G(w_j)$ instead of one. The oversimplification[1] employed to define and compute the global context of a word lead to only one apparent global context. Hence, the algorithms introduced below essentially just refines the implementation of the simulation of the global context in order to avoid merging two global contexts only because their word forms happen to appear similar.

4.1. Introduction

According to the formulated hypothesis, this chapter introduces one new and reviews several existing solutions to automatic and unsupervised word sense induction (WSI). The solution presented here is primarily a lexical word sense induction, because it is based on statistical sentence co-occurrences as context representations. It can be seen as an instantiation of the 'one sense per collocation' observation (Gale, Church, and Yarowsky, 1992). This approach differs from existing approaches to WSI in that it enhances the effect of the one sense per collocation observation by using triplets of words instead of pairs. The combination with a two-step clustering process using sentence co-occurrences as features allows for accurate results.

The method introduced here is specialized on sentence co-occurrences, thus it probably cannot be generalized for syntactic features and syntactic WSI. However, the likewise automatic and unsupervised evaluation method inspired by Schütze's (1998) concept of evaluating word sense disambiguation algorithms (introduced towards the end of this chapter) are easily employable in other scenarios as well, such as syntactic word sense induction. This evaluation method also offers the advantage of reproducibility and independency of a given biased gold standard. It also enables automatic parameter optimization of the WSI algorithm.

[1] The simplification here was the implicit assumption that each atom has exactly one unique meaning.

4.1.1. Related algorithms

The aim of WSI[2] is to find senses of a given target word (Yarowski, 1995) automatically, and ideally in an unsupervised manner. WSI is akin to word sense disambiguation (WSD) both in methods employed and problems encountered, such as the aforementioned vagueness of sense distinctions (Kilgarriff, 1997). The input of a WSI algorithm is a target word to be disambiguated, e.g. *space*, and the output is a number of word sets representing the various senses, e.g. *(3-dimensional, expanse, locate)* and *(office, building, square)*. Such results could, at minimum, be used as empirically grounded suggestions for lexicographers or as input for WSD algorithms. Other possible uses include automatic thesaurus or ontology construction, machine translation or information retrieval. But the usefulness of WSI in real-world applications has yet to be tested and proven.

To date, a substantial number of different approaches to WSI have been proposed. They are all based on co-occurrence statistics, albeit using different context representations including co-occurring words within phrases (Pantel and Lin, 2002; Dorow and Widdows, 2003; Velldal, 2005), bigrams (Schütze, 1998; Neill, 2002; Udani et al., 2005), or small windows around a word (Gauch and Futrelle, 1994), sentences (Bordag, 2003; Rapp, 2004) or windows of up to 20 words (Ferret, 2004). Moreover they all employ clustering methods to partition the co-occurring words into sets describing concepts or senses. Some algorithms aim for a global clustering of words into concepts (Yarowski, 1995; Pantel and Lin, 2002; Velldal, 2005). The majority of algorithms is based upon local clustering: Words co-occurring with the target word are grouped into the various senses the target word has. It is not immediately clear which approach to favor, however aiming at global senses has the inherent property of producing a uniform granularity of distinctions between potentially undesirable senses (Rapp, 2004).

Graph-based algorithms differ from the majority of algorithms in several aspects. Words can be taken as nodes and co-occurrence of two words defines an edge between the respective nodes. Activation spreading on the resulting graph can be employed (Barth, 2004) to obtain the most distinctly activated areas in the vicinity of the target word. It is also possible to use graph-based clustering techniques to obtain sense representations based on sub-graph density measures (Dorow and Widdows, 2003; Bordag, 2003; Biemann, 2006a). However, it remains unclear whether this kind of approach differs qualitatively from the standard clustering approaches. Generally though, the notion of sub-graph density seems more intuitive in comparison to more abstract clustering.

As mentioned previously, there are different types of polysemy, the primary distinction probably being between syntactic classes of the word (e.g. *to plant* vs. *a plant*) and conceptually different senses (e.g. *power plant* vs. *green plant*). As

[2] Also known as word sense discovery (Dorow and Widdows, 2003) or word sense discrimination (Purandare, 2004; Velldal, 2005)

known from work on unsupervised part-of-speech tagging (Rohwer and Freitag, 2004; Rapp, 2005b; Biemann, 2006b), the size of the co-occurrence window for the computation of word similarity plays a decisive role. Utilizing most significant direct neighbors as context representations to compare words results in predominantly syntactical similarity. Alternatively, using sentence co-occurrences results in mostly semantical similarity (Curran, 2003). Although varying context representations, similarity measures and clustering methods have been compared with each other (Purandare, 2004), there is no evidence to date as to whether various window sizes or other parameters influence the *type* of ambiguity found, see also (Manning and Schütze, 1999, p. 259).

Pantel & Lin (2002) used an evaluation method based on comparing obtained word senses with senses provided in WordNet. This method has been successfully employed by other authors as well (Purandare, 2004; Ferret, 2004), because it is straightforward and produces intuitive numbers which help to directly estimate whether the output of a WSI algorithm is meaningful. However, any gold standard such as WordNet is biased and lacks domain-specific sense definitions while providing an abundance of sense definitions that occur too rarely in most corpora. For example the sense #2 of *MALE* (*[n] the capital of Maldives*) from WordNet is represented by a single sentence only in the British National Corpus (BNC).

Furthermore, comparing results of an algorithm to WordNet automatically implicates another algorithm which matches the found senses with WordNet senses. This closely resembles the task of WSD and thus can be assumed to be similarly error prone. These reasons have led some researchers to perform a manual evaluation of their algorithms (Neill, 2002; Rapp, 2004; Udani et al., 2005). Manual evaluation, however, has its own disadvantages, most notably the poor reproducibility of results. In this chapter a pseudoword-based evaluation method similar to Schütze's (1998) method is employed. It is automatic, easily reproducible and highly adaptive to domain specificity of a given corpus.

4.2. Triplet-based algorithm

One sense per collocation: The algorithm described in this section (and illustrated in Figure 4.1) is the most recent version of the one previously described in (Bordag, 2003) and revised in (Bordag, 2006b). It is based on the *one sense per collocation* observation (Gale, Church, and Yarowsky, 1992). Essentially this means that when a pair of words frequently co-occurs in a corpus (hence a collocation), the concept referenced by that pair is unambiguous, e.g. *growing plant* vs. *power plant.* However, as pointed out by Yarowsky (1995), this observation does not hold uniformly for all possible co-occurrences of two words. It is stronger for adjacent co-occurrences and word pairs in a predicate-argument relationship than it is for arbitrary associations at equal distance, e.g. *a plant* is much less clear-cut. To alleviate this problem, the first step of the present algorithm is to build triplets of words (the target word and two of it's co-occurrences) instead of pairs (the target word and one co-occurrence). This means that *a plant* is further restricted by another word. Even a stop word such as *on* rules several potential interpretations of *a plant* out, or makes them far less probable.

Using sentence co-occurrences and neighbor co-occurrences: The newly introduced algorithm is applied to two types of co-occurrence data. To illustrate the influence of window size, both the most frequent sentence-wide co-occurrences and direct neighbor co-occurrences were computed for each word. The significance values were obtained using the log-likelihood measure as described in Section 3.2.4. For each word, only the $t_c = 200$ most significant co-occurrences are kept. This threshold and all others to follow were chosen after experimenting with the algorithm. However, as will be shown in Section 4.3, the ideal set-up of these numbers can be obtained automatically. The presented evaluation method enables automatic detection of the optimal parameter configuration.

Creating triplets: The core assumption of the triplet-based algorithm is that any three (or more) words uniquely identify a topic, concept or sense. In reality this is not always the case (for example *gold, silver, bronze* may mean metals or awards), but the probability is nearly maximized for three words as opposed to two words. Using the previously acquired most significant co-occurrences (of both types), the (ranked) lists of co-occurrences for all three words of a triplet are intersected to retain words contained in all three lists.

If the three words cover a topic, e.g. *space, NASA, Mars*, then the intersection of their co-occurrences is not empty, e.g. *launch, probe, cosmonaut, ...*. If the three words do not identify a meaningful topic, e.g. *space, NASA, cupboard*, then the intersection likely contains few to none words. Intersections of co-occurrences of function word triplets quite possibly contain many co-occurrences even when not identifying a unique topic. This is because function words are so unspecific, that they essentially identify any and no topic simultaneously. These 'stop words' are thus removed from the co-occurrences from which triplets are drawn as well as from the co-occurrences which are used as features. This introduces another parameter $t_f = 1\,000$ meaning that the most frequent 1 000 words are considered stop words.

Incremental clustering of triplets: At this point it is straightforward to create all possible triplets of the co-occurrences of the target word w and to compute the intersection of their co-occurrence lists. Using these intersections as features of the triplets, it is possible to group triplets of words that share features by means of any standard clustering algorithm. However, in order to 'tie' the referenced meanings of the triplets to the target word w, the resulting set of triplets can be

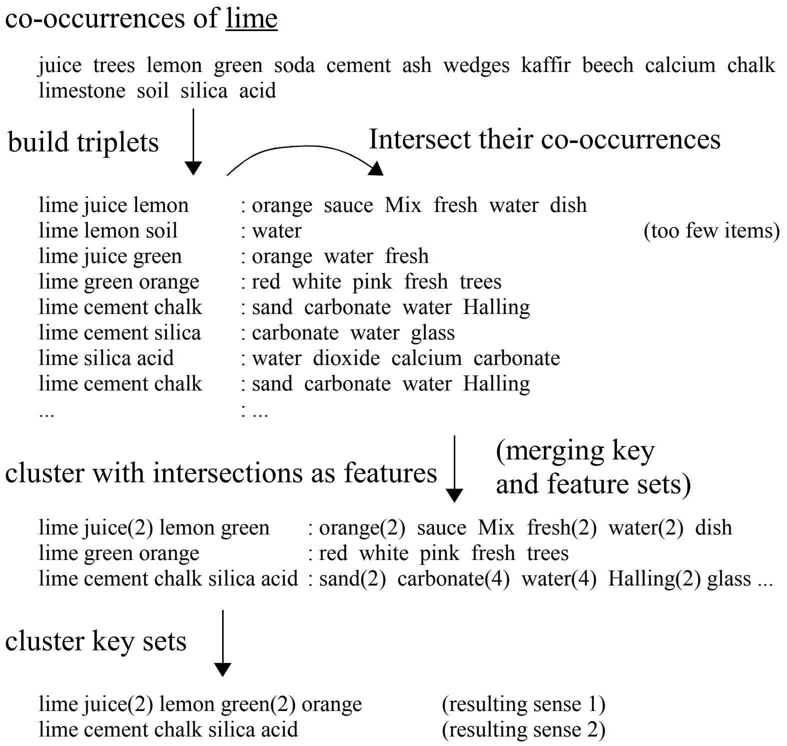

co-occurrences of lime
juice trees lemon green soda cement ash wedges kaffir beech calcium chalk limestone soil silica acid
build triplets
Intersect their co-occurrences
lime juice lemon : orange sauce Mix fresh water dish
lime lemon soil : water (too few items)
lime juice green : orange water fresh
lime green orange : red white pink fresh trees
lime cement chalk : sand carbonate water Halling
lime cement silica : carbonate water glass
lime silica acid : water dioxide calcium carbonate
lime cement chalk : sand carbonate water Halling
... : ...
cluster with intersections as features
(merging key and feature sets)
lime juice(2) lemon green : orange(2) sauce Mix fresh(2) water(2) dish
lime green orange : red white pink fresh trees
lime cement chalk silica acid : sand(2) carbonate(4) water(4) Halling(2) glass ...
cluster key sets
lime juice(2) lemon green(2) orange (resulting sense 1)
lime cement chalk silica acid (resulting sense 2)

Figure 4.1.: Illustration of the WSI algorithm for *lime*.

restricted only to those also containing the target word. This has the useful side-effect of reducing the number of triplets to cluster. To further reduce the remaining number of $\binom{200}{2} = 19\,900$ items to be clustered, an iterative incremental windowing mechanism is used. Instead of clustering all triplets in one step, 30 co-occurrences beginning from the most significant ones (of the input word) are taken in each step to construct $\binom{30}{2} = 435$ triplets and their intersections. The resulting elements (triplets and intersections of their respective co-occurrences as features) are then clustered with the clusters remaining from the previous step.

Merging triplets and their intersections: In each step of the clustering algorithm, the words from the triplets and their features are merged if the overlap factor similarity (Curran, 2003) was high enough (over 80% overlapping words out of 200). Thus, if the elements *(space, NASA, Mars) : (orbital, satellite, astronauts,...)* and *(space, launch, Mars) : (orbit, satellite, astronaut, ...)* were found to be similar, they are merged to *(space=2, NASA=1, Mars=1, launch=1) : (orbital=1, satellite=2, astronauts=1, orbit=1, astronaut=1, ...)*. Since the measure utilizes only features for comparisons, the result can contain two or more clusters with almost identical key sets (which result from merging triplets). A post-clustering step is therefore applied to compare clusters by the formerly triplet words and merge spurious sense distinctions. Thus having established the final clusters, the remaining unclustered words can be classified to the resulting clusters. Classification is performed by comparing the co-occurrences of each remaining word to the agglomerated feature words of each sense. If the overlap similarity of the most similar sense is below 0.8 the given word is not classified. Figure 4.1 illustrates the first iteration of the algorithm using data from the BNC for the ambiguous word lime. The entire cluster algorithm can be summarized as follows:

- Target word is w
- For each step take the next 30 co-occurrences of w
  - Construct all possible pairs of the 30 co-occurrences and add w to each to form triplets
  - Compute intersections of co-occurrences of each triplet
  - Intersections containing less then t_1 words are set to be empty
  - Cluster the triplets using their co-occurrence intersections as features in conjunction with clusters remaining from previous step
    - Whenever two clusters are found belonging together, the words from the triplets (key sets) and the features are merged
    - When merging two sets, for each word contained in both sets a counter is increased (or rather, their counters are summed in later steps)
- Cluster results of the loop by using the merged words of the triplets as features
- Classify unused words to the resulting clusters if possible
- Remove all clusters with less then t_2

Despite using triplets and only statistically significant co-occurrences, there is noise. Noise is, when the intersection of co-occurrences of a triplet of clearly unrelated words contains a few high-frequency words, such as *lime, lemon, soil* in Figure 4.1). Therefore a threshold t_1 defining a minimum acceptable intersection size of 4 was set. Another significant parameter t_2 is that after the last clustering step all clusters containing less than 8 words are removed. Recording how many times a given word has 'hit' a certain cluster (in each merging step) enables the addition of a post-processing step. In this optional step a word is removed from a cluster if it has 'hit' another cluster more often.

There are several issues and open questions arising from this entire approach. These include: why to use a particular similarity measure, a particular clustering method or why to merge the vectors instead of creating proper centroids. It is possible that another combination of such decisions will produce better results. However, the overall observation is that the results are fairly stable with respect to such decisions whereas parameters such as frequency of the target word, size of the corpus, balance of the various senses, etc., have a much greater impact. In fact, an entirely different algorithm (Biemann, 2006a) achieves very similar results to the ones reported below.

4.3. Evaluation

Schütze (1998) introduced a pseudoword-based evaluation method for WSD algorithms. The idea is to choose two arbitrary words, for example *banana* and *door*, and replace all occurrences of either word by the new pseudoword *bananadoor*. Then WSD is applied to each sentence and the amount of correctly disambiguated sentences is counted. A disambiguation is correct, if the sentence *I ate the banana* is assigned to sense #1 (banana) instead of #2 (door). In other words, all sentences where one of the two words occurs are viewed as one set, and the WSD algorithm should then distinguish them correctly. This, is actually quite similar to the WSI task, which is supposed to distinguish sets of words co-occurring with the target word. Thus it is again possible to take two words, view their co-occurrences as one set and let the WSI algorithm sort them apart. For example, the word *banana* might co-occur with *apple, fruit, coconut, ...* and the word *door* may co-occur with *open, front, locked, ...* The WSI algorithm would therefore have to disambiguate the pseudoword *bananadoor* with the co-occurrences *apple, open, fruit, front, locked, ...*

The evaluation is based on word forms occurring in the corpus, rather than using a lemmatizer. One reason is because to date, an unsupervisedly trained (or knowledge-free) lemmatizer does not exist. Another major reason is that reducing the amount of types through lemmatization increases ambiguity. For example, *talked* is a verb, whereas the lemmatized *talk* can either be a noun or a verb. Therefore the pseudo-word based evaluation will certainly be adversely affected by lemmatization. However, this effect has yet to be quantified. Due to the lack of corpus pre-processing, this evaluation cannot be compared with others, where lemmatization, and restrictions to certain syntactic word classes are often used (Pantel and Lin, 2002).

In short, the method merges the co-occurrences of two words into one set. Then, the WSI algorithm is applied to that set of co-occurrences and the evaluation measures the result by comparing it to the original co-occurrence sets. In order to find out whether a given sense has been correctly identified by the WSI algorithm, its **retrieval precision** (**rP**) - the similarity of the found sense with the original sense using the overlap measure - can be computed. In the present evaluations, the threshold of 0.6 was chosen, meaning that at least 60% of the found sense words must overlap with the original sense in order to be considered a correctly found sense. The average values of similarity are significantly higher, ranging between 85% and 95%.

It is further informative to measure **retrieval recall** (*rR*), i.e. the number of words which have been retrieved into the correct sense. If, for instance, two words are merged into a pseudoword and the meaning of each original word is represented by 200 co-occurring words, then it could happen that one of the senses has been correctly found by the WSI algorithm containing 110 words with an overlap similarity of 0.91. It follows that only 100 words representing the original sense were retrieved, resulting in 50% retrieval recall.

The retrieval recall also has an upper bound for two reasons. The average overlap ratio of the co-occurrences of the word pairs used for the evaluation was 3.6%. Another factor lowering the upper bound by an unknown amount is the ambiguity of some words. If the algorithm correctly finds various senses for one of the original words, then only one found sense will be chosen to represent the original 'meaning'. All co-occurrences assigned to the other sense are lost to it.

Using terminology from Information Retrieval is suitable because this task can be reformulated as follows: Given a set of 400 words and one of several word senses, attempt to retrieve all words belonging to that sense (retrieval recall) without retrieving any wrong ones (retrieval precision). A sense is defined as correctly found by the WSI algorithm if its retrieval precision is above 60% and retrieval recall above 25%. The latter number implies that a minimum of 50 words must be retrieved correctly since the initial co-occurrence set contained 200 words. This also assumes that 50 words suffice to characterize a sense if the WSI algorithm is not solely used for self-evaluation. The minimum retrieval precision is set to a value above 50% to prevent an overly strong baseline, see below.

Using these prerequisites, it is possible to define precision and recall (based on retrieval precision and retrieval recall), which will be used to measure the quality of the WSI algorithm.

Precision (*P*) is defined as the number of times the original co-occurrence sets are properly restored divided by the number of different sets found. Thus, precision has an unknown upper bound below 100%, because any two words could be ambiguous themselves. Thus, if the algorithm finds three meanings of the pseudoword, it could be that one of the two words was ambiguous, and hence precision will only be 66%, although the algorithm operated flawlessly.

Recall (*R*) is defined as number of senses found divided by number of words merged to create pseudowords. For example, recall is 60% when five words are used to create the pseudoword and only three senses were correctly found (according to retrieval precision and retrieval recall).

There are several possible baselines applicable to the four introduced measures.

One is an algorithm that does nothing, resulting in a single set of 400 co-occurrences of the pseudo-word. This set has a retrieval precision rP of 50% compared to either of the two original 'senses', because for any of the two senses only half the 'retrieved' words match. This is below the allowed 60% and therefore not considered a correctly found sense. This means that retrieval Recall rR and Recall R are also both 0% and precision P in such a case (nothing correctly retrieved, but also nothing wrong retrieved) is defined to be 100%. Of course, if the allowed retrieval precision would have been set lower than 50%, then that single set of words would be considered as a correctly found sense. Therefore precision for this baseline would be 100%, Recall 50% and retrieval Recall 100%.

As mentioned in previous sections, several parameters significantly impact the quality of a WSI algorithm. One interesting question is: Does the quality of disambiguation depend on the type of ambiguity? In other words, would WSI based on sentence co-occurrences (hence on the bag-of-words model) produce better results for syntactically different senses or for lexically differing senses (as predicted by Schütze (1998))? This can be simulated by choosing two words of different word classes to create pseudowords, such as the (dominantly) noun *committee* and the (dominantly) verb *accept*.

Another interesting concern is the influence of frequency of the word itself or the sense (of the word, if there are several) to be found. The latter, for example, can be simulated by choosing one high-frequency and one low-frequency word, thus simulating a well-represented vs. a poorly represented sense.

The evaluation's aim is to test the described parameters and produce an average of precision and recall and at the same time make it completely reproducible by third parties. Therefore the raw BNC without baseform reduction or lemmatization or POS tags was used and nine groups, each containing five words, were picked semi-randomly (avoiding extremely ambiguous words, with respect to WordNet, if possible):

- high frequency nouns (N_h): picture, average, blood, committee, economy
- medium frequency nouns (N_m): disintegration, substrate, emigration, thirst, saucepan
- low frequency nouns (N_l): paratuberculosis, gravitation, pharmacology, papillomavirus, sceptre
- high frequency verbs (V_h): avoid, accept, walk, agree, write
- medium frequency verbs (V_m): rend, confine, uphold, evoke, varnish
- low frequency verbs (V_l): immerse, disengage, memorize, typify, depute
- high frequency adjectives (A_h): useful, deep, effective, considerable, traditional

- medium frequency adjectives (A_m): ferocious, normative, phenomenal, vibrant, inactive
- low frequency adjectives (A_l): astrological, crispy, unrepresented, homoclinic, bitchy

These nine groups were used to design four tests, each focusing on a different variable. The high frequent nouns are around 9 000 occurrences, medium frequent around 300 and low frequent around 50.

4.3.1. Influence of word class and frequency

In the first run of all four tests, sentence co-occurrences were used as features. In test 1, all words of **equal word class** were viewed as a single set with 15 elements. This results in $\binom{15}{2} = 105$ possibilities to combine two of these words into a pseudoword and test the results of the WSI algorithm. The purpose of this test is to examine whether there is a tendency for word senses of certain word classes to be easier induced. As can be seen from Table 4.1, sense induction of verbs using sentence co-occurrences performs worse compared to nouns. This could be explained by the fact that verbs are less semantically specific, thus needing more syntactic cues or generalizations - both hardly covered by the underlying bag-of-words model - in order to be disambiguated properly. Simultaneously, nouns and adjectives are markedly easier to distinguish through topical key words. These results apparently agree with the prediction made by Schütze (1998).

	P	R	rP	rR
N_{hml}	86.97%	86.67%	90.94%	64.21%
V_{hml}	78.32%	64.29%	80.23%	55.20%
A_{hml}	88.57%	70.95%	87.96%	65.38%

Table 4.1.: Influence of the syntactic class of the input word in Test 1. Showing precision P and recall R, as well as average retrieval precision rP and recall rR.

In Test 2, all three possible types of **of word class combinations** are tested, i.e. pseudowords consisting of a noun and a verb, a noun and an adjective and a verb with an adjective. Each combination has $15 \cdot 15 = 225$ possibilities to combine a word from one class with a word from another class. This test demonstrates potential differences between WSI of varying word class combinations. This corresponds to cases when one word form can either be a noun or verb, e.g. *a walk* vs. *to walk*, or a noun and adjective such as *a nice color* vs. *color TV*. However, the results in Table 4.2 illustrate no clear tendencies, other than that WSI of adjectival senses from verb senses appears to be slightly more difficult.

	P	R	rP	rR
N/V	86.58%	77.11%	90.51%	61.87%
N/A	90.87%	78.00%	90.36%	66.75%
V/A	80.84%	63.56%	81.98%	60.89%

Table 4.2.: Influence of the syntactic classes of the senses to be found in Test 2.

Test 3 was designed to gauge the **influence of frequency** of the input word. All words of equal frequency are taken as one group with $\binom{15}{2} = 105$ possible combinations. The results in Table 4.3 show a clear tendency for high-frequency word combinations to achieve better quality WSI vs. lower frequency words. The steep performance drop in recall becomes immediately clear when observing the retrieval recall of the found senses. This is not surprising, since with the low frequency words (each occurring only about 50 times in the BNC), the algorithm runs into the data sparseness problem that has already been pointed out as problematic for WSI (Ferret, 2004).

	P	R	rP	rR
high	93.65%	78.10%	90.25%	80.70%
med.	84.59%	85.24%	89.91%	54.55%
low	74.76%	49.52%	71.01%	41.66%

Table 4.3.: Influence of frequency of the input word in Test 3.

Test 4 shows the extent to which the overrepresentation of one sense over another influences WSI. For this purpose, three possible **combinations of frequency classes**, high-frequency with middle-frequency, high with low and middle with low-frequency words were created with $15 \cdot 15 = 225$ possible word pairs. Table 4.4 demonstrates a steep drop in recall whenever a low-frequency word is part of the pseudoword. This reflects the fact that it is more difficult for the algorithm to find the sense represented by the less frequent word. The remarkably high precision value for the high/low combination can be explained by the fact that in this case mostly only one sense was found (the one of the frequent word). Therefore recall is close to 50%, whereas precision is closer to 100%.

	P	R	rP	rR
h/m	86.43%	79.56%	92.72%	72.08%
h/l	91.19%	67.78%	90.85%	74.52%
m/l	82.33%	74.00%	85.29%	49.87%

Table 4.4.: Influence of different representation of senses based on frequency of the two constituents of the pseudoword in Test 4.

Finally it is possible to provide the averages for the entire set of test runs (1980 tests). The macro averages for the tests are $P = 85.42\%$, $R = 72.90\%$, $rP = 86.83\%$ and $rR = 62.30\%$, with the micro averages being almost the same. Using the same thresholds, with pairs instead of triplets yielded the following results: $P = 91.00\%$, $R = 60.40\%$, $rP = 83.94\%$ and $rR = 62.58\%$. In other words, more often only one sense is retrieved and the F-measures of $F = 78.66\%$ for triplets compared to $F = 72.61\%$ for pairs confirm an improvement of 6% by using triplets.

4.3.2. Window size

The second run of all four tests using direct neighbors as features failed due to the exacerbated data sparseness problem. Within BNC sentences there were 17.5 million significantly co-occurring word pairs, according to the log-likelihood measure. Even in this case, words with a low frequency showed a strong performance loss, when compared to high-frequency words. Compared to that only 2.3 million word pairs co-occured (significantly) directly next to each other. The overall results of the second run with macro averages $P = 56.01\%$, $R = 40.64\%$, $rP = 54.28\%$ and $rR = 26.79\%$ will not be discussed here in detail because they are highly inconclusive. The inconclusiveness derives from the fact that (contrary to the results of the first run) the results here vary strongly for various parameter settings and cannot be considered stable.

Although these results insufficiently demonstrate the influence of context representations on type of induced senses as they were supposed to, they allow for several other insights. Primarily, corpus size clearly matters for WSI, i.e. more data would have perhaps alleviated the sparseness problem. Secondly, while one context representation may be theoretically superior to another (such as neighbor co-occurrences vs. sentence co-occurrences), the effect various representations have on the data richness were by far stronger in the presented tests.

4.3.3. Examples

In the light of rather abstract, pseudoword-based evaluations some real examples sometimes help to clarify the results. Three words, *sheet*, *line* and *space* were chosen arbitrarily and some words (most significant co-occurrences) representing the induced senses are listed below.

Word : **sheet**

Sense 1 : beneath, blank, blanket, blotting, bottom, canvas, cardboard

Sense 2 : accounts, amount, amounts, asset, assets, attributable, balance

Word : **line**

Sense 1 : angle, argument, assembly, axis, bottom, boundary, cell, circle

Sense 2 : lines, link, locomotive, locomotives, loop, metres, mouth, north

Word : **space**

Sense 1 : astronaut, launch, launched, manned, mission, orbit, rocket, satellite

Sense 2 : air, allocated, atmosphere, blank, breathing, buildings, ceiling

These examples show that the differences found between word senses are indeed intuitive. They further demonstrate that the found senses are only the most distinguishable ones, with further senses missing regardless of their appearance in the BNC, albeit sometimes frequently. It seems that for finer grained distinctions the bag-of-words model is not appropriate, although it might prove to be sufficient for other applications such as Information Retrieval. Varying contextual representations might prove to be complementary to the approach presented here, thus enabling the detection of syntactic differences or collocational usages of a word.

4.4. Conclusions

It has been shown that the approach presented enables unsupervised and knowledge-free word sense induction on a given corpus with high precision and sufficient recall values. The induced word senses are inherently domain-specific to the corpus used. Furthermore, the induced senses are only the most apparent ones, while the type of ambiguity matters less than expected. However, there is a clear preference for topical distinctions over syntactic ambiguities. The latter effect is due to the underlying bag-of-words model, hence alternative contextual representations might yield different (as opposed to better or worse) results. This bag-of-words limitation also implies some senses to be found that would be considered as spurious in other circumstances. For example, the word *challenger* induces 5 senses, three of them describing the *opponent in a game*. The differences found are strong, because the senses distinguished are between a *chess challenger*, a *Grand Prix challenger* and a *challenger in boxing*, each having a large set of specific words distinguishing the senses.

There are several questions remaining open. Since the frequency of a word greatly impacts its correct disambiguation (using the presented methods), the question is then: to what extent does corpus size play a role in this equation as compared to balancedness of the corpus and thus the senses to be found? In other words - is the absolute amount of occurrences of a sense decisive (for that sense to be extractable), or is it the relative amount to the occurrences of the other senses of a word. If it is merely the absolute amount, rather than the absolute, then this implies that the corpus can be simply enlarged using nearly any text source even at the cost of extreme disbalancedness - the quality of the results still improves. Most likely the answer lies in-between. Moreover, some algorithms, such as the collocation extraction algorithms, tend to rely more on the absolute amount of occurrences of a sense. Other algorithms, such as the one presented above, tend to degradate in their performance for largely underrepresented senses, thus relying more on the relative amount of sense occurrences.

Another issue concerns the limitation of the presented algorithm, requiring that any sense to be induced must be representable by a fairly large amount of words. The question then is: Can this (or any other similar) algorithm be improved to discern 'small' senses (also called micro-senses) from random noise? This is of particular importance since it was shown that micro-senses actually improve the performance of WSD algorithms (Agirre et al., 2006). A combination with algorithms finding collocational usages of words offers a potentially feasible solution.

It has not been discussed, how to eventually fuse word senses for different word forms of the same lemma. The problem is, that without lemmatization it is quite possible to use the presented algorithm to find the meanings of *play* and *plays*. But how to find out, that the x-th meaning of the first word form corresponds to the y-th meaning of the second word form? Plain similarity algorithms (as discussed in the previous chapter) do not provide means to distinguish between true synonymy, cohyponymy and a host of other relations. Therefore it would not be possible to use them in their current form to compute this knowledge. However, a combination with the methods introduced in the next two chapters might offer suitable solutions.

Last but not least, the evaluation method employed can be used for an automatic optimization of the algorithm's own parameters.

5. The Morpheme Level

In the two preceding chapters, no method was defined to discern between words and word forms, cosequently leading to no operational distinction between these two notions. Any two word forms were assumed to be two different types (i.e. *likes* and *like* are as different from each other as *likes* and *hike*). Clearly, this simplification is misleading considering that in most languages word forms consist of morphemes. Despite some exceptions (for example *source* and *outsourcing*), most word formation processes generate word forms that are related either syntactically, semantically or both.

There are three major word formation processes: derivation, compounding and inflection. Whereas the first two typically involve a major modification of meaning (similar to figures of speech on the phrase level), inflectional morphemes usually modify their base morpheme only grammatically (e.g. degree of case, etc.) - although semantic modification is often involved as well (number: singular vs. plural, tempus: past vs. present vs. future). Thus, two word forms such as *plops* and *plopped*, differing only in their last inflectional morpheme, usually remain quite similar in their basic meaning. Therefore, models based on such simplifications (i.e. vector space model underlying Chapter 3), as well as all applications based on such models (for example Information Retrieval) are likely to be adversely affected. In fact, computing word (form) similarity based on neighbor co-occurrences often results in a partial list of morphological variants of the input word form.

The branch of linguistics concerned with the "study of the rules for forming admissible words" (WordNet), i.e. also with the above mentioned processes, is morphology. It has succeeded in giving accurate descriptions of these processes, as well as nearly perfect parsers and generators with accuracy ranging above the 90% mark for the languages that were examined. All these results are based on training sets or hard-coded knowledge about the corresponding language.

However, the method itself - how to achieve an adequate description of the morphology of a language, is a matter of debate (Kiparsky, 1982; Halle and Marantz, 1993). Neither of the methods currently discussed are implementable as algorithms such that they would function equally for a new language represented by a large corpus. This is partially due to the underlying problem of how to find constituents of the examined language automatically. Although word forms within sentences are divided by spaces in most languages, morphemes are glued together into word forms without a hint as to where the morpheme boundary is.

The first step towards a knowledge-free and unsupervised morphemic analysis of any language is to provide a method which identifies the boundaries between morphemes. Only then is it possible to analyze the composition and abstraction between the morphemic units. The resulting relations between morphemes should, in turn, be essentially analogous to the relations between word forms, according to Distributed Morphology (Halle and Marantz, 1993). This chapter presents an overview of previous research on this topic, as well as a new solution for knowledge-free morpheme boundary segmentation of word forms. In an additional step, the method for computing word similarity described in Chapter 3 is applied to the morphemes which were the output of the segmentation algorithm.

It is necessary to note however, that the underlying hypothesis in the present work assumes a concatenative (also called linear) morphology. Languages with non-concatenative (non-linear) morphology, which primarily involve modification rules that alter the "appearance" of morphemes instead of concatenating morphemes, are not covered by these algorithms. Several ideas on how to cover them are discussed, and one example (concerning alternations) is prototypically tested on German.

5.1. Introduction

In the beginning of this section, relevant work on morphology as handled in linguistics is briefly discussed. Subsequently, an operational definition of morphology for the purposes of the present work is introduced. This is necessary, because there are several different theories on the nature of word formation processes that were taken into account.

The various views differ marginally, given the most abstract definitions of the term *morphology*:

- morphology: The science of form. (Oxford English Dictionary)
- morphology: A study and description of word formation in a language including inflection, derivation, and compounding. (Webster's Third)
- morphology: Study of the rules for forming admissible words. (WordNet)
- morphology: The branch of grammar that deals with the internal structure of words. (Matthews, 1974)

A major distinction between morphological theories is the one between Lexical Morphology (Lieber, 1990; Sciullo and Williams, 1987; Selkirk, 1982; Kiparsky, 1982) and Distributed Morphology (Halle and Marantz, 1993; Halle and Marantz, 1994). This distinction can be summarized as the difference between a top-down and a bottom-up approach, similar to the classification into analytic and synthetic morphological analysis (Aronoff and Fudeman, 2004), respectively.

Lexical Morphology treats words as units having paradigmatic relations with each other, in particular those based on the internal structure of the words themselves. However, there is no single Lexical Morphology, except that all theories known as such have in common that both word formation rules and phonological rules apply to a single component of the grammar: the lexicon.

However, in **Distributed Morphology** there is no lexicon. Instead, there are contents of syntactical terminal nodes corresponding to the known term 'morphemes'[1]. Elements formerly distinguished as syntactic or morphological enter into the same types of constituent structures. Words, phrases and sentences are constructed following one set of morpho-syntactic rules. There are also lexicalized constructions - but essentially they are considered the same as idioms or figures of speech and are not necessarily constructed by syntax. Distributed Morphology proves to be a good explanation for phenomena such as the fuzzy border between when something is considered a word or a morpheme (Halle, 1997). In German, for example, direct negation is either expressed using an extra word: ***nicht*** *tun* (not to do) or as a different morpheme: *un-nötig* (un-necessary). In Czech, on the other hand, direct negation is realized with a negating morpheme which on other occasions can be used as a word: *nedělat* (not to do) vs. the exclamation: *To ne!* (Not that!).

There is no single unified theory of morphology, and a selection of other approaches include HPSG oriented, unification-based morphology (Krieger and Nerbonne, 1993) or finite state morphology (Koskenniemi, 1983). Further classifications differentiate between single-level declarative models (Selkirk, 1982) and multiple levels of representation (Koskenniemi, 1983). However, these finer grained distinctions are irrelevant for the purposes of the present work, which represents the very beginning of unsupervised and knowledge-free morphological analysis.

Contrary to the majority of approaches to morphology, the work in this chapter is not focused primarily on a proper formalization of morphology. Instead, it aims to provide a formalization of learning processes. Where possible, it attempts to remain compatible with evidence available from psycholinguistic studies about the development of morphological systems of children learning their first language (Berko, 1958; Tager-Flusberg, 1997). Therefore the learning phase is top-down: morphemes are acquired from word forms and the various relations between them. The second, bottom-up step, begins with morphemes, continuing to a model of relations between them. Relations between morphemes are acquired through their usage contexts.

In short, the primary goal of this chapter is to formalize and implement a simulation of morphologic learning processes and the representation of the acquired knowledge.

[1] This set of terminal nodes could naturally still be called a lexicon. But that would be misleading, because elements and the associated processes in such a lexicon would differ in meaning and function, as compared to elements in the lexicon known from Lexical Morphology.

5.1.1. Definition of the morpheme level

The definition of the morpheme level presented below presumes that there are at least two more levels, one below and one above. Language is thus assumed to consist of at least three levels:

$$l = \{lower, morpheme, upper\} \tag{5.1}$$

The lower level can either be that the letters as discussed in Chapter 2, or of phonemes. The choice for the upper level is far less restricted. According to Distributed Morphology, it might be the phrase or sentence level. In this case the intermediate level should not be labeled morpheme level, but morphosyntactic level. It not only covers the morphological construction of word forms, but the entire construction of phrases and sentences as well. Alternatively, assuming a tangible border between morphemic syntax and sentence syntax, the higher level might be the word form level. But especially within the Distributed Morphology theory it is controversial to distinguish between a morphemic and a word form level.

However, the procedures that learn to identify morpheme boundaries presented below depend by definition on this distinction because they attempt to split word forms into morphs. Additionally, some of the algorithms are based on word form appearance patterns within sentences. This is in line with psycholinguistic studies (Vihman, 1982): children may first acquire word forms and then begin to separate them into morphs and morphemes, based on usage of these forms and similarities between them.

Additionally, relations between morphs in this chapter are also acquired using the sentence level after dividing all word forms into their morphs. Hence both the elements of classical Lexical Morphology and Distributed Morphology are used to achieve optimal results.

The morpheme level[2] $L_{morphemes}$ is defined similarily to the example definition given in chapter 2 as a tuple of sets:

$$L_{morphemes} = (A_{morphemes}, C_{morphemes}) \tag{5.2}$$

Under the assumption that the next lower level is the letter level and the next higher level is anything from word forms to sentences or texts, then the atoms and complex units are defined as follows:

$$A_{morphemes} = C_{letters} \tag{5.3}$$

$$C_{morphemes} \subseteq A^*_{morphemes} \tag{5.4}$$

[2] It is possible to imagine a language without any morphology at all. In this case the morpheme level degenerates into a one-to-one mapping between the letter level and the word form level.

In other words, the atoms of the morpheme level $A_{morphemes}$ are the complex units consisting of the atoms of the letter level $C_{letters}$. The morphemic atoms themselves can again be compounded into higher complex units in various ways. Thus, concatenative morphology is the linear compounding of morphemic atoms. On the other hand, when considering non-linear alternation of morphemes, e.g. *knife* - *knives*, then the related word forms are handled as independent atoms and paradigmatic relations between them can be computed - revealing their relatedness.

As mentioned in Section 2.1, this definition does not provide an explicit definition of morphemes. Rather, it defines only morphs, similarly to how the next higher level defines word forms only and not explicitly words or lemmas. In this simplified instantiation of the model a morpheme is viewed as a group of interrelated morphs (paradigmatic relations), such as *er **gib**t, er wird **geb**en, er **gab*** (*he gives, he will give, he gave*). In Finnish, for example, there is a predominantly one-to-one relationship between morphs and morphemes. Exceptions occur due to alternation rules. A less simplified model might take non-linear rules into account, such as the observation that form differences between the various morphs of a morpheme are not entirely free.

Using this morpheme level definition, the formalisms introduced in Chapter 2 and the implementations in Chapter 3, it is possible to observe that, for instance, the prefix *de-* often co-occurs with the suffix *-ize* at the morpheme level. This leads to a hypothesis that some syntactic attribute exists between these two affixes which can be further verified, or unified with another hypothesis. Eventually, it could be possible to arrive at an improved hypothesis that might, among other things, state that verbs can be derived from adjectives by using the suffix *-ize.* However, at a first glance, morpheme boundaries are not directly observable in most languages, rendering such analyses impossible. How can knowledge about where a morpheme begins and where it ends be obtained?

To answer this question it is worthwhile to recall that the existence of the morpheme level is a hypothesis which does not have to be correct in all cases. The hypothesis states that there are ‘meaningful’ units, composed of atomic units of an underlying level (letters, for example), and that these units are distinguishable from those of the next higher level (word forms, for example). ‘Meaningful’ in this context implies syntactic and/or semantic attributes which are responsible for the eventually observed regularities.

It is then possible to define a similarity measure (not to be confused with the context similarity measure from Chapter 2) between units of the next higher level - word forms. This measure must be based on the underlying level, i.e. letters, because it is assumed to produce the morphemes. It can also incorporate further information such as contextual information available from observing complex units on the word level. If this approach is applied to a language lacking morphology, it will result in an empty language level: the set of atoms maps directly into the set of complex units without any combinations.

On the other hand, if there are morphemes to be found, syntagmatic and paradigmatic relations between them can be computed. For this purpose, the local context $K_{lc}(a_i)$ of a given morpheme a_i on the morpheme level $a_i \in A_{morphemes}$ is the set a of all morphemes with which the morpheme a_i occurs together in a complex unit $c_n \in C_{morphemes}$ of the same level:

$$K_{lc}(a_i) = \{a | a \in c_n \wedge c_n \in C_{morphemes} \wedge a_i \in c_n\} \tag{5.5}$$

The remaining variable at this point is interpreting exactly what the complex units $c_n \in C_{morphemes}$ are: word forms, according to Lexical Morphology, or phrases, according to Distributed Morphology. In both cases the definitions for statistical syntagmatic relations, global context and paradigmatic relations between morphemes and categories, possible attributes and correspoding compliances of morphemes remain exactly as stated in Chapter 2 and need not be reiterated at this point. Hence, the computation of such structural knowledge is possible in a manner similar to that on any other level.

These definitions also allow for a natural classification of existing morpheme boundary extraction algorithms, because they differ in the level they assume as the next highest, which other levels they include, and which similarity operator is used.

5.2. Related work

The task of finding morpheme boundaries can be divided into two steps: First, the identification of morphemes with the highest possible precision, regardless of possibly low recall. And second, enlargement of this knowledge by common machine learning methods, ideally with minimum precision loss. This combined knowledge (and perhaps knowledge from additional analyses) can later be reused in the first step to increase overall accuracy. However, as related work shows (see below), this division into two parts is not necessarily the only way to construct a good overall morpheme identification algorithm.

The most straightforward possibility to acquire morpheme boundaries is to employ a similarity measure based on letter differences between words as atoms of the word level. For such tasks the Levenshtein Distance (LD) (Levenshtein, 1965) (also known as edit distance) is typically used. It measures the least amount of single letter insert, replace or delete operations necessary to transform any word w_i into another word w_j. Used to identify morphologically related words, this method can yield useful results, but has a high error rate. If, for example, the threshold is one allowed operation, the LD would relate not only word forms like *house* and *houses*, but also *house* and *horse*, because in many cases unrelated word forms differ by just one letter.

5.2.1. Letter successor variety

A more elaborate mechanism measures the amount of different continuations possible after a given substring such as *in-* within the set of observed words. Combined with some weighting and a threshold, this method is known as the letter successor variety (LSV) (Harris, 1951; Hafer and Weiss, 1974; Déjean, 1998) or accessor variety (Feng et al., 2004). According to this method, correct morpheme boundaries are likely to occur at sudden peaks or increases of the LSV-value. Implicitly this represents a hypothesis: it assumes that if a substring of a word is a morpheme, then many different morphemes (or simply letters) will be observed before or after that string. In other words, it is assumed that morphemes can be almost freely combined into words. Therefore this can be called the hypothesis of **combination**, a rather direct reformulation of the definition of complex units above.

This approach can be modified in different ways (Harris, 1955), but generally, it has yet to be successfully employed for morpheme segmentation, see also Hafer and Weiss (1974) and Frakes (1992), because when applied to the entire list of distinct word forms, the 'noise' from numerous different possibilities and exceptions render the results nearly useless. This basic type of LSV is referred to as *plain global LSV* in the remaining chapter. Global, because it learns the morphological structure of any word form by comparing it to virtually **all** other word forms. Figure 5.3 shows its low performance of $F = 41\%$, as compared to any of the other methods. The plain global LSV has also been employed for generating 'good' candidate lists for postprocessing machine learning steps for morpheme segmentation (Déjean, 1998). Unfortunately, the author does not mention the quality of the results other than giving examples.

Plain global LSV represents a solid hypothesis and its specific problems may derive from the fact that it disregards some important linguistic aspects. Since it does not consider relations between morphemes, possible attributes and categories of morphemes etc., it seems highly improbable that the principle of combination alone suffices to handle morphology in its entirety. Most morphemes are not freely combinable with each other, but underly strong restrictions based on the mentioned relations, attributes and categories. Additionally, various types of exceptions exist, such as overlapping morphemes, non-productive affixes and other irregularities often originating from the influence of foreign languages.

5.2.2. Minimum description length

A more succesful approach to the task at hand is based on the addition of two further hypotheses: the **minimum description length model** (MDL) (de Marcken, 1995) and the hypothesis of **signatures** (Goldsmith, 2001).

The MDL is based on the hypothesis that morphemes not only exist and can be combined into word forms, but that the possible combinations underlie restrictions.

One for example is, that the entirety of possible morphemes represent an optimum of storage space needed for the morphemes and all potential combinations: either in the form of rules or a lexicon. Various reformulations of this hypothesis should not distract from the fact that they all represent the same basic hypothesis. For example, the hypothesis that morphemes are reused when creating words in an optimal manner such that the human brain must remember the least possible is obviously equivalent to MDL (Kazakov, 1997; Kazakov, 2001). Thus it is clear that Kazakovs work belongs to this classification as well.

A different approach to restricted combinability of morphemes was introduced by Bernhard (2006). Her algorithm is essentially a co-occurrence analysis in which substrings are likely to appear adjacently, modeled by conditional probability. Sudden drops exceeding the underlying mean frequency standard variation are taken as possible morpheme boundaries. This implicitly makes use of the relations between morphemes. Combined with several further restrictions and parameters, this was the most successful participating system in the MorphoChallenge 2005 (Kurimo et al., 2006): it had the highest precision and recall for correctly detecting morpheme boundaries.

According to the hypothesis of signatures, morphemes are divided into classes. Each class has an associated group of morphemes, i.e. the signature, where each can be combined with all morphemes of the given class. For example, the signature consisting of the suffixes *-er, -est* can be used only after words such as *great, large, big, ...*.

It is possible to formulate a simple algorithm making use of this hypothesis. It cuts each word form at one position based on the probability of the right substring following after the left, and based on the lengths of hypothesized stems and affixes. It then attempts to categorize various words into signatures (classes of words that have the same morphology), which consequently can be translated into rules. The quality of this algorithm improved in combination with the MDL (see de Marcken (1995), Brent, Murthy, and Lundberg (1995)). As such, the MDL represents a balance between over- or undergeneration of rules. The ideal balance is the most compressed representation of the data (the word forms), in the sense that the lowest neccessary number of morphemes and signatures should be used simultaneously. This removes all free parameters, creating a rather elegant solution. Kazakov (1997) offers a reformulation as a fitness function for genetic algorithms where each element of the population is a randomly created division of all words. The fittest elements are those requiring the least space for a stem, suffix and link lexicon. After a few iterations the fittest element is defined to be the result.

Another approach within the category of MDL-based algorithms is the one introduced by (Creutz, 2003). Adding maximum likelihood ML and later the Hidden Markov Model as a better implementation of the MDL model (Creutz and Lagus, 2005) to classify found morphemes, the authors constructed an improved version of a segmentation algorithm. It randomly segments words, subsequently measuring how well the segmentation fits to the incrementally built up knowledge base. This algorithm seems to score best in agglutinative languages (such as Finnish), tending to overgenerate in other cases. Moreover it requires information about the length and frequency distributions of morphemes of the input language (thus it is not entirely knowledge-free). Another noteworthy improvement was proposed by Argamon et al. (2004) who added a recursive component to the analysis. It could improve results on morphology-rich languages while maintaining steady results for morphology-poor languages.

It is possible that the MDL- and signature-based methods have an upper bound of achievable quality, particularly when considering irregular word formations. This is because correct morphological structure of these forms cannot be learned without further contextual information for each individual (irregular) word. Nevertheless, Goldsmith's implementation has been used as the baseline algorithm for other algorithms to be compared against, due also in part to its free availability. Morfessor, another implementation of the MDL model by Creutz and Lagus (2005) was used as a baseline algorithm for the MorphoChallenge 2005 and barely any of the contestants were able to outperform this baseline. It appears then, that this paradigm of global, wordlist based algorithms which essentially compares any word against the entire wordlist is currently best understood and has the best implementations available.

5.2.3. Semantic context

All hypotheses described so far employed word lists, the atoms of the word level. A new approach taken by Schone and Jurafsky (2001a) is to include information available from the contexts within words occur. This means to include the complex units of the word level into the analysis. This information can be obtained through various co-occurrence methods, and then used in many different ways to detect morpheme boundaries. The underlying hypothesis states that morphemes have certain functions complying with other words within sentences. When words are compared based on their global contexts, other words consisting of the same morphemes are likely to be similar.

The first approach of this kind (Schone and Jurafsky, 2001a) begins the other way around. First, a list of affix candidates is generated by counting the frequency of substrings. Using these candidates as a modification of the letter based similarity it is possible to generate a list of other possible word forms of the same lemma for each input word (for example *listen-ing* and *listen*). In the second step, contextual information (by means of LSA (Deerwester et al., 1990)) is used to determine whether the generated word pairs are semantically similar according to the corpus used (i.e., whether they appear in similar contexts). The goal of this method is not finding morpheme boundaries, but rather, rules of transformation which conflate different word forms together, if morphologically related. These rules, of course, also yield information about valid morphemes.

While the results of the Schone & Jurafsky method indicate high precision, there are problems with recall concerning derivation and other morphological processes, creating semantically opaque word forms. A possible solution would be reversing the algorithm: compute semantically similar words first, then analyze the string differences between an input word and its most similar words, and then generalize from the learned data. The new solution presented below can be seen as such an attempt.

5.2.4. Summary

Several partially interlinked hypotheses can be singled out that proved to be useful in finding morpheme boundaries:

- **Combination:** Morphemes are combined to create words.
- **Regularities:** Morphemes belong to classes and rules exist (or regularities) about which classes of morphemes can be combined with which morphemes from other classes.
- **MDL:** The Minimum Description Length Model assumes that the distribution of possible morpheme combinations results from an optimal storage space usage principle.
- **Dependence:** The co-appearance of morphemes underlies restrictions deriving from relations between morphemes.
- **Context:** Contextual information from sentences or other complex units built from words can be used to align words with similar or equal morphemes.

These hypotheses do not include any language specific information except for the linearity (concatenativeness) of morphology. Thus it can be assumed that corresponding algorithms will be valid for all human languages with linear morphology - or at least for the linear aspects of the respective morphology. The last hypothesis does not assume linear morphology and could probably be used to acquire non-linear morphology as well.

5.3. Combining context similarity and LSV

The approach to morpheme boundary detection introduced in this work is divided into two parts, representing a combination of three of the introduced hypotheses. The first part uses the hypothesis of combination, in the form of letter successor variety. It also makes use of context in the form of similarity, based on sentence co-occurrences, cf. Chapter 3. The second part uses a simplified variant of the regularities hypothesis. This is realized by training a trie-based classifier with knowledge generated in the first step. The classifier then finds new morpheme boundaries, succeeding if the regularities hypothesis holds.

5.3.1. First part: context and LSV

The first improvement takes care of the assumption that the letter successor variety used with the plain list of word forms has to put up with too much noise from irregularities. However, a list of word forms that all share one or more pieces of grammatical information (i.e., gender, case, number) would make the noise for the LSV method manageable. This approach produces even better results, if such a list can be generated for each word individually. For the word *running*, the list ideally contains word forms such as *swimming*, *walking*, *diving*, etc. The first part of the local LSV algorithm can be summarized as follows:

- w = input word
- obtain morphologically similar words
  - compute contextually similar words $A_s(w)$
- measure LSV
  - measure LSV for consecutive letter pairs in w from left, based on $A_s(w)$
  - measure LSV for consecutive letter pairs in w from right, based on $A_s(w)$
  - apply weights to each LSV value
    - substring frequency
    - bi- and trigram weights
    - inverse bigram weight
- each LSV value above threshold t is a morpheme boundary

Context

In order to obtain a list of word forms with the same syntactic information as a given input word form, it is neccessary to reflect on the possibilities of language. Regardless of the language in consideration, syntagmatic dependencies are likely to hold between adjacent word forms. For example, it is highly probable that after *goes*, any kind of lexicalized direction information will appear, such as *home*, *to* or *out*. However, in front of such morphemes or word forms, various verbs of motion will likely occur. Some or many of them will have grammatical markers identical to the input word form (such as *runs*, *walks* or *jumps*). These word forms are crucial for further analyses because of their morphological similarity to the input word.

Such considerations are formalized in Chapter 2 and any of the measures from Chapter 3 can be used to obtain a context representation. Assuming that syntagmatic compliance is more likely to hold between adjacent words, it might then be best to use neighboring co-occurrences and compute words with similar grammatical information. However, here the same problem applies as in Chapter 4, where neighboring co-occurrences proved inferior to sentence co-occurrences due to the significant differences with regards to data sparseness.

The first step then, is to compute all word forms co-occurring with a given word form w. The typical co-occurrences of the word form w (along with their significance values, as found by applying a significance formula) are represented by a vector $\overrightarrow{A_n}(w)$ (the index n means neighbor) in the assumed vector space of word forms. The second step compares word forms based on their neighbor co-occurrence vectors. From Chapter 3 it is given that the log-likelihood family of measures produce the best results for co-occurrence statistics and that the baseline is one of the best measures for similarity. Consequently, this combination (lg_base)is used for the present purposes. Thus, for any given word w, a vector $\overrightarrow{A_s}(w)$ of most similar words to w is retrieved. $\overrightarrow{A_s}(w)$ then 'contains' all words appearing in similar contexts as w.

For example, the most similar words to *running* are: *run (108), using (99), runs (71), working (70), operating (70), moving (67), getting (65).* The bracketed numbers x indicate that in the corpus used, there are x different word forms co-occurring significantly often both with the input word and the respective similar word.

LSV on context similarity vectors

The **local letter successor variety** (local LSV) is computed for each input word with its most similar words as the context. These words are then reranked using the edit distance. Subsequently, the 150 most similar words are retained for further processing. This is because keeping all similar words introduces some of the noise as discussed in Section 4.1.1. In contrast to this, the **global letter successor variety** is computed for each input word with all other words of that language as the context.

The words *clearly* and *early* are used as examples to illustrate the differences between local and global LSV. In one case *-ly* is a suffix, whereas in the other case it is not, although for both word forms a corresponding shorter word form exists, *clear* and *ear*. The contextually most similar words for *clearly* are: $\overrightarrow{A_s}(clearly) =$*closely, greatly, legally, linearly, really, weakly, ...* and for *early*: $\overrightarrow{A_s}(early) =$*rally, July, March, May, day, earlier, ...* From this point on these word forms in the most similar words vector are treated as a set of words (a similarity set) disregarding the ranking. Thus, $v \in \overrightarrow{A_s}(w)$ refers to a word v that has been computed as being similar to w. In the tables provided below, the character # marks the beginning and end of a word form.

The global LSV algorithm counts **for each transition between two letters** i within a given word form $w = \langle l_1, l_2, ..., l_n \rangle$ either all different letters encountered at position $i+1$ after the substring $\langle l_0, ..., l_i \rangle$, or at position i before the substring $\langle l_{i+1}, ..., l_n \rangle$, respectively. Thus, given the following assumptions:

$$w = \langle l_0, l_1, ..., l_n \rangle \tag{5.6}$$

$$v \in \overrightarrow{A_s}(w) \tag{5.7}$$

$$v = \langle k_0, k_1, ..., k_m \rangle \tag{5.8}$$

the plain left and right letter successor variety $plsv_l(w,i)$ and $plsv_r(w,i)$ using the similar words $\overrightarrow{A_s}(w)$ is the number of different letters encountered at the respective positions. For example, in Table 5.2 a total of 76 out of 150 most similar words of *clearly* end with *ly*. For all these words, in front of *ly* 16 distinct letters were observed. Hence, the plain right letter successor variety is $plsv_r(clearly, 5) = 16$.

Finally both values are combined to obtain a final undirected value $plsv(w,i)$ for each transition - by either taking the minimum (Feng et al., 2004) or by summing the values:

$$plsv(w,i) = plsv_l(w,i) + plsv_r(w,i) \tag{5.9}$$

As previously stated, it is possible to detect morpheme boundaries using the plain LSV value $plsv(w,i)$. According to (Hafer and Weiss, 1974), several strategies are possible. One is to define peaks of high LSV values as indicators of morpheme boundaries. This strategy needs at least some finetuning, because as shown in Tables 5.1 and 5.2 the first and last positions in particular tend to represent local maxima, independent of whether or not they represent a morpheme boundary. An alternative strategy is to take all LSV values above a certain threshold. Nevertheless, any such strategies based on the plain LSV value are prone to several effects that influence the results.

Substring frequency weight

One effect is that the frequency of the respective substrings plays an important role: According to Table 5.1, for *early* only 6 words out of 19 ending with *-y* ended with *-ly*, as compared to 76 out of 90 words for *clearly*. As an improvement over the original LSV method, this ratio can be used to obtain a confidence weight for how 'trustworthy' the computed LSV at the particular position is. For *ear-ly* it is $4/19 = 0.2$, as compared to $76/90 = 0.8$ for *clear-ly*, further widening the difference between the two types of *-ly*. Thus, $frq_l(w,i)$ (or simpler $frq_l(i)$) refers to the frequency of the substring to the left of position i within the similar words $\overrightarrow{A_s}(w)$ of w and $frq_r(w,i)$ to the right.

input word:	#		e		a		r		l		y		#
LSV left:		40		5		1		1		2		1	
LSV right:		1		2		1		4		6		19	
freq. left:		150		9		2		2		2		1	
freq. right:		1		2		2		6		19		150	
bigram left:				0.2		0.2		0.5		0.0			
trigram left:						0.0		0.1		0.0			
bigram right:				0.5		0.0		0.1		0.3			
trigram right:				0.0		0.0		0.2					
bigram weight:				0.2		0.5		0.0		0.1			
score left:				0.0		0.0		0.5		1.7			
score right:				1.0		0.0		0.7		0.2			
final score :				1.0		0.1		1.2		2.0			

Table 5.1.: Depicting the local LSV algorithm for *early*. Weights are rounded.

However, this frequency ratio alone does not take into account that some phonemes are represented by more than one letter, such as *th*. This causes the frequency denominator to be 'carried away' by one or more positions. A solution for this problem can be devised from the frequency ranking of bigrams (or trigrams, etc.) in the entire corpus. It is assumed that compared to common bigrams representing two phonemes, bigrams representing only one phoneme are very frequent. Hence, a distribution of scores between 0.0 for the least frequent bigrams and 1.0 for the most frequent bigrams can be created. This score $bw(w,i,i+1)$ (or simpler $bw(i,i+1)$) for a bigram consisting of the letters i through $i+1$ of the word w can then be used to compute the weighted average of the substring frequency weights $fw_l(w,i)$.

Because single phonemes represented by the three letters such as *sch* in German exist, it is necessary to consider trigrams $tri(i,i+2)$, 4- or n-grams as well. They can be variously combined and for this work one of the most simple combinations has been chosen: to take only bigrams and trigrams and choose the one with the higher score. Thus, the final substring frequency weight $fw_l(w,i)$ is computed as follows:

$$fw_l(w,i) = \left\{ \begin{array}{ll} \frac{\frac{frq_r(i)}{frq_r(i+1)} + bw(i,i+1)\frac{frq_r(i)}{frq_r(i+2)}}{1+bw(i,i+1)}, \text{ if} & \begin{array}{l} bw(i,i+1) > \\ tri(i,i+2) \end{array} \\ \frac{\frac{frq_r(i)}{frq_r(i+1)} + tri(i,i+2)\frac{frq_r(i)}{frq_r(i+3)}}{1+tri(i,i+2)} & \text{, otherwise} \end{array} \right\} \tag{5.10}$$

Note: in order to compute the frequency weight for the left LSV value the frequencies of the substrings to the right need to be taken and vice versa:

$$fw_r(w,i) = \left\{ \begin{array}{ll} \frac{\frac{frq_l(i)}{frq_l(i-1)} + bw(i-2,i-1)\frac{frq_l(i)}{frq_l(i-2)}}{1+bw(i-2,i-1)}, \text{ if} & \begin{array}{l} bw(i-2,i-1) > \\ tri(i-3,i-1) \end{array} \\ \frac{\frac{frq_l(i)}{frq_l(i-1)} + tri(i-3,i-1)\frac{frq_l(i)}{frq_l(i-3)}}{1+tri(i-3,i-1)} & \text{, otherwise} \end{array} \right\} \tag{5.11}$$

input word:	#		c		l		e		a		r		l		y		#
LSV left:		28		5		3		1		1		1		1		1	
LSV right:		1		1		2		1		3		16		10		14	
freq. left:		150		11		4		1		1		1		1		1	
freq. right:		1		1		2		2		5		76		90		150	
bigram left:				0.4		0.1		0.5		0.2		0.5		0.0			
trigram left:						0.1		0.1		0.1		0.1		0.0			
bigram right:				0.5		0.2		0.5		0.0		0.1		0.3			
trigram right:				0.1		0.1		0.0		0.0		0.2					
bigram weight:				0.1		0.5		0.2		0.5		0.0		0.1			
score left:				0.1		0.3		0.0		0.4		1.0		0.9			
score right:				0.3		0.9		0.1		0.0		12.4		3.7			
final score :				0.4		1.2		0.1		0.4		**13.4**		4.6			

Table 5.2.: Depicting the local LSV algorithm for *clearly*. Weights are rounded.

Inverse bigram weight

Whereas the frequency ratio effectively removes faulty morpheme boundaries around multiletter phonemes, it does not prevent faulty boundaries within such phonemes. It is therefore probable that morpheme boundaries are detected in instances like *t-heater* or *t-hing*, although based on the high frequency of the corresponding bigram *th* it is clear that such boundaries are highly improbable. This ‘improbability’ is directly translated into an inverse bigram weight $ib(w,i)$ by subtracting the bigram weight $bw(i-1,i)$ of the bigram around the position i from 1.0:

$$ib(w,i) = 1.0 - bw(i-1,i) \tag{5.12}$$

Figure 5.6 shows that adding this weight does slightly improve overall performance by removing false positives while retaining true positives.

Weighted LSV value

The two introduced weights, substring frequency weight $fw_l(w,i)$ and inverse bigram weight $ib(w,i)$ are combined by multiplying them with the plain LSV value to create a weighted LSV value $lsv_l(w,i)$:

$$lsv_l(w,i) = plsv_l(w,i) \cdot fw_l(w,i) \cdot ib(w,i) \tag{5.13}$$

$$lsv_r(w,i) = plsv_r(w,i) \cdot fw_r(w,i) \cdot ib(w,i) \tag{5.14}$$

The final weighted LSV value then is computed similarly to the plain LSV value by adding the left and right LSV values:

$$lsv(w,i) = lsv_r(w,i) + lsv_l(w,i) \tag{5.15}$$

The difference between the final left and right score can be used to classify morphemes, in addition to indicators such as length or frequency (Bernhard, 2006). If the right score is greater than the left score, the morpheme discovered to the right is probably a suffix, and a prefix otherwise. As shown in Section 5.4, it is also possible to measure co-occurrences of morphemes, employing them as fatures to cluster morphemes into classes. A model as described in (Creutz and Lagus, 2005) for a more proficient tagging of morpheme categories may prove to be useful, too. In this respect it is also important to mention the mechanism described by Dejean (1998). If, for a given string such as *light*, more than half of its occurrences were analyzed to be a stem with various suffixes such as *light-s*, *light-er*, *light-NULL* then the other smaller half might as well be assumed to be a suffix (or several affixes) such as *-ning* in *light-ning*. All these possibilities cannot be covered in the scope of this work, but are subject of further research.

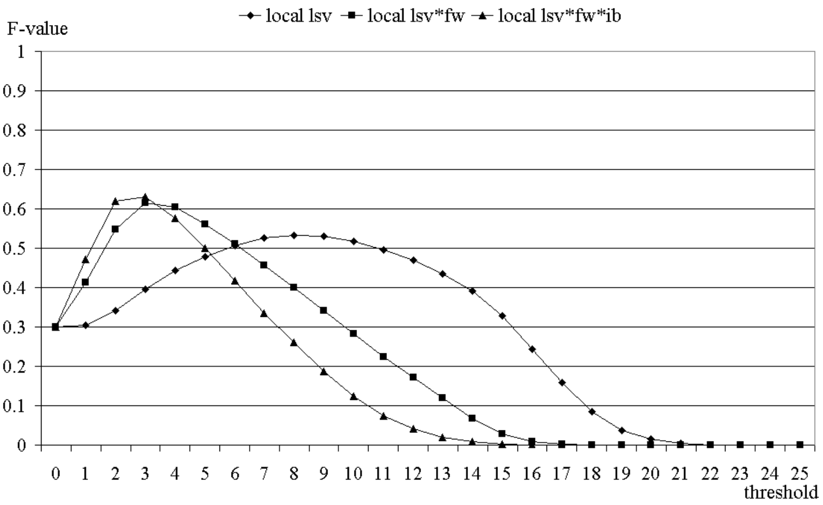


Figure 5.1.: Performance of the local LSV method and the two introduced improvements, frequency weight (fw) and inverse bigram (ib).

Independently from the type of morpheme found, high values of the final score are used as indicators for morpheme boundaries. Figure 5.1 and 5.2 show that both on local LSV and global LSV the proposed weights significantly improve the performance (measured on an 11 million sentences German corpus, see Section 5.3.3). They represent evaluations with varying threshold settings and they clearly show that choosing a treshold such as 5 results in a good tradeoff between precision and recall. Figure 5.3 illustrates that while the overall performance peaks from the previous two figures are approximately the same, the precision at the corresponding peak is much higher for the local variant, as compared to the global LSV. This indicates that using these results as learning input for another algorithm is potentially more successful on the local data. This indication is shown to be true in Figure

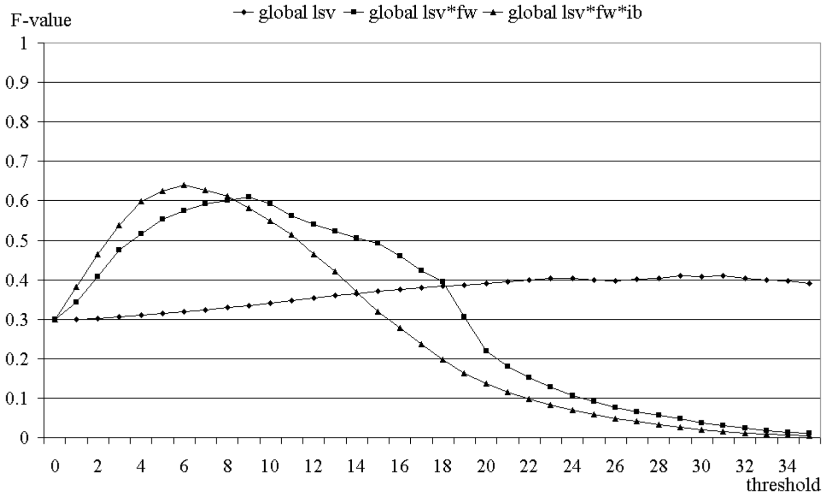


Figure 5.2.: Performance of the global LSV method and the two introduced improvements, frequency weight (fw) and inverse bigram (ib).

5.6.

The exact setting of the threshold should theoretically be dependent (possibly logarithmically) solely on the number of different letters (or rather phonemes) in a given language. In reality, the lack of co-occurrence observations (and subsequent lack of contextual information) if the corpus is not sufficiently large, can effectively prevent the discovery of a valid morpheme boundary with an otherwise correctly set threshold. Low frequency of one of the participating morphemes further aggravates this problem.

In fact, the size of the corpus is a much deeper problem for this algorithm. From Zipfs law (Zipf, 1949) follows that in any corpus the word frequency distribution is skewed towards most words having a very low frequency. But if a given word occurs only a few times, then only a few words can be determined as significant neighbor co-occurrences and on these grounds it is difficult to find similar words to the input word. Consequently, in a small corpus even otherwise common words might be too infrequent for this algorithm as it is based on the assumption that about 150 similar words can be found. Furthermore, for languages such as Finnish this problem is intensified - due to the large amount of various word forms each one occurs much less frequently in a similar sized corpus and thus it is less probable to obtain a sensible set of semantically similar words for any given input word unless the corpus size is significantly increased.

5.3.2. Second part: trie based generalisation

One way to circumvent the sparse data problems of the LSV-based algorithm is to use its result in an attempt to generalize them through other means. For this task it

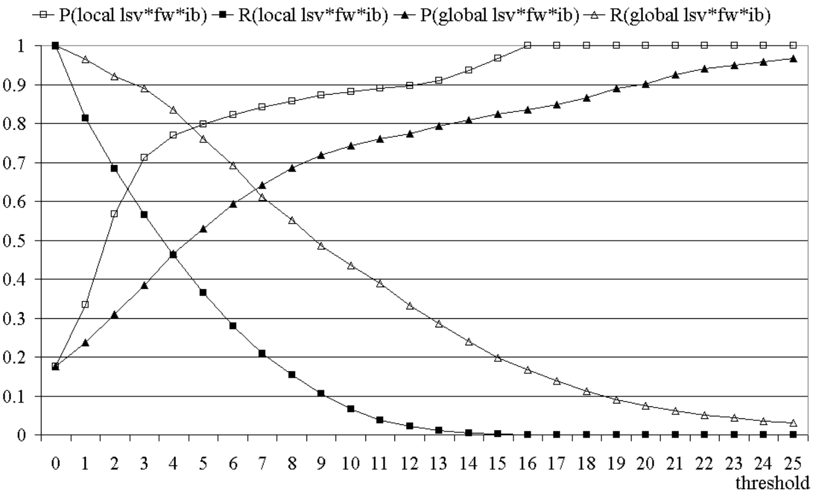


Figure 5.3.: While maximum performance of the local and global LSV method appeared similar, precision at the peak is significantly higher for the local LSV method.

is feasible to use affix trees such as a trie (Fredkin, 1960) or a PATRICIA compact tree (PCT) (Morrison, 1968). Variations of this data structure have already been used for classifications of word strings and their affixes (Cucerzan and Yarowsky, 2003; Sjöberg and Kann, 2004), as well as for other applications. The particular implementation used here is the same as in (Witschel and Biemann, 2005).

One method to resolve the representativity problems would be to use a combination of local and global LSV or use the results of the local LSV as a heuristic for MDL algorithms (some begin with a word list with random set morpheme boundaries). For this work tries were chosen, because of their simplicity.

A PCT can be trained to classify affixes in the following manner: The input consists of the string to be classified, i.e. *clearly*, and the classification, such as *-ly* or 2. This means that either the suffix *-ly* needs cutting or, more simply, that the boundary is at the second position from the right end of the word. From the examples used in the previous section one valid training instance can be acquired: *clearly ly*. The corresponding reversed uncompressed tree structure would have one node, *y* with one possible decision *ly=1* (with the frequency of 1). This node would have a child node *l* containing the same information.

In order to use such a tree for classification, first the deepest possible node in the tree structure must be retrieved. In the instance of *daily* it would be the second node *l*, because the next child node is a mismatch between *i* of *daily* and *a* stored in the tree. The probability for any class of the found node is the frequency of that class divided by the sum of all frequencies of all classes of that node. A threshold (set to 0.51 in these experiments) can be used to discern clear and unclear cases. For *daily*, the probability is 1.0, because there are no other classes stored in the found

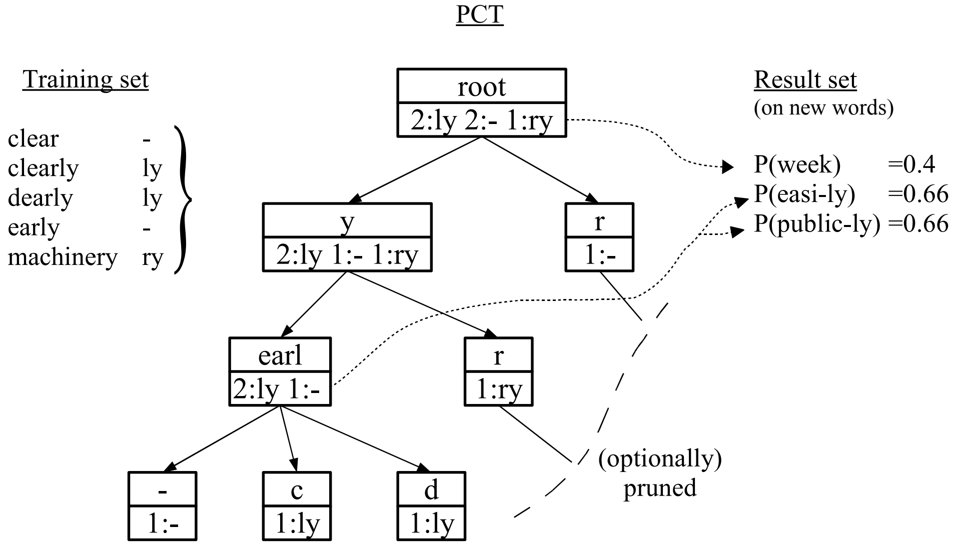


Figure 5.4.: Illustration of training a PCT and then using it to classify previously unseen words.

node. Figure 5.4 gives an example of training a PCT and using it for classification of previously unseen words.

It is noteworthy that such classification trees have strong generalization abilities while retaining all exceptions. Such a suffix tree, trained on three items *clearly ly*, *strongly ly* and *early NULL* correctly annotates hundreds of words ending with *-ly* while remembering the single exception of *early*. However, it can only produce this single exception, so overtraining remains a distinct possibility. Pruning, a common technique to cut seemingly redundant branches of the trie for higher efficiency, has not been used here.

For the present special case of affix classification it is important to decide whether the class to be trained is a prefix or suffix. This is because it does not help to know that a word begins with *mo* to guess whether its trailing *s* is a suffix or not. Therefore a simple strategy to train two distinct classifiers is employed. Given an input string with n boundaries, the outermost is selected recursively as a class and cut off for the next training item. When closer to the right side, the suffix classifier is trained with it. Otherwise the prefix classifier is trained. For example, the word *dis-similar-ly* results in one training item for the suffix classifier *dissimilarly ly*, and one for the prefix classifier *dis similar*.

After training both classifiers as described, they can be used as a morpheme boundary detection algorithm. For any input word both classifiers retrieve their most probable classification. In rare cases this may produce unfitting classifications, such as *-ly* for the input word *May*. Such cases are discarded. Next, the longer of the two classes is taken and a morpheme boundary is set according to the classification. Thus, for the example *undertaken*, the affix *under-* is favored over the affix *-en*. A length threshold of 3 is used to determine a valid classification, meaning that the new affix or the remaining word must equal or surpass the length of this threshold in order to avoid degenerated analyses such as *s-t-i-l-l*. Subsequently, the classification algorithm is recursively applied to both parts again. As a result, long words such as *hydro-chem-ist-ry* are completely analyzed, whereas the initial local LSV-based algorithm failed to analyze them at all, see also the improved results in Table 5.5.

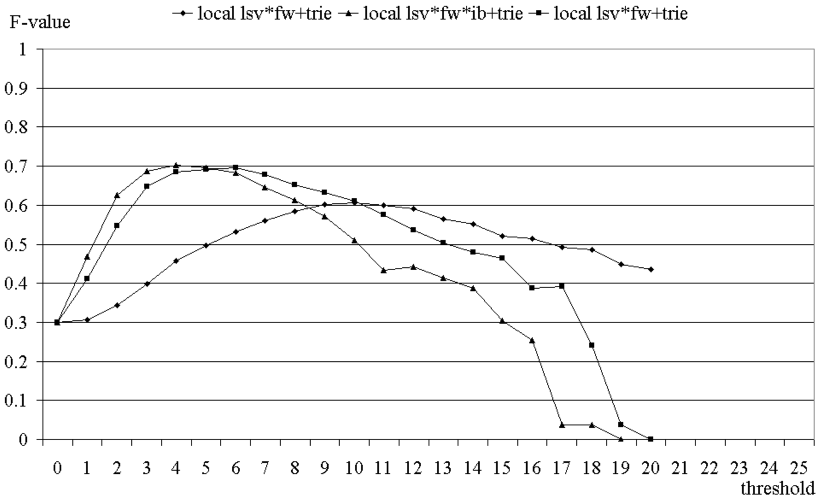


Figure 5.5.: Using the trie classifier on each type of input data generated by the local LSV.

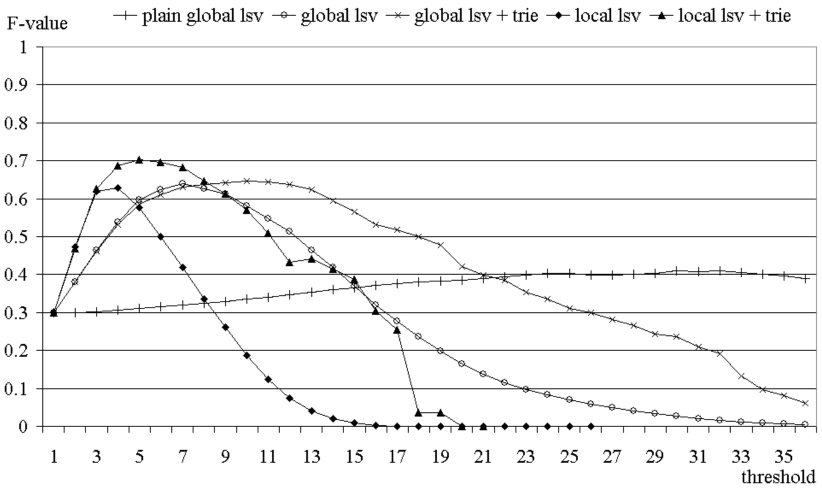


Figure 5.6.: Using the trie classifier on each type of input data generated by the local LSV.

5.3.3. Evaluation of morpheme boundary detection

There are approximately as many different evaluation methods as there are algorithms for any of the tasks of morpheme identification, morphologic segmentation, and lemma - word form clustering. Most algorithms such as (Goldsmith, 2001) and (Schone and Jurafsky, 2001a) are evaluated by measuring accuracy and productivity of word form retrieval. Since in the precision and recall figures provided below only morpheme boundary segmentation is measured, they cannot be compared to the evaluations of the cited algorithms. But since one of the two main goals of the presented algorithm is to produce correct morpheme segmentations, two evaluations are provided, measuring the precision and recall of proper morpheme boundary detection: the results of the MorphoChallenge 2005 and another additional evaluation based on the German and English parts of CELEX (Baayen, Piepenbrock, and Gulikers, 1995).

Unsupervised segmentation of words into morphemes - Challenge 2005

This challenge is a successful first attempt to organize a standardized evaluation framework analogous to SENSEVAL 2 (SENSEVAL 2, 2001) for the word sense disambiguation task. The challenge data consisted of the word lists of three languages: English, Finnish and Turkish. The corresponding gold standard of correct morpheme segmentations for Finnish is based on the two-level morphology analyzer FINTWOL from Lingsoft, Inc[3]. The English gold standard is based on the CELEX English data base and the Comprehensive Grammar of the English Language by Quirk et al. (1985). The Turkish linguistic segmentations were obtained from a morphological parser developed at Bogazii University.

The results of the challenge allow initial comparisons between existing algorithmic paradigms. However, several problems are associated with the challenge:

1. Only three languages were available, two of which were agglutinative (Finnish and Turkish) and one predominantly isolative (English).
2. For the three languages only word lists were available. With most algorithms this is not problematic, as they do not require a corpus. However, when an algorithm such as Schone & Jurafskys (2001a) or local LSV is part of the competion, the size and quality of the underlying corpus becomes a crucial parameter.

The next challenge therefore comprises of several complete corpora of raw text, as well as word lists from which the algorithms are supposed to learn. In the future it should also contain at least one fully flectional language such as Czech (along with a corresponding corpus).

[3] http://www.lingsoft.fi/

Nevertheless, the results (see Kurimo et al. (2006) or Table 5.3 for a selection) of the challenge yield several observations and conclusions. With respect to the F-measure, the described algorithm performed second-best for English and Turkish, and fourth for Finnish. The conditional probability based algorithm by Bernhard (2006) outperformed any other algorithm, with an average F-measure of 63%. However, with respect to precision the LSV based algorithm was best for Finnish and Turkish, reaching figures as high as 79.9% and second for English. There is an exception with the algorithm by Keshava and Pitler (2006) outperforming all other algorithms on English, but this algorithm does make use of language-specific knowledge[4] and is not considered.

	Finnish	Turkish	English
Dang and Choudri (2006)	61.3	55.4	49.8
Bernhard	64.7	65.3	62.4
Bordag	48.3	57.0	61.7
Atwell	61.2	55.9	55.7
Morfessor M3	66.4	70.7	66.2

Table 5.3.: Overview of a selection of participating systems (F-value) of the MorphoChallenge 2005.

Table 5.3 shows that none of the submissions outperformed the baseline set by Morfessor (Creutz and Lagus, 2006), including the two best-performing algorithms (by a notably small margin though). Taking only the three best algorithms, Morfessor, Bernhard's algorithm and the LSV based algorithm one obvious conclusion emerges. Although these algorithms use varying underlying hypotheses and data, they all yield nearly equally good results. This raises the question, whether it is possible to combine these three hypotheses to obtain even better overall results?

The answer is clearly yes, as shown by the out-of-competition committee classifier algorithm CHEAT (Atwell and Roberts, 2006). This is a simple voting algorithm which takes the decisions of each algorithm sent to the challenge as input. It decides to set a specific morpheme boundary if a sufficient number of other algorithms agree on that particular boundary. Of course, an algorithm combining knowledge in a more sophisticated manner could make even better use of the different types of evidence, but see also Section 5.4.2.

Another fact to consider is that the corpora available for the Finnish and Turkish entries of the presented algorithm to the challenge were small - an order of magnitude smaller than for English. Additionally both languages have nearly an order of magnitude more different word forms for the same amount of text compared to English. Hence, contextual information for most Turkish and Finnish word forms

[4] In short, the algorithm attempts removing a substring from a specific word form. If the new, shorter word form is an existing word form, then the substring is a suffix. Obviously, this works well only for English.

was very sparse. This explains, the extremely low recall of $R = 44.8\%$ for Finnish and $R = 47.9\%$ for Turkish as opposed to 62.2 for English. On the other hand, these two languages have a more concatenative morphology than English, and English is also more isolative. Therefore it is generally more difficult to find morpheme boundaries, explaining the lower precision of $P = 61.2$ for English, as opposed to $P = 70.3$ for Turkish.

Evaluation on English and German CELEX

The languages used to evaluate the algorithm separately from the MorphoChallenge 2005 are German and English. The German corpus is a part of the newspaper corpus "Projekt Deutscher Wortschatz" (Quasthoff, Richter, and Biemann, 2006) containing 11 million sentences (out of 35 total. The English corpus also consists of newspaper texts, but contains only 13 million sentences. Information about word form stemmings and correct morphology segmentation was acquired from CELEX (Baayen, Piepenbrock, and Gulikers, 1995).

Due to the large amount of sentences, the computation of the co-occurrences and the similarities based on them takes up by far the most computation time (several hours on a modern PC). Computation of similarity was optimized so that not every word had to be compared with every other: cues from sentence co-occurrences are used in order to single out candidates of words possibly sharing co-occurrences. Once the similarity data is available, the computation time of the LSV based algorithm is negligible in comparison to the previous steps. This applies to the global LSV algorithm as well, which does not make use of similarity data. The most frequent 145 000 word forms were analyzed by the algorithm for the evaluation.

In the evaluation, the overlap between manually and automatically tagged morpheme boundaries is measured. Precision is the number of found correctly identified boundaries versus the total number of found boundaries. Recall is the number of correctly found boundaries versus the total number of boundaries present in the gold standard. The F-measure, or the harmonic mean of precision and recall, compares the various variants of the algorithm with each other.

For example, a complex word analyzed by the algorithm, such as *ent-zünde-t* (inflamed), would have one correctly detected boundary for the prefix *ent-* and one wrongly detected boundary, because the correct analysis would be *ent-zünd-et*. However, this particular word is not present in CELEX, and hence both variants should be counted as 'wrong'. Therefore words not present in CELEX were excluded from the evaluation.

Only one parameter (threshold) affects the performance of the LSV based algorithms, but there are several variants of the algorithm to be tested: local vs. global, plain LSV vs. weighted LSV as well as these variants combined with the trie classifier as a second step. Therefore several tests (one for each variant) were conducted with a sufficiently large threshold interval of at least [0...25]. The global LSV algorithm was measured on a greater range [0...35] of thresholds because comparing against the entire word list naturally produces more various letters seen after any substring. These detailed tests were conducted only with German data.

The results of the first test (see Figure 5.1) demonstrate that the performance of the local LSV algorithm in the absence of weights peaks[5] at about 53%. Adding the average weights improves peak performance to about 62%. Finally adding the inverse bigram weight improves results by a slight margin to about 63%.

The second test (see Figure 5.2) measures the global LSV algorithm. The pattern of the results resembles the first test, except that the unweighted plain LSV algorithm peaks much lower (at 41%) than the local plain LSV algorithm. This strongly supports the hypothesis (formulated in Section 5.2.1) of too much noise when comparing any word against the entire word list.

The peak performances of both local and global LSV algorithms (with the weights) seem to be roughly equivalent, but Figure 5.3 shows that at their corresponding peaks the local LSV has a better precision of 71% compared to the precision of 59% for the global LSV algorithm. This means that the global LSV algorithm represents a precision/recall tradeoff favoring recall. However, for further learning steps recall is less important than precision.

The fourth test (see Figure 5.5) shows that using the trie classifier as a method to generalize knowledge produced by the local LSV variants indeed improves peak performance to roughly 71%. Again, combining average weights and inverse bigrams results in the best performance, while learning from the data produced by the plain LSV method is worse (only 60%). An interesting point is the shakiness of the observed quality of learning towards the outer end of the continuum of thresholds. This happens when the recall of the initial data to be learned from becomes extremely low and only a few very frequent suffixes are learned. These are subsequently found with an extremely high precision, but nothing else.

At this point it is unnecessary to reiterate a test demonstrating that using weights on the global LSV algorithm and the subsequent learning step improves performance. Instead, the fifth detailed test (see Figure 5.6) compares the performance of all variants with the weights: the global LSV, the local LSV, the trie classifier applied to both and finally the plain global LSV as the baseline, because it is the simplest algorithm. The results confirm the hypothesis that precision is more important for learning. This is because learning from the global LSV barely improves overall results, whereas learning from the local LSV algorithm yields the best results.

Identical tests are repeated with the smaller English corpus. Using the weights instead of plain LSV shows the exact same imporvements as observed with Ger-

[5] The peak is reached at a threshold of 3, which is close to the hypothesized ideal threshold of logarithm of different letters $\ln 26 = 3.26$.

man. This shows that the weights most probably do not represent language-specific information, although it does not rule out the possibility that they are specific for Germanic languages. Using the trie classifier on top of the local LSV variant again greatly improves overall performance (see Table 5.4). However, using the trie classifier on top of the global LSV algorithm performs with $F = 57\%$ slightly better than the $F = 55\%$ of the local LSV + trie. This indicates that the better performance of the local LSV + trie combination can only be achieved, if the corpus is sufficiently large. The following example illustrates the reasons:

The local LSV algorithm did not find a boundary after the prefix *mis-* in any of the 162 cases of word forms beginning with it, because their frequency was too low. Subsequently the trie classifier lacked evidence for this substring to be a prefix and could not learn it. On the other hand, the global LSV detected several instances of *mis-* as a prefix, some of which were wrong, i.e. *mis-ter.* The trie classifier then gained sufficient evidence to learn from it. However, due to imprecise quality of the initial data produced by the global LSV, the trie is then unable to distinguish which of the substring *mis-* are prefixes and which simple parts of morphemes. Consequently, almost any occurrence of that substring was annotated as a prefix. The resulting recall is much higher, but precision is lower.

	German		English	
	local LSV	global LSV	local LSV	global LSV
threshold	3	6	3	6
LSV precision	71.15	59.37	79.48	51.61
LSV recall	56.42	69.18	8.80	78.86
LSV F-measure	62.94	63.90	15.84	**62.39**
threshold	4	9	3	8
combined precision	64.14	62.12	55.90	45.98
combined recall	77.50	67.54	59.50	74.27
combined F-measure	**70.19**	**64.71**	**57.64**	56.80

Table 5.4.: Peak performance for each combination under the assumption that ideal threshold settings for each algorithmic variant are known. The highest score for each method is highlighted.

Table 5.4 shows a rather surprising finding. Applying the trie classifier on the data produced by the global LSV algorithm improved the performance only insignificantly for the German data. But while the peak performance on English for the global LSV is 62.39%, applying the trie classifier on that data actually **decreases** performance, reaching 56.80%. For the same threshold of $t = 6$ the F-measure yields 56.70%, precision 42.83% and recall 84.23%. This means that while the trie classifier was able to improve recall, the precision drop traded for the higher recall is more costly. Considering that the trie classifier is essentially also a global learning algorithm, an important conclusion can be drawn from this: When

local LSV + trie	CELEX	correct b.	wrong b.	missed b.
Orient-ier-ung	Orient-ier-ung	2 (2)		
Orient-ier-ungen	- (not in CELEX)	(2)		
Orient-ier-ungs-hilf-e	-	(2)		
Orient-ier-ungs-hilf-en	-	(4)		(1)
Ver-trau-ens-mann	-	(4)		(1)
Ver-trau-ens-sache	-	(3)		(1)
Ver-trau-ens-würd-igkeit	-	(4)		(2)
senegales-isch-e	senegalesisch-e	1 (2)	1	
sensibelst-en	sens-ibel-sten	(1)	1	2 (1)
separat-ist-isch-e	separ-at-istisch-e	2 (3)	1	1
tris-t	trist		1 (1)	
triump-hal	triumph-al		1 (1)	1 (1)
trock-en	trocken	(1)	1	
unueber-troff-en	un-uebertroffen	(2)	2	1 (1)
total		5 (29)	8 (2)	5 (8)
CELEX	$P = \frac{5}{5+8} * 100 = 38.5\%$, $R = \frac{5}{5+5} * 100 = 50.0\%$			
true performance	$P = \frac{29}{29+2} * 100 = 93.5\%$, $R = \frac{29}{29+8} * 100 = 78.4\%$			

Table 5.5.: Comparison of morpheme boundaries by algorithm and CELEX, as well as a sample evaluation. Values in brackets are based on the author's perception of correct morpheme boundaries.

two algorithms utilize the same paradigm, they are likely to produce similar errors. Thus, combining two such algorithms will not likely produce better results. But as observed with the local LSV + trie combinations, high initial precision gives leeway to trade a small amount of precision for a large gain in recall: about 38% pecall for German, compared to a loss of 12% in precision and even better in English.

It is also interesting to oberve that the performance of the global LSV algorithm appears more or less invariant to the corpus size. At the same time the performance of the local LSV is strongly influenced by the corpus size, but only with respect to recall. Precision appears much less affected. This is because once a word is frequent enough, independently of the size of the corpus, the word's evidence allows it to be analyzed correctly. The corpus size almost exclusively determines how many such words there are.

A few final comments on the figures provided are necessary. The precision value of 64.14% for German, for example, appears to be relatively low - meaning that every third morpheme boundary found is wrong. However, this is not the case: An error analysis shows that over 50% of 'errors' according to CELEX were not actual errors, and the majority of the other errors are at least arguable. For example, in most languages gender marking is considered a suffix. In German, due to a lack of neutrum and masculinum suffixes, the femininum termination *-e* is not considered a suffix, disregarding the fact that there are word forms with the same stem and without this element, such as *Schule* and *schulisch*. Consequently, all the femininum termination occurrences identified are marked wrong according to CELEX.

Furthermore, because of CELEX's method of mapping word forms to lemmas, no information is available for the morphological analysis of many compounds and derivates. Thus analyses such as (German) *west-ind-isch*[6], *An-denk-en* or *Super-intendent* are marked wrong despite being correct. Table 5.5 illustrates for several semi-randomly chosen word forms the contrast between their analyses by CELEX and the algorithm, respectively. It also demonstrates the possibility for the occurrence of a large gap between measured and true performance of the algorithm.

Another source of 'errors' are words of foreign origin, particularly latin words in the two evaluated languages. The original latin morphology is not considered valid in the German (or English) morphology as in *Ele-ment* or *Parla-ment*, although the algorithm is able to detect these boundaries.

5.4. Further analyses

Several further avenues can be taken from the current point. These include (as was also discussed at the Morphochallenge 2005) improvements of the morpheme boundary detection and further research on true morpheme retrieval as opposed to mere morph retrieval. The most promising improvements appear to be the following points:

5.4.1. Iterating the morphological analysis

There are hints that iterating the morphological analysis could improve overall results. In the first steps only morpheme boundaries likely to be correct are accepted, i.e. the threshold is set to a high value. Then all word forms in the corpus are split according to these boundaries and the entire algorithm is rerun. The subsequent iterations of the algorithm should be able to profit from two effects. First, for longer word forms the morpheme boundary detection algorithm does not need attempt the correct detection of all boundaries at once. Second, especially when using context representations such as sentence co-occurrences in agglutinative languages (Finnish, Turkish), the amount of information available for each type increases, if in each iteration large types (initially word forms) are split into several smaller types.

From Figure 5.3 it is clear that setting the thresholds to higher values increases precision while decreasing recall. For the example *Kommunikationswissenschaften* (communication science), this means that the highest LSV scores are

[6] In this respect CELEX is inconsistent, because for example *pazif-isch* is correct according to CELEX.

achieved for the two correct boundaries *Kommunikations-wissenschaft-en* and several lower LSV scores can lead to the detection of arguably wrong boundaries such as *Kommunikations-wissenschaf-t-en* if the threshold is lowered. However, after splitting this word into the three initially found parts, the second part *wissenschaft*, for example, can be re-analyzed using a richer contextual representation.

Because nouns are capitalized in German, it will not be the same word type as *Wissenschaft* - as it should be. But firstly, there are many compounds containing this word. And secondly, capitalization can be seen as alternation and the algorithm presented in Section 5.4.5 below reliably detects capitalizations. It could therefore be used to merge *wissenschaft* and *Wissenschaft* into a single word type. This allows to further enrich the contextual representation of *Wissenschaft*. Then, the same high treshold setting can be used again to detect more correct boundaries (*Wiss-en-schaft*). Such iterative re-analyses actually resemble a recursive morpheme ordering, and it is common to treat morphology as a recursive process in theoretical linguistics.

5.4.2. Combination of various methods

The algorithms presented at the MorphoChallenge 2005 differed in two aspects: which methods they employed and which types of mistakes they made. Additionally, a simple voting system combining the best algorithms into a voting system significantly improved the final results of the morpheme boundary detection. These are hints that no single algorithm fully explored all the principles necessary to produce a near-flawless morphological analysis.

As an obvious example, the algorithm presented in this work as well as the algorithm submitted by Bernhard (Bernhard, 2006) appear to be complementary. The LSV based algorithm produces very precise morpheme boundaries for frequent words, therefore covering the majority of irregular words. Typically, frequent words are shorter than less frequent words across all languages, most probably due to the least-effort principle. Because of the sparse contextual information for less frequent words, this algorithm fails to detect any boundaries for those.

On the other hand, the algorithm by Bernhard makes use of contextual information **within** words. The longer a word, the more context can be acquired for any substring of that word. This enables the algorithm to analyze long words very precisely, whereas its performance degradates with shorter, more frequent words. A combination of these two methods with a sophisticated weighting, including length and frequency of a word, appears very promising. Other algorithms could be combined similarly, though their complementarity is less obvious. An example of that is the formalism introduced in the recent versions of the Morfessor algorithm, which seems to be a solid automatic control method for the involved tresholds.

In order to explore the potential gain of combining any methods, it is necessary to analyze which principles are represented by which algorithm. The purpose of the following classification example is to single out general ideas (or principles) characterizing the algorithms. Some of the principles were introduced at the beginning of this chapter, namely the usage of semantic context and the MDL. This classification is used below in Table 5.6 to visualize the interdependencies between the various algorithms.

Morphological typology: There are various types of morphological systems. A crucial aspect of the presented algorithms is whether morphemes concatenate to form word forms or whether other types of composition are used. One extreme is represented by a language such as Finnish, where word forms consist of concatenated chains of morphemes. The other extreme is a language such as Hebrew, where stems are consonants and the places between them are filled with vowels according to the function to be expressed.

In addition to these extremes, languages also exist where each morpheme is represented by a single morph (again, Finnish) or where the mapping between morphs and morphemes is more complex (such as German). Mainly because the related research is conducted primarily on Central-European languages, all algorithms assume simple concatenation and do not make assumptions about the morph-to-morpheme mapping.

Transformation rules: According to recent research (Sproat, 1992), transducers can be used to represent the morphology of any language, except for one feature of a few languages, namely free copying. Thus it is safe to assume that there is a finite set of (finite-state) rules that generate any word form of a given language. This can be exploited in various ways: one possibility is to create explicit hypotheses about signatures (Goldsmith, 2001; Freitag, 2005). Another is to train trie classifiers to implicitly learn signatures by learning regularities, which is done in the second step of the algorithm presented above.

Making use of morpho-tactics (Bernhard, 2006) can be seen as another way to exploit the regularity of morphology. The finite-state rules differentiate between prefixes, stems, suffixes and linking elements, etc. During a morpheme boundary analysis any hypothesized morpheme can be temporarily classified into such a category. Once *-ing* is singled out as a possible morph of English and another algorithm classified it as being a suffix (based on frequent occurrences word endings). This knowledge can be used to decide that the morpheme boundary *ing-rate* is probably wrong.

One of the simplest (yet most successful) usages of hypothesized transformation rules was made by Pitler and Keshava's (2006) algorithm. It assumed that any affix, when stripped off from a word form, leaves another existing word form, i.e. *eating* to *eat*. However, this works well only for a restricted set of languages.

Predictability is closely related to the transformation rules. Many of the algorithms presented at the MorphoChallenge 2005 made use of the following assumption: the less likely a substring Y is to follow another substring X (using a probability model), the more likely there a morpheme boundary between Y and X exists (Bernhard, 2006; Bordag, 2005; Bordag, 2006a; Creutz and Lagus, 2006).

There is another, not entirely equivalent way to express predictability. The greater the variety of substrings observed after X, the higher the probability that there is a morpheme boundary after X (Bordag, 2006a; Hafer and Weiss, 1974; Harris, 1955). The two concepts of conditional probability and variety are related, because the conditional probability $P(Y\|X)$ can only be small(er) $P(X)$, if after X different substrings can follow. However, the two concepts are not equivalent, because a high conditional probability $P(Y\|X)$ can hide the fact, that there is a morpheme boundary between X and Y, which would be obvious if the variety was accounted for.

To give an example, it can be assumed that X=*run* and Y=*ning*, and there are several word forms with their frequencies: *running(95), runs(1), runner(1), runable(1), runabout(1), runoff(1), runnings(1), runned(1).* For *running* this implies a relatively high conditional probability both for $P(ning\|run)$ and $P(ing\|runn)$, effectively preventing the corresponding morpheme boundary to be detected. In comparison, the high variety of different substrings branching off after *run-* strongly indicates a morpheme boundary.

Minimum Description Length: The MDL, as introduced at the beginning of this chapter, seems to represent the restrictions placed upon the morphology by the human brain that actively uses morphology. One such restriction is the principle of least effort: the brain will not store all possible word forms if an easier way to store the same information is possible. MDL is also related to transformation rules, because when creating a finite-state transducer for the morphology of any language, the elegance (in terms of appropriateness and shortness) and size of the rules plays a decisive role.

Semantic context: The usage of co-occurrence statistics and consecutive similarity measures simulating semantic relatedness for morphologic analysis was introduced at the beginning of this chapter. However, there is a difference among the algorithms: the difference between local and global. Jurafsky (2000) computes the globally most similar word pairs and uses differences between the words of the pairs to detect morpheme boundaries. In contrast, the LSV based algorithm looks at the most similar words for a given word and uses this information only for this word. The work of Freitag (2005) can be viewed as an extension of Jurafskys work by combining it with transformation rules, because clusters of similar words are computed and similarities of word forms between clusters are taken to learn signatures.

	Typology			Transf		MDL	Predict		Context	
	cat	tac	alt	sig	smpl		con	var	glob	loc
Morfessor	X					X	X			
Linguistica	X	X		X		X	X			
Bernhard	X	X					X			
Freitag	X			X					X	
Jurafsky	X								X	
HaferW.	X							X		
Bordag	X			X				X		X
Keshava	X				X					

Table 5.6.: Classification of algorithms according to the underlying principles. cat = concatenative morphology, tac = morpho-tactic morpheme classifications, alt = alternations, sig = learning signatures, smpl = learning simple transformation rules, con = conditional probability, var = substring or letter variety, glob = using a global list of most similar words, loc = using a local list of most similar words to input word

Table 5.6 shows an approximation of which algorithms are reformulations of the same aspect and where they differ most. The five primary classes explained above are split further into subclasses if there is reason to believe that the subclasses differ enough for a combination of them to produce improved results. For example, no single algorithm exists using both the global list of semantically most related words and the local list. But it might be the case that especially transformation rules can be learned better from a global list, while morpheme boundaries are better learned by local semantic relatedness (for a given word).

Roughly three distinct groups of algorithms can be observed. The first group consists of Morfessor, Linguistica (Goldsmith, 2001) and Bernhards algorithm. They all make use of conditional probability, primarily in combination with the MDL to control the lexicon and some morpho-tactic restrictions for further improvements. The second group consists of Jurafskys and Freitags algorithms, which attempt to learn morpheme boundaries relying mainly on global semantic relatedness, with a marginal use of the transformation rules hypothesis. The last, and least coherent group, is related through the use of the variety side of the predictability hypothesis. However, each group also contains one member utilizing the transformation rules hypothesis. Other known algorithms (not represented in the table, except for (Keshava and Pitler, 2006)) can basically be considered as less complete implementations of one of these three groups.

Table 5.6 also shows that research on this topic has essentially just begun. Now the challenge is to single out the truly universal principles, further refine the classification of underlying principles and find combinations which produce the most reliable results. Also more research on morpheme detection in languages with non-concatenative morphology is necessary. The first step towards this aim is presented in Section 5.4.5 further below.

5.4.3. Finding transformation rules

Finding morpheme boundaries is an important step towards a complete morphological analysis of a language. Another step is to build transformation rules. While Freitag (2005) attempts to construct transformation rules (primarily to find morpheme boundaries) in one step from similarity clusters, it is also possible to reverse the steps. Once morpheme boundaries are available from any of the algorithms discussed above, it is trivial to find all word forms containing a given morph. However, not all these word forms necessarily belong to the same stem as a derivate, inflected form or a compound. For example the "Span" in *Spanplatte* (chipboard) is not identical with the homonymous morpheme "Span" in *Spanien* (Spain).

Furthermore, word forms are built according to certain principles and not all suffixes can be attached to any stem. For example, the German word forms *um-geb-end* (adv. surrounding) and *Um-geb-ung* (n. surrounding) exist, but simply adding the suffixes of both existing word forms to any one of the two does not produce meaningful words: **um-geb-end-ung* or **um-geb-ung-end*. Since these restrictions (for example inflectional classes) are regular in most languages, it should be possible to induce them. Once the general form of such a rule set is learned (for example as a transformation automaton), it should be possible to fill in missing but meaningful word forms. For example, once the hypothetical algorithm learned the above rules for *Um-geb-ung*, it should be possible to confront it with the word form *Um-kreis-ung* (n. a flight around sth.) in order to produce the unobserved but correct word form *um-kreis-end* (adv. circles around).

As shown by Freitag (2005), contextual information is a good source for such analyses and further research will have to reveal to what extent such automatically created transformation rules and restrictions can compete with manually created ones. Nevertheless, all such rules fall into the classification of syntagmatic and paradigmatic relations as defined in Chapter 2. For example, there is a syntagmatic relation between *geb* and *ung* (derivation) and a paradigmatic relation between *geb* and *kreis* (same word class - verb).

5.4.4. Morpheme co-occurrences and similarity

Once at least a partial morpheme boundary detection is performed on a certain corpus, it becomes possible to observe the typical usage of morphs by means of co-occurrence statistics as demonstrated with word forms in the previous two chapters. Consequently, the same analyses can be performed on them: To find similar morphs, cluster them into classes, find ambiguous morphs etc. Additionally, mistakes can be found based on similarity computations that detect misspellings or alternative spellings (e.g. British English vs. American English). Since this is outside the scope of this work, only a few examples are shown to support these claims.

Disambiguation: For example the morph *ver*, is a German prefix appearing at the beginning of adjectives, verbs and nouns. If it appears as an initial morpheme of a noun, it is capitalized. For this proof-of-concept, the words of 11 million German sentences (part of the 'Projekt Deutscher Wortschatz') were segmented using the algorithm described in this chapter. Then sentence co-occurrences for any type (be it a morph or a word form) were computed and the WSI algorithm (Chapter 4) applied. The result is that the morph *ver* is ambiguous with the following two 'senses':

1. bind, folg, geb, gleich, handl, hind, kauf, lass, lich, lieh, nehm
2. bund, bände, füg, em, end, lang, läng, letz, lust, pflicht, trau

The first sense consists primarily of verb stems (if combined with *ver*) and some other types (suffixes, for example) that typically co-occur with *ver*. The second sense is less uniform and consists primarily of adjective stems and suffixes typical for this prefix. Note that sentence co-occurrences were computed and yet, mostly stems of formerly words with *ver* are computed as significant. Surprisingly, disambiguating the capitalized *Ver* yields two senses as well:

1. bind, brenn, bände, forder, führ, kauf, lauf, leih, lier, schieb
2. anstalter, ation, bands, dienst, eines, fassungs, folg, kehrs, ministerium

The first sense is a mixture of the two senses shown above for *ver*, whereas the second sense is what was expected: mostly stems of nouns (if combined with *Ver*). The mixed first sense is apparently the result of another factor: The first word of any sentence is also capitalized in German. Additionally, there is free word-order in German allowing both adjectives and verbs to appear at the beginning of a sentence as well, in which case they are capitalized.

Similarity: As with co-occurrence statistics, similarity computations can also be applied to morphemes. One application could be to detect mistakes made by the morpheme boundary detection algorithm. For example, the morpheme *deutsch* (German), part of 5 215 other distinct word forms, could have resulted in a number of incorrectly cut morphemes by the morpheme boundary detection algorithm, such as *eutsch* or *eldeutsch* (from *Mitteldeutsch*). However, it still remains the same morpheme, therefore it should be possible to find the correct morpheme among the types most similar to any of the wrong morphemes, as can be observed in the following example, based on the proof-of-concept implementation:

- **deutsch (241306):** hies schaft Deut Europa ansäss der in Eurocop sch ehe europä Handel europa
- **eutsch (134):** rhein Ost west Region süd Insel itali see Süden insel öst ir amerika europa ost Is IRA im land **deutsch**
- **eldeutsch (592):** MDR Rundfunk ostdeutsch funk SFB Berlin Sachsen Anhalt NDR Potsdam Anstalt saar Hörfunk Höppn **deutsch** anstalt

Assuming that the corresponding morpheme boundary detection algorithm had an accuracy of over 50%, in over 2 608 cases the correct morpheme *deutsch* was found correctly. Thus, from the possibilities *deutsch, eutsch* and *eldeutsch*, the correct one should have the highest frequency. This observation could be verified in a few simple experiments, but additional work is necessary, particularly when trying to choose the correct morpheme from all the possible candidates.

5.4.5. Non-concatenative morphology

The most apparent weakness of all discussed morpheme boundary detection algorithms is undoubtedly their inability to handle any form of non-concatenative morphology. These algorithms would almost completely fail to find any morphemes in a language such as Hebrew. However, such morphological systems are not out of reach of the unsupervised approach. To support this claim, a simple algorithm was designed as a proof-of-concept. To further simplify matters, German was taken as the test language. Naturally, the non-concatenative parts of the Hebrew morphology are more complex than in the German parts, yet the basic approach remains the same.

The primary difference between the non-concatenative aspects of Hebrew and German is that in Hebrew, two morphemes are combined into a single word form with the possibility of combining any of the two morphemes with other morphemes. In German only alternation exists (known as allomorphy) - the appearance of a single morpheme is altered and the alternation rules apply only to a certain set of words. Contrarily, in Hebrew it is common that a stem morpheme consisting of three consonants $C_1C_1C_1$ is combined with another morpheme consisting of two vowels V_1V_1 into a word form $C_1V_1C_1V_1C_1$ (Cohen-Sygal and Wintner, 2006). Thus, combining the same stem with a morpheme consisting of two vowels V_2V_2 might result in another word form such as $V_2C_1V_2C_1C_1$ or $C_1V_2C_1V_2C_1$. From the point of view of an algorithm, the difference between the two resulting word forms is rather large: The edit distance amounts to three edit operations out of five letters.

Contrastingly, the non-concatenative aspect of the German morphology is quite simple, because apart from some exceptions at most two letters (and then mostly consecutive) can change between two related word forms, if affixes are disregarded. The purpose remains though - the grammatical specification of the word form is determined by these processes. For example, the lemma */können/* (can) can be expressed e.g. as *kann* (I/he can) or *konnte* (could).

The 11 million sentence subcorpus was used for testing, where the morpheme boundary detection algorithm presented in this chapter was used to segment all word forms into their morphemes. This means that instead of *können, kann, konnte*, this corpus contains *könn, kann, konn, en, te.* The initial step of the algorithm is to compute sentence co-occurrences of the morphs using the log-likelihood significance measure. Subsequently, all morphs are compared to each other morph, using the baseline similarity measure.

The core of the algorithm is to compare each morph against its contextually most similar morph using the edit distance. The algorithm takes the first word which is among the first x (in these experiments $x = 100$) most similar words and has an edit distance lower than some threshold. For the sake of simplicity this threshold was set to 1. The difference between the input word and the word chosen by the edit distance is used to hypothesize a corresponding rule. Thus, for *könn* the first word with edit distance 1 is *kann*, which generates some 'evidence' for a rule *ö-a*.

The core of the algorithm is repeated for each word longer than 2 letters and the frequency of each rule is accumulated. The algorithm ends by discarding all rules having a frequency lower then a threshold f, and printing the remaining rules along with the morph-pairs that generated them. For these experiments, the frequency threshold was set to $f = 25$. The resulting list was evaluated both for correctness of rules extracted and correctness of word pairs that lead to the rules, because for example the morph *und* has the morph *änd* (which is a mistake of the boundary detection) within the 100 contextually most similar words. This pair is among other pairs which generate the otherwise correct rule *u-ä* as exemplified in *wusch* (he washed) and *wäsch* (he washes).

The results of the evaluation of the rules are as following: 33 rules were found, 16 were correct alternation rules. Further 6 rules represent spelling differences of the same words, i.e. *american* vs. *amerikan.* There is also one irrelevant rule *1-*_ meaning that in front of the string 2.000 it is possible to also use the 1 to turn it into 12.000. The remaining 6 wrong rules mostly resulted from mistakes of the boundary detection (_-s) and in a few cases from the pure chance that two contextually similar morphs also have a very similar form. For example the morphs *affen,offen* (monkeys, open) are similar because it is the zoo, which contains monkeys, and the zoo can be open or closed at certain hours. Out of the correct rules, an approximated average of 83% (based on judging the first 20 examples for each rule) examples were correct. The recall of the rules is very good - in fact, not a single rule is missing. The recall of morph-pairs could not be measured, but can be assumed to be very low. This is because for most morphs there is not enough evidence to warrant a reliable ranking of similar words. Raising the recall should be the goal of a more thorough implementation of this algorithm.

Table 5.7 demonstrates the most frequently found rules from the German corpus, including the first few alphabetically sorted examples of morph pairs from which the rule was learned.

Rule	Freq	Examples
a-ä	611	aberglaub,abergläub absatz,absätz abschlag,abschläg acker,äcker
u-ü	275	absturz,abstürz abwurf,abwürf anschluss,anschlüss
_-e	185	ansiedel,ansiedl artikel,artikl astero,astro bahnhofes,bahnhofs
o-ö	179	arbeitungsblock,arbeitungsblöck bahnhof,bahnhöf block,blöck
c-k	172	african,afrikan american,amerikan apocalyp,apokalyp beck,bekk
e-i	142	abgeb,abgib aek,aik agenten,agentin alexej,alexij alpen,alpin
a-e	133	aberkann,aberkenn afg,efg ahmad,ahmed ain,ein alawit,alewit
a-o	101	affen,offen apec,opec assen,ossen barg,borg barst,borst blau,blou
i-o	85	architektin,architekton di.,,do., di.,do. di.:,do.: dich,doch dirb,dorb
a-i	84	aaa,iaa acht,icht achte,ichte ahn,ihn anna,inna anne,inne ans,ins
o-u	71	chong,chung common,commun diktator,diktatur fessor,fessur
e-o	68	abheb,abhob apostel,apostol beat,boat deng,dong derb,dorb
a-u	62	abdallah,abdullah and,und art.,urt. asc,usc band,bund
e-é	52	academi,académi amery,améry atletico,atlético bela,béla cafes,cafés

Table 5.7.: The rules resulting from analysing the 50 000 most frequent words, ordered by the number of morph pairs each rule was learned from.

The algorithm also retrieves the principle of capitalization. The results in Table 5.7 actually contained 26 more rules, all saying that each letter is replaceable by its capitalized version, i.e. *d-D* in *davon - Davon*. Although these rules were removed from the table (and evaluation), they remain correct and can be used as a simple automatic correctness evaluation for unknown languages.

The second set of rules (out of two sets) that was removed from the table (and evaluation) above essentially expresses that any digit can be replaced by any other digit, i.e. *1-2* in *11,12*. In other words, the algorithm successfully learned the free combinability of digits to numbers. This can also be used for a simple assessment of the general quality of the results.

As hypothesized above, similarity can also be used to detect errors of the underlying morpheme boundary detection algorithm. This algorithm essentially finds systematical errors. For example the rule *_-s* was learned from 45 examples, such as *amateur, samateur*. The *s* in front of the *samateur* is a linking element which appears if the morpheme is used within a compound such as *Berufsamateur*. Such examples and rules were counted as wrong in the evaluation.

5.5. Conclusions

This chapter substantiates one of the major claims made by the SIML in Chapter 2: The same principles of composition and abstraction operate on all language levels. Several other hypotheses formulated earlier throughout this work also received strong evidence. One is that combining different algorithms boosts the performance of any participating algorithm. Another is that the same type of algorithms (i.e. similarity computation) can be applied to different language levels in order to obtain the same type of knowledge (i.e. paradigmatic relations between morphs). Hence, in spite of the details discussed in this chapter, it strongly supports the idea of an unsupervised learning system, which is based on only a few universal principles.

One of the primary critiques of the unsupervised approach is that such algorithms are limited in: the ability to produce precise data, and in the ability to learn more complex concepts apart from crude association measures. Contrary to that, this chapter should illustrate that the potential of unsupervised learning methods cannot even be gauged properly. On the flip side, this also means that despite the significant amount of work done, unsupervised learning methods currently stand at the very beginning of their development. In addition, this chapter gives hints for new principles not yet included in the SIML. The fact that knowledge about the usage of units on one language level helps to learn knowledge on the underlying level is not yet modelled in SIML.

With respect to practical applications, a new algorithm for morpheme boundary detection was introduced and successfully evaluated, both with an own evaluation and by means of a participation in MorphoChallenge 2005. Especially the MorphoChallenge evaluation showed that the results compare well with other current algorithms. Furthermore, a prototype of an algorithm was created which reliably finds alternations - enabling the detection of morphemes in non-concatenating languages. Possibilities for further enhancements, such as combining some of the current algorithms were discussed in-depth. However, the overall quality of any algorithm concerned with unsupervised analysis of morphology has an error rate of 30% and above, currently limiting the usefulness for real-world applications.

6. Semantic Relations

The methods described in Chapter 3 allow to partially distinguish syntagmatic and paradigmatic relations between words. The evaluations reveal that both co-occurrence rankings as well as similarity rankings contain samples of all relations, including multi word expressions, synonymy and cohyponymy, but also hierarchical relations such as hyperonymy and meronymy. As in Evert's work (Evert, 2004) on collocations, the results from measuring co-occurrence significance tend to represent syntagmatic relations (idiomatic expressions, typical modifications, multi word expressions). Contrastively, similarity measures yield more paradigmatic relations. In this chapter, this finding is used as a starting point to explore methods for disentangling the various relations. However, only two cues obviously cannot suffice. Therefore one of the purposes of this chapter is to examine further available cues in detail. The aim is to quantify which method yields which types of relations and discover possibilities for new methods. This can help to design algorithms that extract one specific relation.

It should be noted that although this topic can be named *Relation Extraction*, it should not be confused with Relation Extraction in the context of Information Extraction (IE). In IE, particular relations between previously detected entities are extracted, such as *X is a friend of Y*. In this chapter, methods are explored which help to extract abstract, semantic relations between words, such as hyperonymy or cohyponymy.

The procedure in this chapter differs from the one commonly taken in the related research. Usually, the task is to find word pairs standing in a particular relation. A classic example is Hearst's approach (Hearst, 1992), which has remained standard over the years. It is based on a part-of-speech tagger and the identification of so-called lexico-syntactic patterns (also: Hearst Patterns). The results depend on the quality of the corpus, the tagger and the initial training set, but also on the salience of the corresponding relation. Especially the hyperonymy relation in English manifests itself by a set of phrases such as *An X is an Y*, greatly facilitating such approaches. However, no detailed evaluation exists yet, how the named steps of the algorithm contribute to what extent to the final results.

Contrary to Hearst's approach, the one in this chapter is not focussed on describing a program that is able to find a certain relation. Instead, similar to Chapters 3 and 5, this chapter is laid out as a helpful reference for the creation of such algorithms. For example, a particular relation may not be discussed here at all[1].

[1] For example those not fitting into the classification of synonymy, antonymy etc., due to an

Yet, the design of an algorithm extracting such a relation is supported by providing information on the effects the various methods have. For example, if the unknown relation is syntagmatic, it is does not help to attempt finding syntactic patterns appearing between the words standing in the unknown relation. Instead, looking for syntactic constructions that **contain** these words is more helpful.

For these purposes, first an examplary set of possibly desired relations is chosen, using those from WordNet (Miller, 1990; Fellbaum, 1998) and from the Annotation Project (see Appendix B)). Then, for each relation a set of word pairs in this relation is selected from the corresponding gold standard. This data is used for evaluation.

Furthermore, a set of methods is used extracting information about words and relations between them. The set of methods is selectively drawn from literature about the extraction of specific relations from corpora in a semi-supervised or unsupervised manner. The results of each method are examined with particular respect to whether they provide means to partition the space of word pairs according to the chosen relations. For example, it is possible to assume two methods that compute association strength between words. A further assumption is that one method computes relations between hierarchically related words (hyperonyms, meronyms) better, whereas the other method 'prefers' other relations. If enough methods partitioning the space of possible relations along varying paradigms are found, then it might be possible to create an algorithm that actually extracts the given relation by combining these methods in a specific way.

In the described example it would be possible to create an algorithm that to some extent differentiates between hyperonyms (and meronyms) on the one hand and cohyponyms on the other hand. Further differentiation between hyperonyms and meronyms would not be possible though. However, according to related work such as (Hearst, 1992), it probably suffices to add a third method that clusters word pairs based on the syntactic structure commonly observed between them, to further differentiate between hyperonyms and meronyms.

The methods to be examined are roughly categorized into five classes: cues from association strength (co-occurrences) and similarity measures, cues from symmetry of co-occurrence and similarity measures, cues from sentence structure, cues from morphology and finally various clustering methods. Other possible classes, such as the frequently used POS tagging or usage of pre-existing ontologies, are omitted. For POS tagging there are partially successful experiments (Schone and Jurafsky, 2001b; Clark, 2003; Freitag, 2004; Biemann, 2006b), but no accepted, state of the art unsupervised algorithms exist yet. Therefore, only a prototype of the unsupervised part-of-speech inducer and tagger (from Biemann (2006b)) is used on several occasions.

entirely different understanding of semantics. This assumes, though, that even in this different understanding a difference between syntagmatic and paradigmatic relations exists.

6.1. Related work

There is a rich selection of work on the extraction of specific relations. It is primarily concerned with creating or expanding term hierarchies, ontologies, or semantic lexical dictionaries. Especially for ontologies the most sought-for relation is hyperonymy, because it forms the taxonomic backbone. There are also many attempts to extract the hierarchical part-of relation. There are several highly detailed overviews about building ontologies (Biemann, 2005a) or clustering hierarchies (Maedche and Staab, 2004). For the purposes of this chapter, the following is a brief overview focusing on which components of the existing algorithms make use of language-specific knowledge vs. those working in an unsupervised way. Essentially, any of known algorithm consists of a mixture of any of the following methods: Hearst patterns, syntactic selectional preference, measuring co-occurrences, computing similarity, clustering and making use of existing hierarchies.

Hearst Patterns Hearst (1992) describes a method to semi-automatically select a set of characteristic patterns observable between words and their hyperonyms such as *X, Ys and other Zs* or *X is a Z*. The selection is based on the use of a part-of-speech tagger. These patterns can then be used to enlarge an initially tiny set of hyponyms, and were used to extend WordNet. Berland & Charniak (1999) show that this approach can also be used to extract part-of related words. Currently however, this approach is always used in conjunction with a syntactic parser that finds the proper patterns, and it remains unclear to what extent the performance of the method depends on proper parsing.

Syntactic Selectional Preference The majority of existing algorithms depend on a syntactical parser to select specific syntactic constructions, including: predicate-argument structures (Hindle, 1990) (more specifically verb-object relations (Pereira, Tishby, and Lee, 1993)) or conjunctions and appositives (Caraballo, 1999). The selection of constructions in this way (Resnik, 1993) is a very strong feature, because it removes the most irrelevant data while keeping the most relevant. Because parsers of sufficient quality for English are easily accessible, it is one of the most extensively used methods and correspondingly there is a selction of quality toolsets available, such as the Mo'K workbench (Bisson, Nédellec, and Cañamero, 2000) or ASIUM (Faure and Nédellec, 1998). With a near-perfect parser and a sufficiently detailed tagset, manual selection of the right syntactical patterns would suffice to at least outperform any existing approaches for the extraction of the most common relations.

Co-occurrences The co-occurrence significance between two word pairs, either unrestricted or for certain syntactic structures, is used for: the extraction of syntagmatic relations (i.e. idiomatic expressions, MWEs, etc.) (Evert, 2004), as features in clustering algorithms (Biemann, Shin, and Choi, 2004), to illustrate the typical usage of words (Heyer et al., 2001), or as conditional probability for classification algorithms (Riloff and Shepherd, 1997). Furthermore, co-occurrences are used as parts of algorithms which try to simulate paradigmatic relations through comparing typical word co-occurrences. For this purpose, they are represented as a vector-space model (Sahlgren, 2006) or as graphs (Cimiano, Hotho, and Staab, 2004; Widdows, 2003). Co-occurrences are also used in combination with various filters to directly extract information (IE) from texts (Heyer, Quasthoff, and Wolff, 2002).

Similarity Based on pure co-occurrences of words, or on co-occurrences of words after a syntactically motivated selection, similarity measures are employed to compare any type of word contexts. The results can be used either directly or are used as features in further clustering algorithms (Biemann, Bordag, and Quasthoff, 2004). Essentially, any algorithm utilizing co-occurrence data in order to produce clusters or classifications, makes use of one or another similarity measure. Thus (be it directly or indirectly), such second-order comparisons are also frequently used throughout the existing algorithms.

Clustering A consistently recurring idea is that the task of automatic hierarchy creation is primarily a matter of finding the correct set of characteristic features for every word. A good hierarchical clustering algorithm should then be able to trivially produce the desired hierarchy of the words (Bisson, Nédellec, and Cañamero, 2000). However, as numerous attempts illustrate (Caraballo, 1999; Cimiano, Hotho, and Staab, 2004; Pantel, Ravichandran, and Hovy, 2004; Ciaramita et al., 2005), both the selection of features and the choice of the proper clustering algorithm needs to be carried out carefully. The features typically used are the aforementioned co-occurrences, similarity data and syntactic selections, but also existing hierarchies and additional information sources. The choice of the clustering algorithm is more technically influenced, because a common hierarchical agglomerative clustering algorithm is (and will remain) too costly to compute on the typical sizes of data which range well beyond hundreds of thousands of words.

Existing hierarchies Another method in common use begins with an existing ontology or hierarchy and extends it using any number of the methods mentioned above (Ciaramita et al., 2005). This approach is often expressed as a classification task (Roark and Charniak, 1998; Witschel, 2005), where each node of the hierarchy is a possible class to which new words can be added.

Morphology Though only for a restricted set of languages, morphology can sometimes help to find relations between words as well. On the one hand, certain suffixes exist carrying information about the word type (such as *-ing* in *running*), enabling a simple and correct selection of *moving* as a similar word from the retrieved set of hypothetically similar words (*wood, moving, away*). On the other hand, if present in the given language, compounds often carry direct information about a hyperonymy hierarchy (i.e. *Bahnfahrt* - *Fahrt* in German). More examples of this sort exist and typically this knowledge is manually translated into rules in order to utilize it.

6.1.1. Summary

There is also a number of common problems throughout all approaches concerned with the present topic. One is the recurring problem of data sparseness. For many words it is impossible to gather enough information to properly classify them. A popular solution to this problem is the additional acquisition of large amounts of text from the Web (Agirre et al., 2000; Keller, Lapata, and Ourioupina, 2002; Brill, 2003; Cimiano, Hotho, and Staab, 2004), but for very restricted domains or rare languages this does not help in the long run.

Another common problem is the generally low precision of the results, for example 35% – 55% (Caraballo, 1999). But even this performance is only achieved with a combination of all known methods and in restricted areas only (Cimiano, Hotho, and Staab, 2004), such as using an encyclopedia as input (Pereira, Tishby, and Lee, 1993). This indicates that a number of underlying factors have yet to be understood.

Finally, many of the methods, particularly those with higher performance, depend on the existence of a quality parser as well as pre-existing hierarchies. This creates language-dependent solutions which cannot easily be transferred to other languages, regardless of the availability of a parser for these languages. This has lead to the development of several completely unsupervised algorithms, based solely on co-occurrence measures and clustering (Biemann, Bordag, and Quasthoff, 2004).

6.2. Evaluation issues

In the related Chapter 3, primarily GermaNet and WordNet, but also the Annotation Project are used as gold standards for the evaluations. The reasons for this are two-fold: First, GermaNet and WordNet are well known thesauri, especially regarding the quality of the annotations. Second, the purpose of the evaluations in the present chapter is to determine the relative performance of the various measure combinations. Based on these results, one of the best measures each for co-occurrence (log-likelihood) and for similarity computations (baseline) is taken.

For the purposes of this chapter, the Annotation Project is more appropriate as the primary gold standard. Despite its weaknesses (cf. Appendix B), it contains several syntagmatic relations (along with thousands of word pairs annotated for each of these relations), while in GermaNet only paradigmatic relations exist. Furthermore, the Annotation Project was created using the same corpus on which the evaluations in this chapter are run, which guarantees data compatibility (same domain, same genre, etc.).

However, several issues arise with this choice. First, the claimed data compatibility needs to be quantified. Second, the quality of annotations in Annotation Project is essentially unknown - consequently a comparison to GermaNet is needed. Third, it is well known that it is difficult for humans to agree on annotations of semantic relations (Moldovan et al., 2004). Since the goal of this chapter is to explore cues for the automatic extraction of such relations, the inter-annotator agreement between humans gives a good upper bound of achievable - or rather measurable - performance.

The contents of GermaNet and the Annotation Project can be only partially compared. In both thesauri there are relations which have no clear translations in the other thesarus, such as *typical action* in the Annotation Project. The remaining relations can be compared in several ways. One way is to compute the **overlap** (amount of word pairs in the intersection) between the two sets of word pairs for each relation. This overlap can be taken as a simplified inter-annotator agreement. For example, there are approximately $6K$ antonymical word pairs according to GermaNet and $3K$ according to the Annotation project, hence the maximal overlap is $3K$ word pairs.

Table 6.1 illustrates the amount of word pairs for each relation present both in GermaNet and the Annotation Project divided by the smaller of the two sets. Although the Annotation Project is based on word forms (as opposed to GermaNet), the hyperonymy overlap is very high: over 90%. But this might be a skewed result, because there is a significant difference in the amount of annotated pairs. The meronymy relation (i.e. part-of and consists-of relations) provides a more balanced view, because in both thesauri the corresponding word pair sets are approximately equal-sized. The resulting overlap of less than 50%, indicates either the difficulty of the task or that different domains were covered by the annotation (the latter possibility is tested further below). The antonymy relation, despite similar amounts of annotated word pairs for both thesauri, has a much lower overlap of 30%. This indicates, that various types of relations also differ in how easy annotators agree on their annotations. For example, it is easy to generally agree on what a *car* consists of, as opposed to agreeing on what the antonym to *cold* is: *hot*, *warm* or *mild*.

The extremely low overlap for cohyponymy is probably due how cohyponyms are drawn from GermaNet, which lacks an explicit annotation for cohyponymy. To obtain an explicit annotation, all words sharing a common direct hyperonym were assumed to be cohyponyms. This creates one order of magnitude more word pairs for this relation, as compared to the Annotation Project. In the latter, cohyponymy was directly annotated and was not further automatically enlarged. However, due

relation	overlap	MAP(G,A)	MAP(A,G)	pairs(G)	pairs(A)
synonyms	82.15	23.62	84.34	71K	28K
cohyponyms	9.13	1.26	12.54	983K	114K
antonyms	30.83	14.19	48.28	6K	3K
hyponyms	90.81	42.63	93.81	71K	8K
hyperonyms	90.66	34.61	88.53	72K	8K
part-of	48.10	66.35	69.08	7K	8K
consists-of	48.12	60.83	60.79	7K	8K
total	25.78	33.90	61.74	1 217K	156K

Table 6.1.: Inter-annotator agreement between the Annotation Project and GermaNet.

to the set of cohyponyms in the Annotation Project being smaller, the probability to achieve a high overlap ratio should only increase, as observed with the hyperonymy relation. Therefore another factor must be responsible for the low overlap.

Most likely this factor reflects the fact that in GermaNet the relations between the included set of words are (at least supposed to be) completely annotated. If a new word introduced into GermaNet, then all relevant relations to the words already present are added as well. In the Annotation Project the approach was different: here, for a number of input words their co-occurrences and contextually similar words were annotated, without checking possible relations to other existing words. This means that cohyponymy within the Annotation Project is more open and diverse, but it also means that many possible relations are omitted. For cohyponyms this can be an especially strong effect, because although, for example, *car* may be considered a cohyponym of *bike*, the co-occurrences of each respective word will most likely not include the other word. This also demonstrates that different approaches to thesauri annotation can have a strong effect on the resulting content.

Another reason for the low overlap may be that both thesauri describe different parts of world knowledge. It could be the case that words solely from the domain of *cars* are annotated in one thesaurus and solely from *biology* in the other. Then, regardless of a general agreement between the annotators, the overlap would be zero. To test this possibility, additionally the mean average precision described in Section 3.4.6, accumulated for each possible input word of each thesaurus, is given in Table 6.1. This corresponds to the assumption that one of the thesauri is the output of an algorithm, which is evaluated using the other thesaurus. Of course, if several words are annotated as related to some input word in one thesaurus, then treating them as an output of an algorithm introduces a random ranking of the annotated words.

There is a high mean average precision for hyponyms when measuring the GermaNet 'algorithm' against the Annotation Project of $MAP(A, G) = 93.81\%$, but only mediocre ($MAP(G, A) = 42.63\%$) in the reverse case. Two reasons are responsible for this discrepancy. One is the above-mentioned order of magnitude more annotated word pairs in GermaNet, making hits easier to achieve in the much smaller set of hyponym word pairs based on the Annotation Project. Another reason is the (also above-mentioned) fact that GermaNet contains only base forms, which generally renders it impossible to achieve high precision values when evaluating against GermaNet (unless all non-base forms are removed). Especially when considering these two factors, as well as the high precision value, the possibility that the two gold standards describe different parts of the world can be clearly ruled out. For all other relations (except for the meronymy hierarchy), the same pattern repeats itself with different values. The meronymy hierarchy, due to its nearly equal size in both thesauri, illustrates well to which extent the two thesauri are compatible.

Based on these observations, it can be said as a general rule, that half of the thesauri overlaps and another half does not. In the overlapping parts, the inter-annotator agreement can be approximated to about 60%. This approximation is to be treated carefully though, as it is based on comparing annotations of two different experiments.

Baselines

The two baselines used in this chapter are co-occurrences (**justCoocc**) (using the log-likelihood significance measure) and similarity computations (**justSim**) (using the baseline measure) as described in detail in Chapter 3. A comparison of the results divided into individual relations in Tables 6.2 and 6.3 (MAP_a is the mean average precision including all words, whereas MAP includes only words present in the gold standard) shows that the same two baseline algorithms achieve quite different results - all MAP values are half as high when using GermaNet as the gold standard. In both tables, however, the tendencies for each particular relation are the same. Cohyponymy is, for example, the easiest relation to extract using the two baselines, both with respect to the MAP and the pure number of extracted word pairs.

However, measuring mean average precision on all words (MAP_a) considers any output word as a potentially correct hit, even if not contained in the corresponding gold standard. This means that such words automatically increase the miss-rate, even though they might be correct from the point of view of an external observer. In order to quantify the difference, additionally the mean average precision MAP including only words also present in the gold standard is given in Tables 6.2 and 6.3. Most of the figures only increase slightly. However, for GermaNet and for cohyponyms particularly, the performance is nearly doubled - reaching approximately the same level as when measured against the Annotation Project. This is an additional indication that the data produced by the algorithms generally is more compatible to the Annotation Project.

	justCoocc			justSim		
	MAP_a	MAP	num	MAP_a	MAP	num
synonyms	1.20	1.31	4K	4.09	4.30	7K
cohyponyms	4.40	5.92	57K	6.02	6.62	56K
antonyms	0.73	0.91	1K	0.71	0.76	1K
hyponyms	1.22	1.68	6K	5.11	5.44	7K
hyperonyms	0.50	0.65	4K	1.10	1.21	6K
part-of	0.84	1.07	3K	1.15	1.26	3K
consists-of	0.46	0.62	3K	0.63	0.69	3K

Table 6.2.: Measuring the baselines against the Annotation Project, nouns only. MAP_a and MAP in %.

	justCoocc			justSim		
	MAP_a	MAP	num	MAP_a	MAP	num
synonyms	0.38	1.00	2K	3.13	3.46	5K
cohyponyms	1.13	4.98	25K	3.45	6.15	51K
antonyms	0.66	1.24	1K	0.83	0.94	1K
hyponyms	0.72	1.82	6K	3.88	4.57	14K
hyperonyms	0.29	0.89	3K	1.37	1.83	8K
part-of	0.76	1.23	1K	1.18	1.41	2K
consists-of	0.32	0.51	1K	0.56	0.64	1K

Table 6.3.: Measuring the baselines against GermaNet. MAP_a and MAP in %.

However, the most important finding is that regardless of the gold standard, the MAP returns approximatly the same scores for the same relations and the same algorithm. MAP_a differs in this respect significantly, clearly due to items not contained in the gold standard. Therefore, MAP is used throughout the remaining chapter.

Another finding about the nearly identical scores for MAP (irrespective of the gold standard) is that the meaning of the relations must be similar. That means that cohyponymy in the Annotation Project describes the same relation as cohyponymy in GermaNet, despite the low overlap measured above and also despite the slightly artificial method used to obtain cohyponymy from GermaNet.

Overall however, the achieved performance is significantly lower than the approximated upper bound of 60%. This leaves a lot of potential for improvements using further algorithms. Contrary to these baselines, some algorithms cannot produce a stable amount of output words for any given input word, even if its frequency is sufficient. In fact, the goal of some algorithms can be to make a selection of input words. As described below, an algorithm searching for cliques in which a given input word participates can result in there being no such cliques of sufficient size for some input word. This means that the measured mean average precision can only be improved significantly by removing such words. In order to properly compare the rankings produced by such algorithms, a size-corrected baseline measure is necessary, measured with the modified mean average precision introduced in Section 3.4.6.

Originally, each algorithm in this chapter is run and evaluated on the 100 000 most frequent word forms of the german newspaper corpus. Only a portion of these input word forms actually appear in the gold standards: 45 300 in the Annotation Project and 25 475 in GermaNet. Additionally (as described above), it is possible for an algorithm to not yield any results for some words. Therefore in the caption of each results table, the number of input words the measurement is based on, i.e. the **size-correction**, is provided.

In addition to the more objective evaluations against the Annotation Project, a few recurring examples are listed. For each example, the top 5 ranked words according to the corresponding algorithm are given as a subjective measure of the results. The arbitrarily chosen example words are *Elefant* (elephant), *Papier* (paper), *Tschechien* (Czech (Republic)) and *Chopin*. The resulting rankings are not translated, nor are they filtered for words not contained in any of the gold standards.

6.3. Possible cues

With respect to the focus of this work on unsupervised algorithms, there is an essential difference between the various methods used throughout the related literature. Some methods are inherently language-independent, such as measuring co-occurrences and clustering, whereas other methods make extensive use of language specific knowledge, such as syntactic selectional preference. The classification helps to determine which underlying dimensions and principles are highlighted by the corresponding method. For example, measuring co-occurrences emphasizes a different aspect of language compared to the extraction of typical sentence structures or comparing the morphological structure of words. In this chapter, all language-independent methods are explored with respect to whether they can be used as cues to extract a certain semantic relation.

6.3.1. Possible cues from co-occurrence and similarity measures

One aim of this chapter is to quantify the possible benefit of combining co-occurrence and similarity measures in various ways. In order to elucidate the effects of the various combinations, the relations in the Annotation Project (as the primary evaluation gold standard) are classified again into the summarizing classes as described in Chapter 3, namely syntagmatic, symmetrical paradigmatic, hierarchical paradigmatic, derivates, other, total.

The question to be investigated is, do any of the methods (see below) show preference for any of these classes of relations (or perhaps even a particular relation), when compared to the other classes? According to the definitions in Chapter 2, pure co-occurrence measures should be better at computing syntagmatic relations, while similarity measures should primarily compute paradigmatic relations. However, various factors can cause that a word pair standing in a purely syntagmatic relation to behave essentially like a paradigmatic one and vice versa. For example, both parts of a proper name such as *New York* occur separately only slightly more often compared to the frequency of the entire expression (in a german corpus). Thus, the distributed similarity of both words is nearly maximized, because the contexts are mostly the same. Nevertheless this remains a syntagmatic relation.

The five methods to be compared are the following:

Baseline Co-occurrence measure (**justCoocc**): This method is exactly the same as described extensively in Chapter 3. The chosen significance measure is log-likelihood. The definition of statistical syntagmatic relation in Section 2.2 implies that this reflects syntagmatic relations.

Baseline Similarity measure (**justSim**): Based on the results of the co-occurrence measure, this method, as described in Chapter 3, compares the typical context of each word. The definition of statistical syntagmatic relation in Section 2.2 implies that this resembles paradigmatic relations.

Co-occurrence measure minus Similarity (**Coocc**): This method reflects the intuition that MWEs consist of words which co-occur frequently with each other due to the MWE, but are not semantically similar to each other (see also (Biemann, Bordag, and Quasthoff, 2004)). An example is *trojan horse* - both words co-occur frequently, but the literal meaning of *horse* is not related to *trojan* in any way. The computation of the ranking of the most related words according to this method has two steps. First, all co-occurring and/or similar words are distributed in a normalized two-dimensional space. One dimension is co-occurrence strength, while the other dimension is contextual similarity. Second, all words are reranked based their distance from the point of maximal co-occurrence in this space. Thus, a word like *trojan*, co-occurring frequently with the input word *horse*, yet having no contextual similarity with it at all, has a distance of near-zero and thus receives the highest rank. On the other hand, the word *rider*, also co-occurring with *horse* very frequently, but also being contextually very similar, has a large distance of 1, thus receiving a low rank.

Similarity measure minus co-occurrence (**Sim**): This method is the counterpart to the previous one. It reflects the intuition that words which co-occur infrequently, yet are contextually similar, stand in hierarchical paradigmatic relations, such as hyperonymy. This is because whenever a word such as *horse* is used, there is a small chance that an explanation is added that it is an *animal.* But the contexts of both *animal* and *horse* overlap to a larger extent, because both animals and horses need food, are living entities etc. It is computed in a fashion similar to that of co-occurrence minus similarity, by creating the two-dimensional space and reranking based on the distance from the optimal point.

Co-occurrence measure combined with similarity (**CooccSim**): This method represents the idea that especially cohyponyms can be expected to both co-occur frequently and have a high contextual similarity. It is also computed in a manner similar to that of the previous methods.

Using only the two initial dimensions of co-occurrence and similarity, it is impossible to create any more meaningful combinations. However, more dimensions could be added, including the distance in which the words co-occur within sentences or the variance of that distance (Büchler, 2006). It is also possible to measure the degree of symmetry for any ranking: If word B is the most similar word to A, and A is the most similar word to B, then this relation is completely symmetrical. It would be expected that only symmetrical relations such as cohyponymy or synonymy produce mostly symmetric rankings, but see also below Section 6.3.2.

The experimental setup for comparing the methods is as follows. For the 100 000 most frequent words (except those not in the Annotation Project), the 100 most relevant (according to the respective ranking and again minus those not in the

Annotation Project) words are computed by each method. The evaluation of each measure employs the MAP to measure the output against two gold standards, the Annotation Project (see Table 6.4) and GermaNet (see Table 6.5). In addition, the total number of found relations (out of all possible relations from the Annotation Project) is provided for comparison. The MAP helps to detect which relation is preferred by which method, despite a strong imbalance between the total number of words standing in a given relation. Hence, although all methods find word pairs that are predominantly cohyponyms, the MAP enables to distinguish which method is better at extracting hyperonyms.

meth.	justCoocc		coocc		justSim		sim		cooccSim	
	MAP	num	MAP	num	MAP	num	MAP	num	MAP	num
synt	8.01	39K	3.62	28K	3.83	24K	1.78	18K	5.87	33K
para	8.18	71K	2.02	32K	12.07	75K	7.70	67K	11.69	81K
h. para	3.56	18K	0.85	9K	7.81	26K	5.89	23K	6.61	27K
deriv	0.22	1K	0.04	1K	2.05	3K	1.98	3K	1.05	3K
other	2.58	8K	0.66	4K	2.12	7K	1.02	6K	2.62	8K
total	16.07	133K	4.75	73K	18.48	131K	11.66	113K	19.12	147K

Table 6.4.: Measuring the top 100 ranked results for all five methods against the Annotation Project. Out of the 100 000 most frequent words only 45 300 were in the gold standard, thus influencing the mean average precision. *MAP* in %.

The first Table (6.4) allows for several observations. The methods *justsim* (plain ranking of most similar words) and *cooccSim* (additionally how often the two words co-occur) have the best overall results. The difference between these two methods is marginal. However, it is also obvious that all these methods offer different mixtures with respect to the types of relations they extract with higher positions in the rankings.

Paradigmatic relations : It appears that *justSim* is the best method to achieve the highest rankings for paradigmatic relations, especially because it differentiates so sharply between paradigmatic and syntagmatic relations. On the other hand, *cooccSim* has the largest distance between symmetrical paradigmatic and hierarchical paradigmatic relations. In addition to that, Table 6.4 shows that *cooccSim* actually has the highest recall.

Assuming that hierarchical paradigmatic relations should be characterized by low co-occurrence, the *sim* method should extract these above all others. However, even this method ranks symmetrical paradigmatic relations higher than hierarchical, although only marginally. Thus, the assumption proved correct, but with a very weak effect. Additionally, the results can be interpreted such that *justSim* is the method of choice when trying to compute hierarchical paradigmatic relations, because it achieves higher scores for them. However, the relative ranking against the other relations is better for *sim*. The lowever numbers could also result from the same bias observed in Section 3.4. This bias resulted in plain co-occurrence measures to achieve comparable results to similarity measures for the Annotation Project. This is because the Annotation Project is based on co-occurrence data as hints for the annotators. Nevertheless, attempts to construct a corresponding algorithm extracting either symmetrical paradigmatic or hierarchical paradigmatic relations should take the distinction between *justSim*, *cooccSim* and *sim* into account, but they should not rely on them exclusively.

Interestingly, both *cooccSim* and *justSim* also achieve rather high MAP values for syntagmatic relations. Syntagmatic relations often hold between words of different word classes. Therefore, a part-of-speech based filter should help by removing words with a differing word class. In fact, when using Biemann's prototypical unsupervised part-of-speech tagger, for example for *cooccSim*, the MAP for syntagmatic relations drops to 2.56 (from 5.87), whereas the paradigmatic (from 11.69 to 10.80) and hierarchical paradigmatic relations (from 6.41 to 6.61) remain stable.

Syntagmatic relations : As described above, the best method to extract syntagmatic relations should be *coocc*. It ranks those words highest, that co-occur often, but have low contextual similarity. Yet, this method does not produce the highest mean average precision for syntagmatic relations. The simpler *justCoocc*, which does not account for similarity, achieves more than twice as good rankings for syntagmatic relations. However, *coocc* clearly prefers syntagmatic relations, because it ranks them significantly higher than any other type of relation. This indicates that this method can be used directly to create an algorithm which extracts syntagmatic relations.

The reason why *cooccSim* achieves such good results for syntagmatic relations is because the majority of relations that were subsumed under the label syntagmatic were semantically compositional relations. For example the relation "typical location", while clearly syntagmatic, contains word pairs in which both words are also frequenty semantically related: examplified by the word pair *(Christ, Kirche)* (christian, church). Another issue derives from contextual similarity through too high co-occurrence counts, such as with *New York*: If the frequency of the non-compositional MWE is sufficiently high, then it automatically also raises the contextual similarity. It may be helpful to recompute the contextual similarity without the co-occurrences from the specific word pair. The words of the word pair could then be verified as really similar to each other, if after this operation they are still found to be similar.

As an example, all sentences in which both *elephant* and *giraffe* occur are removed from a corpus. Many sentences should remain where *elephant* co-occurs with *animal, africa, zoo, ...* and other sentences where *giraffe* co-occurs mostly with the same words. Hence, the contextual similarity would still result in a high similarity between *elephant* and *giraffe* and would be verified as not merely a result of an overly frequent MWE.

For *New York* this verification should fail, because no specific reasons exist as to why *New* should co-occur with anything that *York* also co-occurs with, once all sentences are removed that contain both words. In cases such as *Sri Lanka*, it is likely that no sentences are left after removing all sentences containing both words. This would represent the most extreme case of a verification failure. Another possibility to disentangle MWEs from other relations could be to utilize the distance between the occurrences of both words, in addition to the overall variance of that distance.

Finally, Table 6.4 gives evidence that using *justSim* as a simulation of morphological relatedness (as in the previous Chapter 5) was the best choice. This is because *justSim* extracts the most derivationally related word pairs and additionally ranks them highest, in comparison to the other methods.

Nevertheless, the achieved MAP values remain very low. Considering that for 45 300 input words, approximately 147 000 relevant output words were found, for each input word three output words were relevant. If, for example, the first relevant is at position 4, the second at position 6 and the third at position 7, then a MAP of 19% (similar to the total MAP of 19.12% of the *cooccSim* method) is achieved. That assumes that evaluations against a gold standard like the Annotation Project are trustworthy enough to derive absolute values. However, due to the incompleteness of the gold standard, it is highly probable that the real performance of the algorithm is slightly (or a lot) better. The sole effect really observable from the evaluation of the new methods (*coocc*, *sim* and *cooccSim*), as compared against the baselines (*justCoocc* and *justSim*), is a shift of preferences between various relations. While this in itself is a success, these methods can probably be used only as parts of other algorithms, such as clustering algorithms or lexicographers toolsets.

Another gold standard: GermaNet Using an alternative gold standard for a comparative evaluation enables the detection of specific properties of the gold standards. GermaNet and the Annotation Project differ in nearly every possible aspect, including: how relations are annotated, whether lemmatized words are used, as well as how elaborate and deep the hierarchy is, etc. It is encouraging that the majority of observations previously made can be verified when using GermaNet. Some observations, particularly those concerning syntagmatic relations, cannot be verified because no syntagmatic relations exist in GermaNet. However, the evaluation against GermaNet offers additional insights, especially regarding hierarchical and symmetrical paradigmatic relations.

The *sim* method comes closest to the goal of ranking hierarchical relations higher

meth.	justCoocc		coocc		justSim		sim		cooccSim	
	MAP	num	MAP	num	MAP	num	MAP	num	MAP	num
synset	1.00	2K	0.25	1K	3.46	5K	3.31	5K	2.67	5K
cohyp	4.98	25K	2.85	10K	6.15	51K	5.55	50K	5.75	48K
antonym	1.24	1K	0.12	1K	0.94	1K	0.40	1K	1.15	1K
hyponym	1.82	3K	0.76	3K	4.57	14K	4.20	13K	3.96	13K
hyperon	0.89	6K	0.58	2K	1.83	8K	1.80	8K	1.56	7K
consists	0.51	1K	0.15	1K	0.64	1K	0.53	1K	0.59	1K
part-of	1.23	1K	0.12	1K	1.41	2K	0.97	2K	1.40	2K
total	7.85	27K	4.47	9k	10.72	57k	9.74	56k	9.87	57k

Table 6.5.: Measuring the top 100 ranked results for all five methods against GermaNet. Out of the 100 000 most frequent words, only 25 475 were in the gold standard and thus were used for the mean average precision. MAP in %.

than symmetrical relations. But once again, the effect is too weak to be of any direct use.

The achieved mean average precision, in the case of counting any relation as relevant (total), is lower for GermaNet as compared to the Annotation Project. This is partly due to a smaller variety of relations in GermaNet. The lower values stem from the fact that the Annotation project was created with the same corpus to which the algorithms were applied, while the creation of GermaNet was primarily not corpus-driven. Another difference is that in the Annotation Project word forms were annotated, whereas in GermaNet, only base forms as representations of lemmas were used.

The examples in Table 6.6 demonstrate some of the observed effects. *cooccSim* contains two words (*Porzellanladen* and *Mücke*) as parts of idiomatic expressions, while *coocc* contains four words which are parts of MWEs. However, *coocc* is dominated by both very frequent and very infrequent words, indicating that this method could benefit from a stop words list. Alternatively, a frequency rating could be included to increase the usefullness of this method as a collocation extraction tool. However, stop words may also be parts of MWEs and in this particular example it is the idiomatic expression *Sich wie **ein Elefant** in einem **Porzellanladen** zu benehmen - to behave like an elefant in a china shop* (i.e. to be very rude or ignorant).

As clearly observed in the evaluation results, the difference between *justSim* and *cooccSim* is minimal. It is noteworthy that in the examples the difference between *sim* and *cooccSim* is that in the results of the former for *Tschechien* there are four variants of the input word (i.e. synonyms or parts thereof) and the latter contains only true cohyponyms. This is in accordance with the hypothesis that *sim*

justCoocc	
Elefant	Porzellanladen, ein, Löwe, Mücke, Nashorn
Papier	auf, dem, das, Blatt, Euro
Tschechien	Polen, Ungarn, Slowakei, in, Slowenien
Chopin	Beethoven, Werke, Liszt, Mozart, Schumann
coocc	
Elefant	Porzellanladen, ein, Mücke, Weißer, Maus
Papier	dem, das, Das, heißt, und
Tschechien	in, und, gegen, Jana, aus
Chopin	von, und, Uhr, George, Sand, Charlotte
justSim	
Elefant	Elefanten, Löwe, Affe, Tier, Nashorn
Papier	Pappe, Metall, Kunststoff, Kunststoffe, Karton
Tschechien	Ungarn, Polen, Tschechische, Republik, Finnland
Chopin	Beethoven, Debussy, Schubert, Liszt, Brahms
sim	
Elefant	Elefanten, Hund, Löwe, Affe, Hase
Papier	Metall, Aluminium, Plastik, Behälter, Altpapier
Tschechien	Tschechische, Republik, Tschechischen, CSFR, Schweden
Chopin	Vivaldi, Hindemith, Britten, Händel, Violine
cooccSim	
Elefant	Löwe, Elefanten, Nashorn, Affe, Tier
Papier	Pappe, Glas, Kunststoff, Karton, Metall
Tschechien	Ungarn, Polen, Slowakei, Slowenien, Österreich
Chopin	Beethoven, Liszt, Debussy, Schubert, Mozart

Table 6.6.: Top 5 ranked results for the four examples of each method.

computes more semantically similar words (i.e. true synonymy), whereas *cooccSim* is better at computing cohyponyms.

6.3.2. The effect of symmetry

Apart from combining co-occurrence and similarity measures, it is also possible to consider the symmetry of a ranking (Weeds and Weir, 2005). For example, if the most similar word to *horse* is *animal*, the reverse case (that *horse* is the most similar word to *animal*) may not be true. It could be expected that symmetrical paradigmatic relations tend to have symmetrical rankings, i.e. if *elephant* is one of the most similar words to *giraffe*, so should *giraffe* be one of the most similar words to *elephant*. At the same time, hierarchical paradigmatic relations should be unsymmetrical.

These hypotheses were tested on co-occurrence and similarity rankings. For each word (100 000 most frequent ones) of the test corpus, the most significantly co-occurring and the most similar words were acquired (i.e. the same as *justCoocc* and *justSim* in the previous subsection). Then the word with the highest ranking (for example the most similar) was assigned the highest value of 1.0, and all other words were distributed into lower values according to their similarity. The similarity (or co-occurrence significance) was scaled down logarithmically. For example, the most similar words (with their similarity values) for *elephant* are *giraffe (52), elephants (48), bison (34), antilope (24).* Instead of the direct similarity values *sim* the new values are then $log(sim)/log(maxSim)$, resulting in the following values: *giraffe (1.0), elephants (0.98), bison (0.89), antilope (0.8).* The logarithm is applied to the values to linearize the power-law distribution of the values.

From the initial rankings of *justCoocc* and *justSim*, there are (at least) two possibilities for incorporating symmetry: symmetrical (*sym*) and unsymmetrical (*unsym*). If the input word is A and one of the output words is B and the relevance/similarity value of B for A is $rel_A(B)$, then:

- **symmetric** : The symmetric relevance $sym_A(B)$ is the harmonic mean of the relevance of A for B ($rel_B(A)$) and B for A ($rel_A(B)$): $sym_A(B) = \frac{2 \times rel_A(B) \times rel_B(A)}{rel_A(B) + rel_B(A)}$.

- **unsymmetric** : The unsymmetric relevance $unsym_A(B)$ is the harmonic mean of the inverse relevance of A for B $(1 - rel_B(A))$ and the relevance of B for A ($rel_A(B)$): $unsym_A(B) = \frac{2 \times rel_A(B) \times (1 - rel_B(A))}{rel_A(B) + (1 - rel_B(A))}$

Note that it is two orders of magnitudes more costly to acquire symmetry based rankings than the previous trivial combinations of the two rankings (co-occurrence and similarity). This is because the ranking of each (i.e. in these experiments 100) relevant word has to be retrieved and transformed into the relevance values.

The two ways to combine relevance values result in four new "methods":

- **symCoocc**: Symmetrical co-occurrences reflect the assumption that if word B is the most significant co-occurrence of A and vice versa, then it is more likely a symmetrical paradigmatic relation, as opposed to any other.

- **unsymCoocc**: On the other hand, if word B is the most significant co-occurrence of A, but not reversely, then the relation is more likely syntagmatic.

- **symSim**: The symmetrical similarity values should prefer exclusively symmetrical paradigmatic relations.

- **unsymSim**: Alternatively, the unsymmetrical similarity should prefer hierarchical paradigmatic relations.

Table 6.7 compares the results of the two variants of combining co-occurrence rankings. The two relevant methods (*justCoocc* and *coocc*) from the previous experiments are included as well. One observation is, that the *justCoocc* results resemble

meth.	justCoocc	coocc	symCoocc	unsymCoocc
synt	8.01	3.62	8.15	2.70
para	8.18	2.02	9.78	1.50
h. para	3.56	0.85	3.43	1.62
deriv	0.22	0.04	0.24	0.08
other	2.58	0.66	2.61	0.76
total	16.07	4.75	17.26	4.61

Table 6.7.: Measuring MAP in % for the top 100 ranked results for each input word against the Annotation Project for *symCoocc* and *unsymCoocc*, and a baseline comparison. (Restricted to the 45 300 words in the gold standard.)

those of *symCoocc*, whereas *coocc* is very similar to *unsymCoocc*. This is mainly the result of the co-occurrence-equals-similarity effect desribed above.

Another noteworthy observation is that *unsymCoocc* ranks hierarchical paradigmatic relations higher than *coocc*, yet ranks all other relations lower or approximately equal. Thus, *unsymCoocc* is useful as another cue for hierarchical paradigmatic relations.

Table 6.8 summarizes the results for the two variants of combining similarity rankings. Only the *cooccSim* method is included for reference. As expected, the symmetric similarity *symSim* shows a clear preference for symmetrical paradigmatic relations. In fact, it scores even better than both *cooccSim* and *justSim* at distinguishing them from both syntagmatic and hierarchical paradigmatic relations. It can be concluded that *symSim* is a strong cue particularly for symmetrical paradigmatic relations.

UnsymSim is also the first method that really ranks hierarchical relations higher than the symmetrical paradigmatic relations. A manual inspection of the *unsymSim* result sets made it apparent that a strong frequency bias is present. Table 6.8 additionally provides MAP computed only for the first half of the test words (the frequent words), and then for the second (infrequent words), independently. The results show that the strong preference for hierarchical paradigmatic relations disappears in the first half and is reinforced in the second. The reason for this is complex. First, it is possible that abstract words have a rather high frequency. Thus, within the first 50000 words, there are significantly more words which are higher up in any hierarchy than within the second half. Second, a hierarchy typically takes the form of a pyramidal structure in which any level has less words than the next higher level and more words in the next lower level.

The following example illustrates the effect this has on the perceived performance of an algorithm: The input set consists of 12 words: one is *animal* and the other 11 are specific animals such as *elephant*. *Animal* is in the group of the most frequent

meth.	cooccSim	symSim	unsymSim	unsymSim 0 - 50K	unsymSim 50K - 100K
synt	5.87	3.14	2.34	1.50	3.61
para	11.69	11.45	4.24	3.54	5.29
h. para	6.61	5.49	4.35	2.84	6.64
deriv	1.05	1.90	0.38	0.40	0.35
other	2.62	1.75	1.08	0.68	1.70
total	19.12	15.57	8.35	5.58	12.56

Table 6.8.: Measuring MAP in % for the top 100 ranked results for each input word against the Annotation Project for *symSim* and *unsymSim*. Additionally a comparison to the baselines and to one of the methods applied only to the first half of the 100 000 most frequent words (Restricted to the 45 300 words in the gold standard.). Additional measurements for *unsymSim*, split into the first 50 000 and last 50 000 words, ranked by frequency.

50 000 words, along with 5 of the other animals. The remaining 6 words belong to the other group of less frequent words. Let further be assumed that the algorithm computes *animal* as the fifth most relevant word for each of the specific animals. For *animal* itself it computes each of the eleven animal words at ranks such as *3,6,9, ..., 33*. The resulting mean average precision for the lower-frequency group is higher, because in the higher-frequency group *animal* causes a drop in the average.

If this method is to be used as a cue to hierarchical paradigmatic relations, then it is advisable to weight its results according to the frequency of the respective input word: The lower the frequency, the more reliable it is, but only up to the point where not enough information is available to compute similar words.

symCoocc	
Elefant	Porzellanladen, Mücke, Löwe, Weißer, Nashorn
Papier	Pappe, Blatt, Glas, Karton, Bleistift
Tschechien	Polen, Ungarn, Slowakei, Slowenien, Jana Novotna
Chopin	Liszt, Beethoven, Schumann, Klavierabend, Werke
unsymSim	
Elefant	Tier, Hund, Mensch, Käfig, Affen
Papier	Kurs, Euro, Textilien, Tonne, gelben
Tschechien	Titelverteidiger, Ländern, Viertelfinale, Jugoslawien, Halbfinale
Chopin	Konzert, Solisten, Orchester, Philharmonie, Mozarts

Table 6.9.: Top 5 ranked results for the four examples using *symCoocc* and *unsymSim*.

Table 6.9 shows the effect of the methods *symCoocc* and *unsymSim* on the example words. The first method surprisingly combines both parts of MWEs and cohyponyms. This is because apart from cohyponyms, 'balanced' MWEs also result in a symmetrical co-occurrence ranking. MWEs are balanced if all parts of the MWE are used frequently in their MWE meaning. For example, in *New York*, *New* is used in many other contexts as well without *York*, whereas *York* is used almost exclusively in this MWE. Hence, this MWE is not balanced. For example the MWE *Sich wie ein Elefant in einem Porzellanladen benehmen* is one of the dominating usages of *Elefant* and *Porzellanladen*.

The examples of *unsymSim* demonstrate the difficulty of defining and/or extracting the correct hyperonym. In a certain sense, for three out of four input words (*Elefant*, *Tschechien* and *Chopin*) the method found something that could be considered a correct hyperonym at the first position. However, only in the case of *Elefant* is this undisputably correct. For the other two words better choices exist at the second positions. Nevertheless, it is quite possible to create semantically sound sentences of the type *X is a Y*, where X is the input word and Y is the top-ranked word according to *unsymSim*. At least the predicted preference for hierarchical paradigmatic relations computed by *unsymSim* cannot be refuted.

6.3.3. Cues from sentence structure

Most of the work regarding semi-supervised extraction of particular relations (also known as ontology learning, or lexical acquisition) uses a method commonly referred to as **sentence structure**. As described in Section 6.1, the goal is to enlarge a small set of word pairs that stand in a certain relation, e.g. hyperonymy. The key elements of the algorithms can be summarized as follows:

- **training set**: A manually created (and therefore highly precise) set of word pairs standing in a particular relation is given: for example the pairs *elephant - animal* and *horse - animal*.
- **correct part of speech**: A tagger ensures that any two words of a newly found word pair match with respect to their part-of-speech tags (i.e. a hyperonym of a noun can only be a noun again).
- **phrases**: A tagger is used to extract typical phrases where the known word pairs occur, for example *A is a B*.

Currently, all three elements are obviously dependent on explicit language-specific knowledge. The question to be answered in this section is: is it possible to construct a baseline for an unsupervised sentence structure based algorithm? For this purpose, three key elements need to be reformulated, so as not to utilize language-specific knowledge:

- **training set**: The set of known word pairs standing in a given relation can be replaced with the result of any algorithm from the two preceding sections.

The drawback is that mistakes of these algorithms could produce further mistakes by the enriching algorithm. Additionally, none of the aforementioned algorithms extracts only one particular relation. Instead, they only show relative ranking preferences for certain relations. However, utilizing sentence structure, it could be possible to narrow down the amount of various extracted relations.

- **correct part of speech**: Despite some attempts at creating unsupervised part-of-speech taggers (Clark, 2003; Freitag, 2004; Biemann, 2006b), they have yet to reach a state which would make it possible to take a state of the art implementation and employ it for testing purposes. The results of the baseline-algorithm thus lack part-of-speech matching and therefore may contain additional errors.
- **phrases**: The absence of an unsupervised tagger (or parser) for phrases is more intricate. It renders the extraction of real sentence structure (such as phrases), where known word pairs occur, impossible. The question then arises: how to define or recognize the "sentence structure" to be learned? Given that information about the real sentence structure is not available: is it possible to resort to known information, such as frequency and typical co-occurrences of the words of any given sentence? To explore this question, various ways to extract **sentence signatures** resembling sentential structure are defined. One example is to remove all words except the most frequent ones between the two input words. For hyperonyms that could produce the correct (in the sense of useful) structure *A is a B*.

Because the latter two parts of the algorithm are either missing, or can be simulated only by simple mechanisms, the corresponding experiments can be viewed as baselines. However, as soon as the aforementioned unsupervised POS taggers, or even an unsupervised phrase recognizer, become available, the performance of these kinds of algorithms should rise significantly.

For the experiments in this section, Table 6.4 is useful to estimate the possible gain of using signatures. According to this table the preferences for the various relations differ: The total numbers of correctly found word pairs is strongly dominated by symmetrical paradigmatic relations, most notably cohyponymy and synonymy. If sentence structure (with or without taggers) is to be learned from this data, a similar prevalence can be predicted. Another difference with respect to the original algorithms (i.e. the ones described in the literature) is important as well: Whereas a small (i.e. low recall) but highly precise training set was used to improve recall, the situation with using the output of other algorithms is different: only data with low precision and a mediocre recall is available.

Several experiments were run using various methods to simulate learning from signatures. The experiments were run on two algorithms from the previous two sections: *justSim* due to it having the highest preference for symmetrical paradigmatic relations and *unsymSim* due to it having the highest preference for hierarchical paradigmatic relations. The learning proceeded as follows:

1. Input is a set of sentences S, a set of word pairs P, and a method M to construct signatures $sign(s, A, B)$, given a sentence $s \in S$ and a word pair (A, B)
2. For each sentence $s \in S$
   For each possible pair of words (A, B) of sentence s
   a) if $(A, B) \in P$, construct a signature $sign(s, A, B)$ using M
   b) store $sign(s, A, B)$ and increase its count if it has been already observed in an earlier sentence
3. For each sentence $s \in S$
   For each possible pair of words (A, B) of sentence s
   a) construct a signature $sign(s, A, B)$ using M
   b) if signature $sign(s, A, B)$ has a count greater than a given threshold t_1
      i. increase the count of the pair (A, B)
4. write out all word pairs whose count is higher than a given threshold t_2, ranked by their count

This learning algorithm crucially depends on how the signatures are constructed. For example, given the sentence *An elephant is a large animal.*, it would be intuitive to take *is a* as the signature and add A and B as anchors to form the final structure *A is a B*. Therefore, three strategies were tested, which were slight variations of the ad-hoc version of the *elephant-animal* example. They have in common that a distinction is made between frequent words (the 1000 most frequent words) and other words (bluntly called **content words**).

First, every method replaces the first input word with A and the second with B. Then in the **Method 1** all content words are replaced with the single character C, each. Second, from each of the two input words, the first non-content words to the left and to the right are searched. Anything beyond the first non-content word is replaced by an asterisk. For example, if the two input words are *elephant* and *animal* and the sentence is *The Indian elephant eventually gained a higher status than the horse, which was an extremely important animal in Indian culture.*, then the resulting extracted structure is *The C A C C a * an C C B in **. The two other methods differ in the following details:

- **Method 2**: This differs from Method 1 in that all lower-case words are replaced with c instead of C. For the given example sentence, the result is *The C A c c a * an c c B in **

- **Method 3**: This method differs from Method 1 in that everything is retained (instead of replacing with an asterisk) between the input words A and B. For the given example sentence, the result is _The C A C C a C C than the C , which was an C C B in *_.

It would be possible to create more variations, but these three variants suffice to measure the influence of more information about the words (differentiating between capitalized or not), or finer grained (but possibly less relevant and sparser) sentence structure (Method 3). Rules having A and B directly next to each other are omitted, because these probably do not indicate paradigmatic relations very well. However, this could be language-specific.

method	syn	para	hier	total	measured	out/word
justSim_siz2	3.09	16.85	7.13	26.45	21 613	3.6
cooccSim_siz2	4.70	15.88	5.86	25.36	23 030	4.2
unsymSim_siz2	1.30	3.67	2.60	7.16	20 289	3.0
justSim_1	1.53	10.35	1.96	12.77	27 941	6.3
cooccSim_1	1.74	10.00	1.75	12.59	24 964	4.8
unsymSim_1	1.57	10.02	1.66	12.59	21 568	3.6
justSim_2	1.61	12.17	1.85	14.78	22 147	3.6
cooccSim_2	1.76	10.85	1.83	13.54	23 803	4.2
unsymSim_2	1.56	11.10	1.74	13.74	20 255	3.0
justSim_3	1.78	5.74	1.44	7.76	37 193	14.2
cooccSim_3	1.78	5.53	1.45	7.57	37 553	14.6
unsymSim_3	1.82	5.46	1.41	7.57	36 953	13.7

Table 6.10.: Performance of learning from signatures, compared to the previous algorithms with adjusted output sizes. MAP in %.

When measuring the performance, there is one particular difference between learning new word pairs from signatures and the algorithms in the previous two sections. Learning from signatures does not produce a guaranteed amount of result words for any input word, even if the input word is sufficiently frequent. Therefore, as described in Section 6.2, a size-corrected baseline comparison is also given in Table 6.10. Since each rule-variant generates varying amounts of output, the size-correction is given only for the best-performing rule (Method 2).

Table 6.10 clearly shows that for all rules and for all initial input sets, only one type of relations is highly favored: the symmetrical paradigmatic relations, in particular the cohyponymy relation. The rule differentiating between lower-case and upper-case words (Method 2) performs significantly better with respect to precision and only slightly worse with respect to recall. However, this might be German specific, because of noun capitalization. It indicates that all Hearst-pattern-based algorithms depend heavily on the syntactic parsers. Compared to both size-corrected baselines and the plain baselines given in section 6.2, the ranking is always worse, but the preference for only one relation is much stronger. This is not surprising considering the following examples of most typical sentence structures extracted:

```
* , A , * , B , *
* ( A ) * ( B ) *
* , B , A und *
B c A , *
* ( B C C ) Millionen A .
```

UnsymSim also produces predominantly symmetrical paradigmatic relations as opposed to hierarchical ones. In the results of *unsymSim*, used here as a training set, word pairs standing in a hierarchical paradigmatic relation were ranked higher. However, the amount of word pairs in a symmetrical paradigmatic relation was still several times higher (cf. Table 6.8). This means that learning from sentence rules always yields the one relation represented most frequently in the training set. On the other hand, the slight drop in precision also reflects that the predominance of the most frequent relation vs. all other relations in the training set has an influence, but is not vital.

cutoff	para	num pairs	measured
5 00	20.23	11 625	11 806
1 000	24.85	10 231	8 741
1 500	26.64	9 592	7 481
2 000	27.00	8 881	6 516
2 500	27.58	8 403	5 880

Table 6.11.: Performance of learning from signatures using method 1 and *justSim* as initial training data for various cutoff settings measured in MAP in %.

The threshold t_1, which is used to cut-off sentence structures based on their frequency has a large impact on the quality of the results. Table 6.11 shows that for extremely high cut-off values the mean average precision improves significantly, but with a severe drop in recall. This represents the typical precision/recall trade-off. However, with particular respect to the discussed evaluation issues discussed, it is necessary to reevaluate the results: whether out of 40 064 word pairs (the raw output of the algorithm with the highest threshold) really only the measured 8 400 are cohyponyms. A manual analysis of the result set revealed that most word pairs were proper names, such as *(Volkswagen, BMW)* or *(Sydney, Hongkong)*. Furthermore, a few word pairs could clearly be ruled out as cohyponyms and many mismatched in their word class. The rate of false negatives is estimated to be 81% (manually rating 1 000 randomly chosen pairs). Given the low agreement rate between GermaNet and the Annotation Project with respect to cohyponyms, this is not surprising. However, even if all 79 730 word pairs were correct cohyponym pairs, this is insignificant compared to the $119K$ cohyponym pairs in the Annotation Project and $608K$ in GermaNet.

These findings show that any additional information about word classes or sentence structure would greatly improve results. But in its present form, restricted to the unsupervised extraction of information, it is not yet viable to directly extract word pairs standing in a given relation based on a training set from another algorithm. Using the (previously mentioned) prototypical unsupervised part-of-speech tagger (Biemann, 2006b) as a simple same-word-class filter, the MAP for $t_1 = 2\,500$ could be improved to 31.23, which is an increase from 27.58. In further research, more combinations with such algorithms should be explored. Nevertheless, the method in its present form might still serve as an additional validation method for other algorithms.

sentence structure based learning	
Elefant	Löwe
Papier	Textilien, Glas, Holz, Maschinen, Keramik
Tschechien	Ungarn, Polen, Österreich, Slowakei, Spanien
Chopin	Beethoven, Mozart, Liszt, Schumann, Schubert

Table 6.12.: Top 5 ranked results for the four examples using sentence structure based learning (rule 1). The *justSim* method was used to produce the initial training data. The cutoff threshold was 1500 hits for the corresponding signature.

Table 6.12 also shows that the results do not differ much from the previous methods, except that in some cases (such as *Elefant*), there are less words in the result sets.

6.3.4. Cues from morphology

In German and some other languages, hyponyms are often constructed by combining a specific word with a more abstract one to form a compound, for example *Auto* (car) and *Fahren* (n. driving) are combined into *Autofahren* (car driving). In German, the head of the compound always remains on the right side and the compound can be head of a new compound. Though it is possible to take the position of the head into account, languages exist where there is no compounding. Furthermore, there are languages where the ordering within compounds is reversed (French). Consequently, this cue cannot be used as a language-independent method to extract hyperonymy. But if it exists in a given language, then a system designed to find hyperonyms might improve its performance by detecting that fact and utilizing it. Moreover, if there is no compounding in a language, or it plays only a minor role, then the same information must be conveyed by other means, such as multi word expressions in English (i.e. *car driving* instead of *cardriving*), which might be easier to utilize.

An experiment was conducted measuring the strength of this effect in German. This was done for two purposes: First, to measure the strength of this effect. If it is strong, it could provide further insights in how suitable the gold standard used in the sections above is for measuring the performance of the other algorithms. It could also be used to gauge the amount of false negatives measured in other experiments.

The algorithm used in this experiment was run on the entire corpus, not just the first 100 000 most frequent words. It produces two output table columns: *partOf* and *contains*. For each input word w it first acquires the 100 most (contextually) similar words, based on their significant co-occurrences. Then these 100 similar words are searched for any word, which either is contained in w or contains w. The former is a result summarized in the column *partOf* and the latter into column *contains* in Table 6.13. Before actually accepting any such word pair, two conditions must be met: First, if one string is cut from the other, then the remaining substring must be longer than 3 characters. Second, the string contained in the other, must be at the end or beginning of that other string. These simple rules ensure that the majority of errors is removed from the results. For example, the word *I* is contextually similar to *it*. It is also a part of the second word and without those rules, it would have to be accepted as a result.

This semi-supervised, language-specific algorithm produces very clean hyperonym-hyponym pairs for German. Thus, the effect indeed is quite strong in German. However, the purity of the results is not reflected when applying the mean average precision based on data from the Annotation Project, the same measure as in the previous sections, see Table 6.13. The mean average precision here (27.16) differs only marginally from the 27.58 (Table 6.11) of the entirely unsupervised sentence-signature-based learning in Section 6.3.3. Additionally, the total number of 22 360 words for which 5 287 correct hyponym/hyperonym pairs were found is roughly equivalent to the 8 403 cohyponym pairs which were found in the signature experiment with a severely restricted rule set.

A manual analysis of 1 000 randomly chosen negatives reported by the Annotation Project revealed that 73% of them were false negatives. This means that instead of 5 287 correct pairs out of 22 360 words, there were approximately about 19 000 correct pairs (taking only the first returned word for any input word). This confirms the discrepancy between the perceived (good) performance of those algorithms when inspecting examples and the performance measured (either by the Annotation Project or GermaNet). It also confirms the discrepancy with the base-

	partOf		contains	
	MAP	num	MAP	num
syn	0.86	193	1.02	67
para	3.86	863	8.58	449
h. para	27.16	6 074	26.92	1 886
deriv	0.56	125	1.52	86
other	1.22	272	1.20	81
total	32.95	7 367	38.06	2 476
hyperonym	0.08	17	**23.99**	1 673
hyponym	**23.60**	5 287	0.15	7

Table 6.13.: Results of semi-supervised algorithm extracting hyperonymy for German. The gold standard is the Annotation Project and MAP is in %. Total measured relevant words: 22 360 for *partOf*, 3 179 for *contains*.

lines described in Section 6.2. This means that the results reported in the preceding sections appear considerably worse than they actually are. In other words, the maximum achievable mean average precision using these evaluations is by far lower than 1.0. As a rough estimate, adding the 73% false negatives to the true positives results in about 0.45 (or 45.0%) of maximum achievable mean average precision using the Annotation Project and even less for GermaNet. This is precisely between the 33.90% and 61.74% MAP range that occurs when using one gold standard as an 'algorithm' and one as the evaluation gold standard (see Table 6.1).

Using morphology indirectly is a viable way to utilize information it encodes in the absence of a morphological analyzer. The unsupervised POS tagger by Biemann (2006b) uses a trie-based classifier to learn typical suffixes and prefixes for the intermediately built POS clusters to improve overall results. However, there are two different purposes. First, morphology can express semantic and grammatical information (often both are interlinked). For example in German, the suffix *-keit* is a derivational suffix, typical for nouns. Therefore it is syntactic information. Second, it also marks a certain semantic class of words expressing mostly abstract attributes. In a language where morphology is only used for grammatical information (which is in some way completely separated from semantic information), an algorithm intended to extract semantic relations would not profit from a morphological analysis. In most cases however, even an indirect morphological component will increase performance, even if only slightly.

6.3.5. Clustering methods

In the recent years, several articles described how knowledge extracted from a corpus can be represented as a graph (Widdows and Dorow, 2002; Bordag, 2003; Biemann, 2006a). Such a graph is then analyzed for local structural properties in order to obtain information about the participating words. Words are taken to be the nodes of the graph. Any relational information between words, such as statistically significant co-occurrences, similarity, morphological dependence etc., can be used as sources for edges between the nodes. An example of a structural property is the observation that there is a group of 10 completely interconnected words (i.e. they form a clique), because they are all significantly co-occur with each other.

Structural information from graphs can be exploited in several ways. One is to apply an algorithm separating the entire graph into subgraphs (Biemann, 2006a). The result can be a partition of the corresponding words into POS classes (Biemann, 2006b), topics, or languages (Biemann and Teresniak, 2005). This depends on the underlying method of defining the edges. Another way is to separate a subgraph around a given word in order to obtain distinct usages of that word (Bordag, 2003; Ferret, 2004) (as a reformulation of the algorithm presented in Chapter 4). It is further possible to add a certain amount of correct information (to a co-occurrence graph, for example) and enlarge this information by making use of the structure in the graph (Biemann, Bordag, and Quasthoff, 2004; Biemann and Osswald, 2004). A simple visualization of the graph or its parts is useful for certain applications as well (Biemann et al., 2004).

It is important to note though, that apart from the visualization, every algorithm utilizing a graph structure can also be reformulated in terms of clustering or classification. Nevertheless, a visualized graph allows very intuitive and obvious explanations as well as formulations of algorithms.

To demonstrate the usefulness of reformulating knowledge extraction as a graph algorithm, an algorithm extracting cohyponymy is described in this section. The extraction of cohyponymy can be reformulated as the extraction of cliques. A clique in graph theory is a set of completely interconnected nodes. The hypothesis is that cohyponymy is a transitive and symmetrical relation: If Word A is cohyponym of B and B is cohyponym of C, then A is a cohyponym of C. Because of the symmetry this also means that C is cohyponym of A. Transitivity in combination with symmetry may also hold true for other relations, such as synonymy. Nevertheless, transitivity and symmetry can be translated into a corresponding extraction algorithm. The input is a graph, where the words are nodes with edges between those co-occurring significantly. In fact, any method from the preceding sections can be used as edge definitions, especially *cooccSim* or *justSim*. The algorithm then extracts all cliques, i.e. all subsets (given a minimum size parameter of 5) of nodes in the entire graph which are completely interconnected.

Clique-based clustering

The actual implementation of this algorithm is more difficult. Since the task to find cliques in graphs is extremely time-consuming (NP-complete), a heuristic is used instead. The heuristic extracts the largest clique a given input word w is part of using the following algorithm:

1. input is word w and the set of at most t_{nbmax} neighbours $nb \in NB(w)$ of w
2. Let C_w be a set of cliques, each consisting of w and one nb
3. for each C_w create a set $C'_w = \{w\}$
   - for each neighbour $nb \in NB(w)$ in decreasing order of significance of co-occurrence with w: if neighbour nb is connected to each word of the clique C_w add it to that clique
4. if the largest clique C_{max} contains at least t_{noise} words:
   - remove all cliques whose size-difference to C_{max} is larger than t_{diff}
5. cluster remaining cliques based on overlap of contained words and merge the cliques in each cluster into one word set representing that cluster
6. one such merged cluster is chosen by using a combination of size and average ranking of contained words with respect to the initial ranking of the neighbours
7. the result is printed out with the ranking of each particular neighbour equal to its original ranking in the set of neighbours $nb \in NB(w)$

The first steps (1 through 3) of this algorithm constitute a bottom-up approach: Minimal cliques are enlarged as long as it is possible to find a new word, which is connected to all other words of the current clique. Note that the parameter t_{nbmax} is used when checking whether a connection between two words w_1 and w_2 exists: The connection is assumed to exist only if one of the words is at most the t_{nbmax}-nth ranked neighbor of the other. The heuristic in this approach is that not all possibilities are explored. For example, if the neighbour nb_{44} is added to clique C_{24}, it might prevent nb_{45}, nb_{46} and nb_{47} to be added. The order in which the neighbours are chosen depends on their ranking according to the method which produced the graph, and therefore the most significant co-occurrence is attempted first.

Steps 4 and 5 essentially constitute a noise filter. Even in a randomly created graph it is possible that a given node belongs a small clique of three or four words. In a word graph however, almost any word is part of several small cliques of at least three to four words. These cliques cannot be exploited in any useful way, because there are too many various reasons for them (near-duplicates of sentences in the corpus, MWEs, proper names, function words, etc.).

The remaining cliques typically reflect the ambiguity of the input word. Usually, there are several groups of cliques, where within each group the cliques nearly totally overlap, while differing strongly from the cliques of the other groups. In step 6 these groups are identified and the cliques which constitute such a group are merged. Thus, a side-effect of this algorithm is yet another word sense induction.

However, the unintended word sense induction is highly problematic in this case. Usually, only one of the meanings of a word is related to the cohyponyms of that word. For example, *zoo* might have two such groups, one representing typical sightseeings of a city (museum, famous restaurant, city center, etc.) and another representing the transportation needed to arrive at the zoo (tram, bus, subway, highway, etc.). The difficulty lies in choosing the one group which most likely represents the cohyponyms, assuming that only one of the meanings of the input word has (many) cohyponyms. Using the hypothesis that on average there are more cohyponyms (than words in any other relation) and they are higher ranked, a score is computed in step 7 for each group. It combines the number of words in that group with the average ranking of them. The group with the highest score is selected to be the result. Any other method of selecting a group leads to significantly worse results.

In the final step 8, the selected group is printed. The order in which the words are printed out corresponds to their initial ranking as neighbours of the input word w. This produces results more comparable to the initial ranking and other baselines. It is noteworthy that the final group of words selected by this algorithm is not a clique in that some of the links are missing. However, the number of links is almost at the maximum.

It would have also been possible to directly cluster the neighbours of the input word w instead of searching for cliques and clustering those. However, at least one effect is responsible for a decrease of the quality of the results from clustering algorithms. This effect can be best explained by the following example. A group G of 10 completely interconnected words was already found. The next step is to decide whether a further word w_{11} fits into this group. This word has only three edges, all connecting it to words in group G. Most cluster algorithms would clearly decide in favor of adding w_{11} to G. However, because 7 more links are missing, the clique-based algorithm would clearly decide to not add it.

Evaluation of clique-based clustering

The evaluation of the clique-based clustering of words should provide answers to several questions.

1. Is it better to use *justSim* or *cooccSim* to produce the graph, because both methods already perform fairly well in finding cohyponyms?

2. Is the ranking produced by the algorithm better than the size-adjusted baseline?

3. How do the several introduced parameters t_{nbmax}, t_{noise} and t_{diff} influence the results?

Table 6.14 answers the first two questions. Using *cooccSim* as the source of edges to define the graph only 5 234 input words remain for which near-cliques were found, which at the same time are part of the Annotation Project. The mean average precision for paradigmatic relations for these words is 14.06, which is clearly higher than the 11.86 of the size-adjusted baseline *cooccSim*. This is also better than the 9.10 of the full *cooccSim* baseline in Section 6.2. Clearly, this is a trade-off in favor of precision. In this case it is a positive trade-off, because not all words necessarily have a large amount of cohyponyms. Obviously this algorithm is able to select words more likely to have many cohyponyms. Furthermore, the algorithm is able to filter out non-cohyponyms, which is why the rating of the paradigmatic relations outperforms the baseline. In order to show that indeed most of the relevant paradigmatic relations are cohyponyms, in the Table 6.14 additionally to the summarizing but abstract paradigmatic relations, the specific relation ‘noun cohyponyms’ was included.

	orig cooccSim		clique cooccSim		orig justSim		clique justSim	
	MAP	num	MAP	num	MAP	num	MAP	num
syn	5.80	3 519	5.76	3 243	3.93	3 303	3.95	3 020
para	12.13	11 252	**13.82**	12 061	11.85	13 626	12.33	14 067
h. para	6.29	2 770	6.09	2 780	8.05	4 112	6.92	3 747
deriv	1.10	362	1.23	366	1.21	547	1.17	510
other	3.14	1 361	3.17	1 355	2.49	1 454	2.63	1 427
n cohyp	9.40	9 498	10.82	10 169	8.23	10 937	9.02	11 431
total	22.57	18 422	24.58	18 973	21.29	22 029	21.32	21 880
relevant	10 564		10 564		13 835		13 835	

Table 6.14.: Clique-based clustering of words, comparison of baseline ranking with the ranking produced by the algorithm as well as a comparison of different edge sources (*cooccSim* and *justSim*). MAP in %. $t_{nbmax} = 20$, $t_{noise} = 5$ and $t_{diff} = 1$

As obvious from Table 6.14, both the results of the algorithm and the baseline achieve relatively high scores for syntagmatic and hierarchical paradigmatic relations. This makes it impossible to treat the results as an almost pure source of cohyponym relations. Filtering the syntagmatic relations can be achieved relatively easily utilizing a word class filter. Again, using Biemann’s prototypical part-of-speech tagger as a word class filter on the results of the clique algorithm on the *cooccSim*-based graph, it was possible to increase MAP for paradigmatic relations from 13.82 to 17.57 while simultaneously decreasing MAP for syntagmatic relations from 5.76 to 3.35. Alternatively, if the simple distinction between capitalized words (nouns and proper names in German) and non-capitalized words (all other words) is included (as a semi-supervised method), the ranking of syntagmatic relations is as low as 1.09 MAP. The MAP of all other relations is unaffected. Thus, the combination with other algorithms bears potentially large improvements.

The same issue is not as clear in the case of hierarchical paradigmatic relations. Hyperonyms of a set of cohyponyms are both connected to each of the cohyponyms and vice versa. A combination with the *unsymSim* method might be helpful here, but several experiments failed to (clearly) produce the desired divisive effect. However, *unsymSim* alone clearly favors hierarchical paradigmatic relations, see Table 6.8.

When *justSim* is used as the underlying graph instead of *cooccSim*, the algorithm returns near-cliques for more input words, but the precision values are lower. The increase in precision, compared to the corresponding size-adjusted baseline of *justSim* is only half of that shown in the previous results. This supports the hypothesis that various algorithms can boost each others performance, if they are independent. Hence, knowledge about contextual similarity and co-occurrence significance is combined with transitivity and symmetry in the resulting graph, and this combination performs better than any isolated solution. For example, using *justSim* (as opposed to *cooccSim*) as the graph is only a partial solution, because knowledge about co-occurrence significance is not used directly (only indirectly for computing contextual similarity).

When using *justSim*, the mean average precision values for the symmetrical paradigmatic relations are lower than those of *cooccSim*, but the values for the hierarchical paradigmatic relations are higher. This supports the hypothesis (formulated in Section 6.3.1) that *justSim* does not differentiate between symmetrical and hierarchical relations, whereas *cooccSim* does.

clique-based clustering with $t_{nbmax} = 20$, $t_{noise} = 5$ and $t_{diff} = 1$	
Elefant	Tiger, Löwe, Leopard, Nashorn, Zebra, Giraffe
Papier	Holz, Glas, Metall, Plastik, Kunststoff
Tschechien	Polen, Ungarn, Rumänien, Bulgarien, Slowakei
Chopin	Werke, Bach, Klavier, Mozart, Beethoven

Table 6.15.: Top 5 ranked results for the four examples using the clique-based clustering

The examples in Table 6.15 show that in comparison to previous methods the results are slightly cleaner. Except for the results of *Chopin*, only pure cohyponyms were extracted. Such results were not achieved by any previous method, which is also reflected in the evaluations.

To explore the effects of various parameters on the quantity and quality of the results, several experiments were run, each varying one of the three parameters separately from the fixed starting point of $t_{nbmax} = 20$, $t_{noise} = 5$ and $t_{diff} = 1$.

	$t_{diff} = 0$		$t_{diff} = 1$		$t_{diff} = 2$		$t_{diff} = 3$	
	alg	base	alg	base	alg	base	alg	base
syn	5.65	5.75	5.76	3.93	5.77	5.80	5.79	5.81
para	14.04	12.24	13.82	11.85	13.67	12.10	13.59	12.07
h. para	5.95	6.22	6.09	8.05	6.10	6.31	6.12	6.32
deriv	1.21	1.08	1.23	1.21	1.24	1.10	1.24	1.11
other	3.11	3.07	3.17	2.49	3.24	3.16	3.26	3.16
n cohyp	10.99	9.50	10.82	8.23	10.07	9.37	10.63	9.34
total	24.98	22.87	24.58	21.29	24.35	22.44	24.27	22.38
relevant	10 564		10 564		10 564		10 566	

Table 6.16.: Various parameter settings for t_{diff} using *cooccSim* as the source data for the graph

The first series of experiments altering t_{diff} is evaluated in Table 6.16. A setting of $t_{diff} = 0$ essentially disables the last three steps of the algorithm, because in most cases there is only one largest cluster that is then directly given as the result. On the other hand, increasing this parameter allows more words to be added to the initially found clique through the clustering step. This means that in the resulting set of words, less and less links are necessary for the clique to be accepted. The small changes to the results allow the conclusion that this parameter does not have a significant influence, especially since it cannot be set to much higher values, because in most cases the largest found clique contains less than 20 words.

	$t_{nbmax} = 20$		$t_{nbmax} = 40$		$t_{nbmax} = 60$	
	alg	base	alg	base	alg	base
syn	5.76	3.93	4.92	5.07	4.34	4.43
para	13.82	11.85	9.93	9.14	8.65	7.92
h. para	6.09	8.05	5.10	5.47	5.15	5.52
deriv	1.23	1.21	0.96	0.82	0.89	0.79
other	3.17	2.49	2.50	2.40	2.26	2.17
n cohyp	10.82	8.23	7.49	6.66	6.33	5.61
total	24.58	21.29	17.91	17.12	15.55	14.90
relevant	10 564		19 593		23 827	

Table 6.17.: Various parameter settings for t_{nbmax} using *cooccSim* as the source data for the graph

The next series of experiments alter t_{nbmax}. The evaluation in Table 6.17 shows a strong degradation of precision with higher settings. The parameter determines the maximal amount of edges for each node to construct the graph from. Thus, if *justSim* was set to compute 100 most similar words, then with $t_{nbmax} = 20$ only the 20 most similar are taken for any given input word. Because for any input word B the word A might be the 20th most similar word to B, but B only the 30th (or worse) most similar word to A, the resulting graph is most likely a directed graph. Therefore each link is treated as symmetrical during the construction of the graph. This means that irrespective of the setting of t_{nbmax} to 20, any given node in the graph will tend towards having more than 20 edges connecting it to other nodes. However, increasing this parameter also increases the overall number of edges.

Obviously, increasing this parameter allows weaker associations to be included in the graph. This also explains the degradation of precision. However, the decrease in precision occurs with a strong increase in recall. This increase is stronger than the decrease in precision. But it is also a fact that while the precision for paradigmatic relations drops with higher settings, the precision for all other kinds of relations remains virtually unchanged. This supports the claim that this method utilizes the inherent properties of cohyponyms. To conclude, this parameter can be used to manipulate the tradeoff between precision and recall. But a too high setting might produce results worse than the baseline.

	$t_{noise} = 4$		$t_{noise} = 5$		$t_{noise} = 6$	
	alg	base	alg	base	alg	base
syn	5.46	5.62	5.76	3.93	6.10	6.03
para	12.54	11.29	13.82	11.85	16.13	13.35
h. para	5.49	5.77	6.09	8.05	6.62	6.52
deriv	1.05	0.93	1.23	1.21	1.57	1.32
other	2.59	2.52	3.17	2.49	3.73	3.61
n cohyp	9.22	8.16	10.82	8.23	13.08	10.08
total	22.21	20.88	24.58	21.29	28.05	24.47
relevant	17 771		10 564		5 733	

Table 6.18.: Various parameter settings for t_{noise} using *cooccSim* as the source data for the graph

The last series of experiments manipulates t_{noise}. The results are shown in Table 6.18. The contribution of this parameter is to define the smallest acceptable clique. Setting $t_{noise} = 4$ means that: apart from the input word only three other words, and the 6 possible edges between them are needed to produce an output. Approximately every second word is part of at least one such clique, but only every fourth word is part of a $t_{noise} = 5$ clique. Each increase of this parameter halves the amount of words for which an acceptable clique can be found. This corresponds also to the observations about Small Worlds in word graphs (Ferrer i Cancho and Sole, 2001; Steyvers and Tenenbaum, 2005).

In addition to the strong decline in recall there is also a strong increase in mean average precision. The stricter this parameter was set, the greater the increase of precision. However, the size-corrected baseline also improves in precision at approximately the same rate. Thus, the raw contribution of the algorithm is an absolute increase of about 3% in MAP (the relative increase varies for different threshold settings). This algorithm additionally produces a good selection with respect to which words have (easily) extractable cohyponyms vs. words possibly having few or none at all.

A manual analysis of the result files of the best-performing settings $t_{noise} = 6$ revealed that the primary source of mistakes is syntagmatic dependency. As noted in Section 6.3.1, the method *cooccSim* cannot differentiate between syntagmatic and paradigmatic dependencies. This is because the similarity computations do not account for artificially high similarity values of word pairs co-occurring with each other too frequently. For example, *Sri* and *Lanka* are contextually very similar, because their respective global contexts, based on sentence co-occurrences are identical - they always co-occur. For the clique-based clustering it means that (for example) the typical verb such as *drinking* or a modificator of a number of cohyponyms such as *beer, wine, juice, coffee* is connected to all the cohyponyms. The cohyponyms are also fully interconnected. This subsequently leads to the clique *drinking, beer, wine, juice, coffee*, which is not perfect. A word class filter might be helpful, but improved similarity computation taking too frequent co-occurrence into account is the correct solution.

6.4. Conclusions

The various algorithms tested in this chapter clearly show that it is possible to influence the type of relations extracted. The algorithms are based on the basic syntagmatic vs. paradigmatic distinction, as well as abstract properties such as symmetry and transitivity. On other words, no language-specific knowledge is utilized and it can be expected that they perform similarly for other languages. However, being able to influence does not mean being able to clearly extract words in a given relation. Hence, the most likely use for such algorithms in their current state is as parts of semi-supervised systems, increasing their hit-rate (and therefore productivity) for certain relations. Other possible uses include advanced Information Retrieval and classification algorithms, or applications visualizing word associations, such as Semantic Talk (Biemann et al., 2004).

Despite such restrictions, this chapter also provides rich information on how new methods can be designed. Specific examples discussed include the co-occurrence frequency adjusted similarity or utilizing unsupervised POS tagging to clean result sets leading to better sentence structure learning. But also using word sense induction, utilizing MWE extraction algorithms to fuse words into multi words (and treat them as single units), and many more.

This chapter also shows how using a proper underlying model such as the one proposed in Chapter 2 allows for the direct translations of linguistic hypotheses into extraction algorithms. At the same time it shows that such hypotheses are often incomplete or interfere with other hypotheses (such as symmetry, which holds both for cohyponymy and hyperonymy). In this respect this chapter represents only a further step down the road of a better understanding of language in general and lexical semantic relations in particular.

7. Conclusions

This thesis develops a model (SIML) for automatic lexical acquisition based on a minimal set of structuralist principles.

The theoretical part (Chapters 1 and 2) examines relevant aspects of structuralism, language sampling and other topics that form the basis of the model and the algorithms developed within its framework. The core concepts evolve around the definition of syntagmatic and paradigmatic relations between linguistic units. This thesis proposes two main hypotheses. One states that these principles operate equivalently on all language levels. The second states that any knowledge about language units can be derived from observing the language usage. Any algorithm based on these two hypotheses can be designed in a language independent manner. Other concepts, such as linguistic categories, syntactic and semantic agreement, or semantic primitives are discussed and demonstratively expressed in terms of the model.

The empirical part (Chapters 3 to 6) validates the model. Each chapter addresses several key aspects of unsupervised and knowledge-free lexical acquisition. At the beginning of each chapter, existing algorithms are classified and interpreted with respect to their adherence to the introduced principles. Subsequently, new algorithms are introduced, along with thorough evaluations of their performance. The algorithms include new or significantly improved solutions for computing word similarity, word or morpheme ambiguity, morpheme segmentation and semantic relations.

Figure 7.1 outlines how the discussed algorithms relate to each other. The core concept is to begin with a raw collection of text and employ increasingly complex analyses utilizing previous steps to acquire further pieces of lexical knowledge. This thesis covers a selection of acquired lexical knowledge pieces based on combinations of previous steps:

- **Semantic similarity of word forms.** (Chapter 3) A simulation of contextual (i.e. semantic) similarity is achieved by comparing significant co-occurrences of the word forms. The quality depends on how significance of co-occurrence is judged, as well as other external factors, such as corpus size or word form frequency.

- **Distinct senses of word forms.** (Chapter 4) The significant co-occurrences are used as features in a triplet-based clustering word sense induction algo-

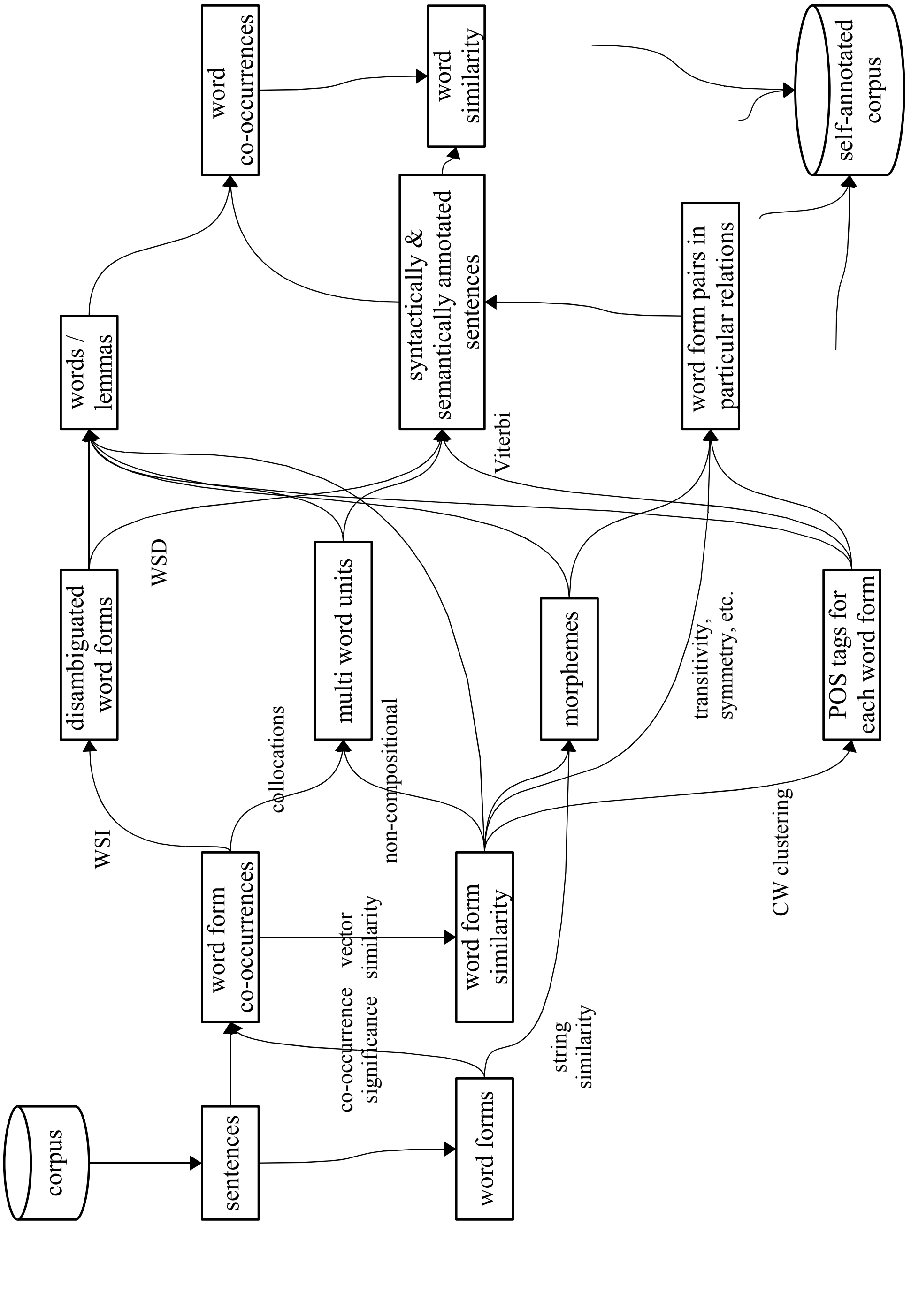


Figure 7.1.: Interactions of various knowledge acquisition algorithms.

rithm. The result is a set of word form groups that represent various distinct usages of the input word.

- **Segmentation of word forms into morphs.** (Chapter 5) The simulated semantic similarity enables candidate word forms carrying identical syntactical or semantic attributes to be found. Using these candidates, the performance of the existing LSV method increases significantly over using it without such a restricted set. Combined with a Trie based machine learning algorithm it reliably detects true morpheme boundaries.
- **Distinct senses of morphs.** (Chapter 5) Using the obtained morpheme boundaries the initial corpus was segmented into morphs, allowing for a reapplication of the co-occurrence analysis. Then the word sense induction was computed on morphs, instead of word forms. The results, similar to those when distinguishing word form senses, support the hypothesis of identical principles operating on different language levels.
- **Different types of relations.** (Chapter 6) Measuring abstract properties, such as symmetry or transitivity of co-occurrence or contextual similarity, introduces methods to distinguish between types of relations, especially between syntagmatic, paradigmatic and hierarchical paradigmatic relations.

Hence, one part of the figure contains existing solutions developed in this thesis with the required algorithm combinations, whereas the other part remains mostly hypothetical. Nevertheless, a side effect of any algorithm is the further enrichment of the annotations of the initially raw corpus. However, the entire information is derived solely from the corpus itself.

Many further possible algorithmic combinations were discussed throughout the thesis. Several such combinations are most promising for further research, because they depend only on existing or introduced algorithms, as opposed to hypothesized ones. As an example, one important goal is to acquire knowledge about the adherence of individual word forms to their lemmas and about the transformation rules that produced the originally observed word forms. The two most relevant algorithms for this task are the contextual word form similarity and the morpheme segmentation. Additionally, knowledge about the various senses (resulting from a WSI algorithm or a multi word unit extraction algorithm) of each word form could be useful, but at the very least these senses would also need to be merged across different word forms.

Essentially every single algorithm can be used to annotate the initial raw sentences. Discussed examples include: syntactic information using abstract POS tags (Biemann, 2006b), semantical information such as word sense annotations, morpheme classes or semantic relations to other words in the same sentence. Such annotated sentences would enable finer grained analyses, such as detecting syntactic or rare senses of words, or enriching the lexical knowledge about each individual word. The detection of multi word units (i.e. collocations, idiomatic expressions) could be improved by computing the modified contextual similarity (discussed in Chapter 6) combined with results of the morpheme segmentation algorithm and POS tagging.

However, despite the academically interesting results, the low accuracy of the acquired lexical knowledge (as detailed by the evaluations in each chapter) certainly is a limiting factor with respect to the possible applications of the algorithms. Unsupervised and knowledge-free algorithms clearly cannot (and in the foreseeable future will not) compete with semi-automatically or manually acquired lexical knowledge.

Nevertheless, because they require no more than raw text and processing power, they can always be used as a fall-back solution. Since they are inherently language and domain independent, they can be used on any language or domain specific task while retaining the same overall quality. They can also be used for many applications where an approximation of the corresponding lexical knowledge suffices. In some cases, when applied to highly specialized domain specific corpora, they may even outperform algorithms based on manual lexical knowledge. In short, research into unsupervised and knowledge-free algorithms currently represents the extreme of being able to (cheaply) provide large quantities and good coverage while sacrificing accuracy.

Finally, this thesis demonstrates that research on unsupervised and knowledge-free algorithms, although in its early stages, yields great potential. It is very motivating that every advancement yields algorithms that directly improve the performance of applications (for example morpheme segmentation improving speech recognition (Kurimo et al., 2006)) or enable entirely new applications (such as SemanticTalk (Biemann et al., 2004)). Finally, it is motivating that this kind of research improves understanding of what language is and how it works.

A. Appendix: Corpora

The majority of algorithms developed throughout this thesis are tested on two languages - English and German. Correspondingly, two corpora are used. For English the British National Corpus (BNC) was chosen, which is frequently used in related work. However, apart from the tokenization (available in the BNC) no further preprocessing (such as lemmatization) was used.

Berliner Zeitung	8 140 465
Die Welt	5 992 459
Sddeutsche Zeitung	4 030 173
Die Zeit	1 893 385
Die Tageszeitung	1 854 209
Süddeutsche Zeitung	1 732 464
Stuttgarter Zeitung	1 729 573
Frankfurter Rundschau	1 500 720
Bild	1 053 943
Tagesspiegel	975 889
Der Spiegel	603 427
Junge Welt	448 540
Projekt Gutenberg	327 652
Junge Freiheit	214 443
OTS-Newsticker	158 362
Freitag	153 238
Rheinischer Merkur	149 529
Frankfurter Allgemeine	141 785
Netzeitung	134 034
Schweriner Volkszeitung	97 324
Onvista Wirtschaftsnews	84 245
Spektrum Wissenschaft	77 295
Neue Juristische Wochenschrift	71 905
Neues Deutschland	63 762
Financial Times Deutschland	62 983

Table A.1.: A selection of the largest contributors to the 'Wortschatz Project', ordered by amount of sentences.

For German, parts of the 'Projekt Deutscher Wortschatz' (Wortschatz) (Quasthoff, 1998), a corpus initiative of the Natural Language Processing Department, Univerisity of Leipzig, were used. This corpus is based on newspaper texts of several prominent newspapers (see Table A.1 for an overview) and other sources, such as the German ‘Projekt Gutenberg’[1]. The corpus grows gradually with new releases every few years with the inclusion of recent newspapers and is available online[2]. The subcorpora used throughout this thesis are all based on the 2004 release, which contains 35 million sentences (517 319 977 running word forms).

For Chapter 3, a small subcorpus of 100 million running word forms (5.95 million sentences) was created by randomly selecting sentences from the entire corpus. This was done to match the size of the BNC to make the results obtained more comparable. Contrarily, for the corpus size influence experiments (Section 3.5.1), several subcorpora were created by drawing one million sentences from the entire corpus in chronological ordering for each increase in corpus size.

However, for Chapters 5 and 6, an 11 million sentences part of the entire corpus was used. The 11 million is an attempt for a trade-off between increased quality of results and run-time restrictions posed by some of the more complex algorithms.

Table A.2 outlines the main statistics of both corpora. It shows that the English on average has slightly longer sentences. It also shows that it has significantly less distinct word forms and the the word forms on average are shorter.

	BNC	German subcorpus
running word forms	110 619 231	104 186 236
distinct word forms	660 539	3 768 208
sentences	5 150 632	5 950 632
avg. word form length	4.15	5.21
avg. sentence length	21.48	17.51
most frequent word	the(5 191 153)	die (2 956 863)

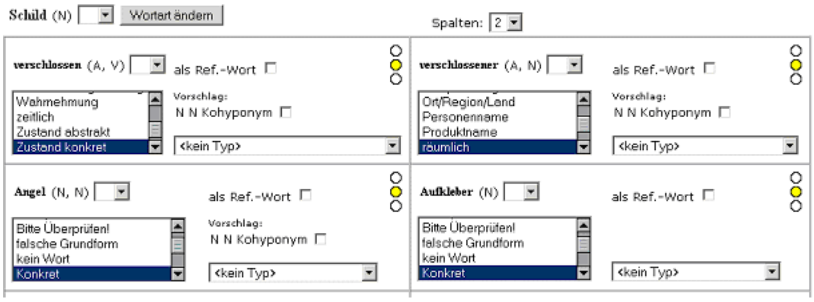
Table A.2.: Statistics for some of the used corpora.

[1] http://gutenberg.spiegel.de

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

B. Appendix: The Annotation Project

The Annotation Project is an initiative of the Natural Language Processing Department, Univerisity of Leipzig, to create a semantic net. It was first publicly described in Biemann (2005b). The goal was to acquire as broad a coverage as possible while keeping the effort on behalf of the annotators at a minimum. The annotation phase of the initiative lasted one year and six annotators participated in it.



Schild (N) Wortart ändern

Spalten: 2

verschlossen (A, V) als Ref.-Wort □
Wahrnehmung
zeitlich
Zustand abstrakt
Zustand konkret
Vorschlag:
N N Kohyponym □
<kein Typ>
○ ● ○

verschlossener (A, N) als Ref.-Wort □
Ort/Region/Land
Personenname
Produktname
räumlich
Vorschlag:
N N Kohyponym □
<kein Typ>
○ ● ○

Angel (N, N) als Ref.-Wort □
Bitte Überprüfen!
falsche Grundform
kein Wort
Konkret
Vorschlag:
N N Kohyponym □
<kein Typ>
○ ● ○

Aufkleber (N) als Ref.-Wort □
Bitte Überprüfen!
falsche Grundform
kein Wort
Konkret
<kein Typ>
○ ● ○

Figure B.1.: Screenshot of the web-based annotation tool. The input word is in the upper left corner, with each possibly associated word given as an entry.

Before and during the annotation several tools were created to aid the annotators in their undertaking. These tools combined externally provided knowledge (such as results of the algorithms) and previously annotated knowledge to make each particular annotation step a matter of choosing between very few possibilities. Essentially, the entire system of tools was designed such that its proposals should only be revised by annotators.

The majority of externally provided knowledge was the result of co-occurrence measurements, as they are described in Chapter 3. Given an arbitrary input word this allows to provide a list of related words. Assuming that cohyponymy is the most frequent relation, it can be preset for each word found to co-occur frequently with the input word.

The first annotation tool (Figure B.1 is a web-based form. Using externally provided, as well as existing relations, it allows to view the general usage of the input word. The provided controls allow to confirm the proposed relations, or choose different ones. Additionally it allows several annotators to work on the same data base.

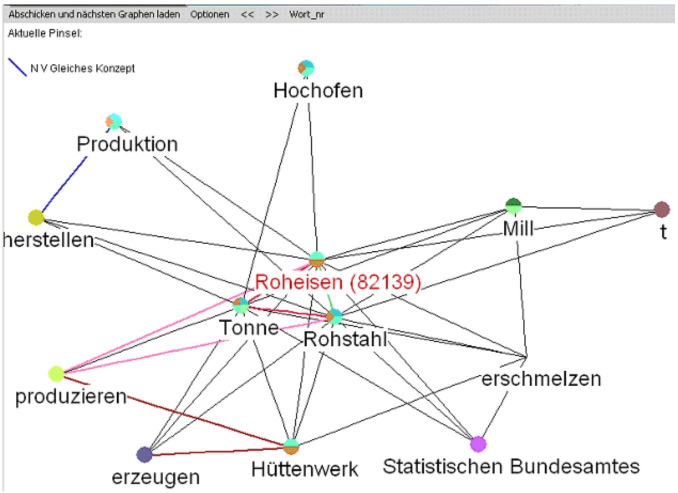
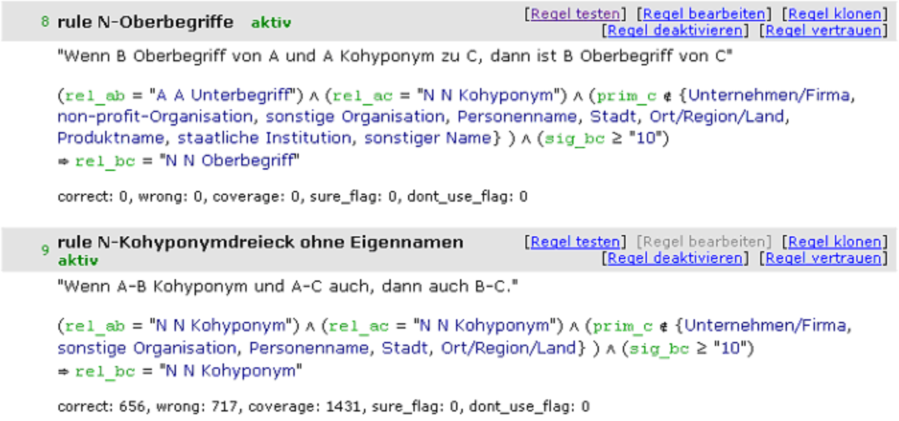


Figure B.2.: Screenshot of the graph visualization annotation tool. Colors in the nodes and the links identify semantic primitives and relation types.

The second annotation tool (Figure B.2) uses a force-directed arrangement of the words in a graph to visualize their relations. Each element (nodes and links) can be selected and an annotation chosen. Selecting a node opens a menu with selectable primitives for that node, while selecting links offers relation types. The nodes to be shown in each graph are selected by taking an input word and its significant co-occurrences, as well as other words already annotated as standing in relation with the input word. The links (between any two words) in the graph are based on whether a co-occurrence of these words was found significant or a relation was previously annotated between them.



8 **rule N-Oberbegriffe** aktiv [Regel testen] [Regel bearbeiten] [Regel klonen] [Regel deaktivieren] [Regel vertrauen]

"Wenn B Oberbegriff von A und A Kohyponym zu C, dann ist B Oberbegriff von C"

(rel_ab = "A A Unterbegriff") ∧ (rel_ac = "N N Kohyponym") ∧ (prim_c ∉ {Unternehmen/Firma, non-profit-Organisation, sonstige Organisation, Personenname, Stadt, Ort/Region/Land, Produktname, staatliche Institution, sonstiger Name}) ∧ (sig_bc ≥ "10")
⇒ rel_bc = "N N Oberbegriff"

correct: 0, wrong: 0, coverage: 0, sure_flag: 0, dont_use_flag: 0

9 **rule N-Kohyponymdreieck ohne Eigennamen** aktiv [Regel testen] [Regel bearbeiten] [Regel klonen] [Regel deaktivieren] [Regel vertrauen]

"Wenn A-B Kohyponym und A-C auch, dann auch B-C."

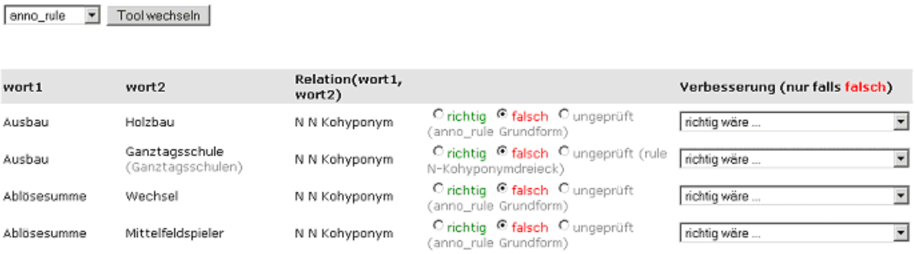
(rel_ab = "N N Kohyponym") ∧ (rel_ac = "N N Kohyponym") ∧ (prim_c ∉ {Unternehmen/Firma, sonstige Organisation, Personenname, Stadt, Ort/Region/Land}) ∧ (sig_bc ≥ "10")
⇒ rel_bc = "N N Kohyponym"

correct: 656, wrong: 717, coverage: 1431, sure_flag: 0, dont_use_flag: 0

Figure B.3.: Screenshot of the rule editor tool. Each rule can be viewed, edited, deleted or copied. For each rule a statistic about its predictive power is tracked.

Additionally to the direct annotation tools, the ‘rules tool’ (Figure B.3) allows to define and apply meta-rules to automatically generate further proposals. For example, it is possible to define the transitive closure for cohyponyms, i.e. if A is already annotated as being a cohyponym of B and B cohyponym of C, then A is probably also a cohyponym of C. This tool keeps track, for each rule, how often predictions made by it were correct, thus allowing to refine rules, delete bad rules altogether, or assign weights to rules.

Out of all automatically generated proposals of words standing in a particular relation with each other, the co-occurrence measures and meta-rules proposed roughly equal amounts. The above mentioned cohyponymy transitivity closure was by far the most productive one.



wort1	wort2	Relation(wort1, wort2)		Verbesserung (nur falls falsch)
Ausbau	Holzbau	N N Kohyponym	○ richtig ● falsch ○ ungeprüft (anno_rule Grundform)	richtig wäre ... ▾
Ausbau	Ganztagsschule (Ganztagsschulen)	N N Kohyponym	○ richtig ● falsch ○ ungeprüft (rule N-Kohyponymdreieck)	richtig wäre ... ▾
Ablösesumme	Wechsel	N N Kohyponym	○ richtig ● falsch ○ ungeprüft (anno_rule Grundform)	richtig wäre ... ▾
Ablösesumme	Mittelfeldspieler	N N Kohyponym	○ richtig ● falsch ○ ungeprüft (anno_rule Grundform)	richtig wäre ... ▾

Figure B.4.: Screenshot of correction tool. To decrease errors by particular annotators, this tool allows to review each decision.

Further supplementing the annotations, a correction tool (Figure B.4) allows to systematically review annotations by other annotators or proposed annotations by algorithms or meta-rules.

Table B.1 gives an overview of the resulting size and contents of the Annotation Project. The total number of different word forms annotated in the Annotation Project is larger than in GermaNet (which contains only lemmas), but still smaller than in WordNet. However, the amount of words contained both in the knowledge source and the corresponding corpus is largest for the Annotation Project. Contrarily, the density of annotated relations between these words is considerably richer for GermaNet and WordNet. Particularly the cohyponymy relation achieves huge counts of word pairs, because of the transitivity of the relation. The other relations result in comparable counts, which correspond to the general size of the corresponding knowledge source.

Despite such quantitative comparisons, it is also necessary to highlight several peculiarities about the Annotation Project. The most important one is the strong influence of the co-occurrence measure based proposals that were used to speed up the manual annotations. Despite the possibility to add words other than the proposed by the tools, the annotators mostly relied on the proposed ones. Hence, for each word only the most frequent usage is represented. This also introduces the bias for syntagmatic relations, as discussed in Section 3.5.7. Additionally, for

	Annot	GermaNet	WordNet
total words	75 728	52 620	146 212
words in corpus	75 728	40 703	57 990
distinct relations	58	13	26
cohyponyms	134 836	1 682 680	13 867 140
hyper(o)nyms	92 820	226 248	680 370
synonyms	47 258	53 758	297 508
n adj typ. property	18 073	n.a.	n.a.
n v typ. obj. of	15 425	n.a.	n.a.
part/consists of	16 844	17 166	69 400

Table B.1.: Statistics of the Annotation Project compared to GermaNet. All relations are assumed to be directed, meaning that counts for symmetric relations are doubled.

an unknown reason, the relation *A N Derivation* (derivational relation between an adjective and a noun) comprises nearly exclusively proper names and the adjective variants of them. This is in comparison to all other relations which comprise sets of word pairs that fit better to the name of the relation.

To illustrate the differences between the Annotation Project and GermaNet (additionally to Table 6.1 in Chapter 6 which measures the overlap) for two words the cohyponyms, hyperonyms and hyponyms are listed, highlighting mismatches between the two knowledge sources (A = Annotation Project, G = GermaNet):

Papier

cohyponyms

A : **Aktie Alu Anleger Anleihe Bleistift Bundesbank Büchse Büchsen Dose Dosen Feder Folie Folien Gips Glas Holz** Karton **Kartonage Kartonagen Kartons Keramik Kunststoff Kunststoffe Leim Metall Nachmittag Optionsschein Papp** Pappe **Pappen Plastik Rendite Stift Tüte Tüten Weißblech Wellpappe Wertpapier**

G : **Ausweispapier Behindertenausweis Bibliotheksausweis Block Briefumschlag Fahrausweis Führerschein Heft** Karton **Kinderausweis Krankenversicherungsausweis Krankenversicherungskarte Kundenkarte Kuvert Papier Papierblock** Pappe **Pappkarton Parkausweis Pass Paß Personalausweis Rentnerausweis Schülerausweis Seniorenausweis Sozialversicherungsausweis Studentenausweis Videotheksausweis**

hyperonyms

A : Papierware

G : **Ausweis** Papierware

hyponyms

A : **Aktie Ausweis** Briefpapier Butterbrotpapier **Dokument Jagdschein** Krepppapier Packpapier Pauspapier Paß **Rentenpapier Schreibpapier** Seidenpapier **Wertpapier** Zeichenpapier Zeitungspapier

G : Briefpapier Butterbrotpapier **Krepp-Papier** Krepppapier Packpapier Pauspapier Seidenpapier Zeichenpapier Zeitungspapier

Elefant

cohyponyms

A : **Bär Bären Dickhäuter Elefanten Esel Flußpferd Hirsch Hyäne Hyänen Leopard Löwe Mammut Mammuts Maus Nashorn Nilpferd Raubkatze Raubkatzen Rhinozerosse Rüssel Tiger Tigern Zirkus**

G : **Elefant Hase Kaninchen Karnickel Reittier Schlachtvieh Vieh Zugpferd**

hyperonyms

A : Nutztier Rüsseltier **Tiere Tier**

G : Nutztier Rüsseltier

hyponyms

A : -

G : **'Afrikanischer Elefant' 'Indischer Elefant'**

These examples impressively demonstrate the disagreement between the annotators of the Annotation Project and GermaNet. Apparently there are at least four underlying factors resulting in such a low agreement.

One factor is a differing understanding of what the various relations really are. In the Annotation Project, the cohyponyms of *Papier* (paper) consist of many words not directly related to it, such as *Glas* (glass) and *Holz* (wood) sharing with *Papier* that they represent other forms of construction material. Contrary to that, the cohyponyms in GermaNet contain mostly words that could easily be hyponyms instead, such as *Kuvert* (envelope), being a special form of paper.

As the cohyponyms of *Elefant* demonstrate, another factor is certainly the lack of completeness in any of the knowledge sources. Despite this word being a fairly easy one to agree upon its related words, not a single match can be observed among the cohyponyms. In fact, the type of words encountered in both cohyponym sets is very similar. Yet apparently there are many more words related in a similar way to *Elefant* which simply are not annotated (close enough in the hierarchy of GermaNet, or not at all in the Annotation Project), such as *Erdferkel, Nasenbär, Kuh, Wasserochse, Maultier, Esel* and many more, depending on how coarse-grained the terminological hierarchy is.

The third factor is, whether the knowledge source (or the annotation process) is corpus based. For the Annotation Project, the examples show a clear preference for typical usages of the words in the newspaper corpus. Contrarily, in GermaNet the introspective approach produces a more consistent annotation, but also produces questionable examples, such as *Elefant - Zugpferd* (elefant, cart horse). Theoretically, these two words are cohyponyms. However, in modern German the meaning of *Zugferd* (cart horse) is not really used anymore. Instead it has a high frequency entirely due to its metaphorical usage of ‘a leading person’, hence it is not a cohyponym of *Elefant*.

The fourth factor is the usage of word forms in the Annotation Project as opposed to the usage of lemmas in GermaNet. Obviously, non-baseforms cannot match in GermaNet then. However, compared to the other factors, surprisingly this is responsible only for few mismatches.

Nevertheless, the quality of both knowledge sources suffices to enable large-scale evaluations based on them and thus makes them extremely valuable resources in the development of unsupervised and knowledge-free algorithms.

References

Agirre, Eneko, Olatz Ansa, Eduard Hovy, and David Martínez. 2000. Enriching very large ontologies using the www. In *Proceedings of the Ontology Learning Workshop at the ECAI*, Berlin, Germany, August.

Agirre, Eneko, David Martmnez, Oier Lspez de Lacalle, and Aitor Soroa. 2006. Evaluating and optimizing the parameters of an unsupervised graph-based wsd algorithm. In *Proceedings of the Human Language Technology Conference of the NAACL (HLT/NAACL)*, June.

Agirre, Eneko and German Rigau. 1995. A proposal for word sense disambiguation using conceptual distance. In *Proceedings of the First International Conference on Recent Advances in Natural Language Processing (RANLP)*, Tzigov Chark, Bulgaria, September.

Altmann, Gabriel. 1980. Prolegomena to menzerath's law. In R. Grotjahn, editor, *Glottometrika 2*. Brockmeyer, Bochum, Germany, pages 1–10.

Altmann, Gabriel and Viktor Krupa. 1964. On relations of structure and inventory in linguistic systems. *Jazykovedny casopis.*, pages 97–100.

Argamon, Shlomo, Navot Akiva, Amihood Amir, and Oren Kapah. 2004. Efficient unsupervized recursive word segmentation using minimun description length. In *Proceedings of The 20th International Conference on Computational Linguistics (COLING)*, Geneva, Switzerland, August.

Aronoff, Mark and Kirsten Fudeman. 2004. *What is Morphology?* Blackwell Publishing Limited, Boston, MA, USA.

Atwell, Eric and Andrew Roberts. 2006. Combinatory hybrid elementary analysis of text. In *Proceedings of the PASCAL Challenges Workshop on Unsupervised Segmentation of Words into Morphemes*, Trento, Italy, April.

Baayen, R. Harald, Richard Piepenbrock, and Léon Gulikers. 1995. *The CELEX lexical database (CD-ROM).* Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA, USA.

Banarjee, Satanjeev and Ted Pedersen. 2002. An adapted lesk algorithm for word sense disambiguation using WordNet. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, Mexico City, Mexico, February.

Banko, Michele and Eric Brill. 2001. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39. Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 26–33, Toulouse, France, March.

Baroni, Marco and Sabrina Bisi. 2004. Using cooccurrence statistics and the web to discover synonyms in a technical language. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, Lissabon, Portugal, May.

Barth, Michael. 2004. Extraktion von Textelementen mittels "spreading activation" für indikative Textzusammenfassungen. Master's thesis, University of Leipzig, Leipzig, Germany.

Barthes, Roland. 1983. *Elemente der Semiologie.* Suhrkamp, Paris, France.

Bensch, Peter A. and Walter J. Savitch. 1992. An occurrence-based model of word categorization. In *Presented at the 3rd Meeting on Mathematics of Language (MOL3)*, Austin, TX, USA, November.

Benson, Morton, Evelyn Benson, and Robert Ilson. 1986. *The BBI Combinatory Dictionary of English: A Guide to Word Combinations.* John Benjamins, Amsterdam, Netherlands.

Berko, Jean. 1958. The child's learning of English morphology. *Word*, 14:150–177.

Berland, M. and Eugene Charniak. 1999. Finding parts in very large corpora. In *Proceedings of the 37. Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 57–64, College Park, MD, USA, June.

Bernhard, Delphine. 2006. Unsupervised morphological segmentation based on segment predictability and word segments alignment. In *Proceedings of the PASCAL Challenges Workshop on Unsupervised Segmentation of Words into Morphemes*, April.

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

Biemann, Chris. 2005b. Semantic indexing with typed terms using rapid annotation. In *Proceedings of the Workshop on Methods and Applications of Semantic Indexing at the TKE*, Copenhagen, Denmark, August.

Biemann, Chris. 2006a. Chinese whispers - an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the Workshop on Textgraphs at the HLT/NAACL*, New York City, NY, USA, June.

Biemann, Chris. 2006b. Unsupervised part-of-speech tagging employing efficient graph clustering. In *Proceedings of the Student Research Workshop at the COLING/ACL*, Sydney, Australia, July.

Biemann, Chris, Karsten Böhm, Gerhard Heyer, and Ronny Melz. 2004. SemanticTalk: Software for visualizing brainstorming sessions and thematic concept trails on document collections. In *Proceedings of The 15th European Conference on Machine Learning (ECML) and the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), LNAI 3202*, Pisa, Italy, September. Springer.

Biemann, Chris, Stefan Bordag, and Uwe Quasthoff. 2004. Automatic acquisition of paradigmatic relations using iterated co-occurrences. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, May.

Biemann, Chris and Rainer Osswald. 2004. Automatic extension of feature-based semantic lexicons via contextual features. In *Proceedings of the 29th Annual Conference of the German Classification Society (GfKl)*. Springer, March.

Biemann, Chris, Sa-Im Shin, and Key-Sun Choi. 2004. Semiautomatic extension of corenet using a bootstrapping mechanism on corpus-based co-occurrences. In *Proceedings of The 20th International Conference on Computational Linguistics (COLING)*, Geneva, Switzerland, August.

Biemann, Chris and Sven Teresniak. 2005. Disentangling from babylonian confusion - unsupervized language identification. In *Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing), LNCS 3406*, Mexico City, Mexico, February. Springer.

Biemann, Christian, Stefan Bordag, Gerhard Heyer, Uwe Quasthoff, and Christian Wolff. 2004. Language-independent methods for compiling monolingual lexical data. In *Proceedings of the Fifth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, pages 215–228. Springer, February.

Biemann, Christian, Uwe Quasthoff, Karsten Böhm, and Christian Wolff. 2003. Automatic discovery and aggregation of compound names for the use in knowledge representations. In *Proceedings of International Conference on Knowledge Management 2003*, pages 530–541, Graz, Austria, June. Journal of Universal Computer Science, Volume 9, Number 6.

Bisson, Gilles, Claire Nédellec, and Dolores Cañamero. 2000. Designing clustering methods for ontology building: The mok workbench. In *Proceedings of Ontology Learning Workshop at the ECAI-2000*, Berlin, Germany, August.

Bordag, Stefan. 2003. Sentence co-occurrences as small-world-graphs: A solution to automatic lexical disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing), LNCS 2588*, pages 329–333. Springer, February.

Bordag, Stefan. 2005. Unsupervised knowledge-free morpheme boundary detection. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, Borovets, Bulgaria, September.

Bordag, Stefan. 2006a. Two-step approach to unsupervised morpheme segmentation. In *Proceedings of the PASCAL Challenges Workshop on Unsupervised Segmentation of Words into Morphemes*, Venice, Italy, April.

Bordag, Stefan. 2006b. Word sense induction: Triplet-based clustering and automatic evaluation. In *Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Trento, Italy, April.

Bordag, Stefan and Gerhard Heyer, 2006. *A Structuralist Framework for Quantitative Linguistics*, volume 209 of *Studies in Fuzziness and Soft Computing*, chapter Part III - Quantitative Linguistic Modeling, pages 171–189. Springer, Berlin / Heidelberg, Germany.

Bordag, Stefan, Hans Friedrich Witschel, and Thomas Wittig. 2005. Evaluation of lexical acquisition algorithms. In W. Hess und W. Lenders, editor, *Proceedings of GLDV-Frühjahrstagung 2005*, Bonn, Germany, March. Peter-Lang-Verlag.

Brants, Thorsten. 2000. TnT — a statistical part-of-speech tagger. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP)*, April-May.

Brent, Michael, Sreerama K. Murthy, and Andrew Lundberg. 1995. Discovering morphemic suffixes: A case study in MDL induction. In *Proceedings of the 5th International Workshop on Artificial Intelligence and Statistics*, Ft. Lauderdale, Florida, USA, January.

Brill, Eric. 1992. A simple rule-based part-of-speech tagger. In *Proceedings of the third conference on Applied natural language processing*, pages 152–155, Trento, Italy, March-April. Association for Computational Linguistics (ACL).

Brill, Eric. 2003. Processing natural language without natural language processing. In Alexander Gelbukh, editor, *Proceedings of the 4th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, pages 360–369. Springer, February.

Brown, Peter F., Peter V. de Souza, Robert L. Mercer, T. J. Watson, Vincent J. Della Pietra, and Jennifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Büchler, Marco. 2006. Flexible computing of co-occurrences on structered and unstructered text. Master's thesis, University of Leipzig, Leipzig, Germany.

Burgess, Curt and Kevin Lund. 1997. Modelling parsing constraints with high-dimensional context space. *Language and Cognitive Processes*, 12:177–210.

Burnard, Lou. 1995. *Users reference guide for the British National Corpus.* Oxford: OUCS.

Caraballo, Sharon A. 1999. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 120–126, June.

Carroll, John B. 1967. On sampling from a lognormal model of word-frequency distribution. In H. Kucera and W.N. Francis, editor, *Computational Analysis of Present-Day American English.* Brown University Press, Providence, RI, USA, pages 406–424.

Chanod, Jean-Pierre and Pasi Tapanainen. 1995. Tagging french - comparing a statistical and a constraintbased method. In *Proceedings of 7th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 149–156, Dublin, Ireland, March. Association for Computational Linguistics (ACL).

Chomsky, Noam. 1957. *Syntactic Structures.* Mouton.

Chomsky, Noam. 1959. Review of skinners verbal behavior. *Journal of the Experimental Analysis of Behavior*, 13:83–99.

Choueka, Yaacov, Shmuel T. Klein, and E. Neuwitz. 1983. Automatic retrieval of frequent idiomatic and collocational expressions in a large corpus. *Journal of the Association for Literary and Linguistic Computing*, 4(1):34–38.

Church, Kenneth Ward and William A. Gale. 1995. Poisson mixtures. *Natural Language Engineering*, 1:163–190.

Church, Kenneth Ward, William A. Gale, Patrick Hanks, and Donald Hindle. 1989. Parsing, word associations and typical predicate-argument relations. In *International Workshop on Parsing Technologies.* CMU.

Church, Kenneth Ward, William A. Gale, Patrick Hanks, and Donald Hindle. 1991. Using statistics in lexical analysis. In Uri Zernik, editor, *Lexical Acquisition: Exploiting On-Line Resources to Build up a Lexicon.* Lawrence Erlbaum, Hillsdale, NJ, USA, pages 115–164.

Church, Kenneth Ward and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16:22–29.

Church, Kenneth Ward and Robert L. Mercer. 1994. Introduction to the special issue on computational linguistics using large corpora. In Susan Armstrong, editor, *Using large corpora.* The MIT Press, Cambrigde, MA, USA, pages 1–24.

Ciaramita, Massimiliano and Marco Baroni. 2006. A figure of merit for the evaluation of web-corpus randomness. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 217–224. Association for Computational Linguistics (ACL), April.

Ciaramita, Massimiliano, Aldo Gangemi, Esther Ratsch, Jasmin Sarié, and Isabel Rojas. 2005. Unsupervised learning of semantic relations between concepts of a molecular biology ontology. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, Edinburgh, Scotland, UK, July-August.

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

Clark, Alexander. 2003. Combining distributional and morphological information for part of speech induction. In *Proceedings of 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Budapest, Hungary, April.

Cohen-Sygal, Yael and Shuly Wintner. 2006. Finite-state registered automata for non-concatenative morphology. *Computational Linguistics*, 32(1):49–82.

Cooper, Richard. 1996. Head-driven phrase structure grammar. In K. Brown and J. Miller, editor, *Concise Encyclopedia of Syntactic Theories.* Pergamon, Oxford, UK, pages 191–196.

Cover, Thomas M. and Joy A. Thomas. 1991. *Elements of Information Theory.* Wiley Series in Telecommunications. Wiley-Interscience, New York City, NY, USA.

Creutz, Mathias. 2003. Unsupervised segmentation of words using prior distributions of morph length and frequency. In *Proceedings of the 41. Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 280–287, Sapporo, Japan, July.

Creutz, Mathias and Krista Lagus. 2005. Unsupervised morpheme segmentation and morphology induction from text corpora using morfessor 1.0. In *Publications in Computer and Information Science, Report A81*, Helsinki, Finland, March. Helsinki University of Technology.

Creutz, Mathias and Krista Lagus. 2006. Morfessor in the morpho challenge. In *Proceedings of the PASCAL Challenges Workshop on Unsupervised Segmentation of Words into Morphemes*, Trento, Italy, April.

Cucerzan, Silviu and David Yarowsky. 2003. Minimally supervised induction of grammatical gender. In *Proceedings of the Human Language Technology Conference (HLT) of the NAACL*, pages 40–47, Edmonton, Canada, May.

Curran, James Richard. 2003. *From Distributional to Semantic Similarity.* Ph.D. thesis, Institute for Communicating and Collaborative Systems, School of Informatics. University of Edinburgh, Edinburgh, Scotland, UK.

Cutting, Douglas R., Julian Kupiec, Jan Pedersen, and Penelope Sibun. 1992. A practical part-of-speech tagger. In *Proceedings of the 3rd Applied Natural Language Processing Conference (ANLP)*, pages 133–140, Trento, Italy, March-April.

Dagan, Ido and Kenneth Ward Church. 1994. Termight: Identifying and translation technical terminology. In *Proceedings of the 4th Conference on Applied Natural Language Processing (ANLP)*, pages 34–40, Stuttgart, Germany, October.

Dagan, Ido, Lillian Lee, and Fernando Pereira. 1997. Similarity-based methods for word-sense disambiguation. In *Proceedings of the 35th annual meeting of the Association for Computational Linguistics (ACL) and the 8th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 56–63, Madrid, Spain, July.

Dagan, Ido, Lillian Lee, and Fernando C. N. Pereira. 1999. Similarity based models of word cooccurrence probabilities. *Machine Learning*, 34(1-3):43–69.

Dagan, Ido, Shaul Marcus, and Shaul Markovitch. 1995. Contextual word similarity and estimation from sparse data. *Computer Speech and Language*, 9:123–152.

Dang, Minh Thang and Saad Choudri. 2006. Simple unsupervised morphology analysis algorithm (sumaa). In *Proceedings of the PASCAL Challenges Workshop on Unsupervised Segmentation of Words into Morphemes*, Trento, Italy, April.

de Marcken, Carl. 1995. The unsupervised acquisition of a lexicon from continuous speech. Memo 1558, MIT Artificial Intelligence Lab.

de Saussure, Ferdinand. 2001. *Grundfragen der allgemeinen Sprachwissenschaft.* de Gruyter, 3rd edition. C. Bally and A. Sechehaye (eds.).

Debusmann, Ralph, Denys Duchier, and Geert-Jan M. Kruijff. 2004. Extensible dependency grammar: A new methodology. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING) Workshop on Recent Advances in Dependency Grammar*, Geneva, Switzerland, August.

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

Deese, James. 1959. On the prediction of occurrence of particular verbal intrusions in immediate recall. *Journal of Experimental Psychology*, 58:17–22.

Déjean, Hervé. 1998. Morphemes as necessary concept for structures discovery from untagged corpora. In D.M.W. Powers, editor, *Workshop on Paradigms and Grounding in Natural Language Learning at NeMLaP3/CoNLL98*, pages 295–299, Adelaide, Australia, January.

D'ejean, Hervé, Eric Gaussier, Jean-Michel Renders, and Fatia Sadat. 2005. Automatic processing of multilingual medical terminology: applications to thesaurus enrichment and cross-language information retrieval. *Artificial Intelligence in Medicine*, 33(2):111–24.

Der, Ralf. 2001. Self-organized acqusition of situated behavior. *Theory Bioscience*, 120:1–9.

Dice, Lee R. 1945. Measures of the amount of ecologic association between species. *Journal of Ecology*, 26(3):297–302.

Dorow, Beate and Dominic Widdows. 2003. Discovering corpus-specific word senses. In *Proceedings of 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 79–82, Budapest, Hungary, April.

Dowty, David R. 1979. *Word Meaning in Montague Grammar*. D. Reidel Publishing Company, Dordrecht, Holland.

Doyle, Arthur Conan. 1902. The hound of the baskervilles.

Dumais, Susan T. 1995. Latent semantic indexing (LSI). In D. K. Harman, editor, *Overview of the Third Text Retrieval Conference (TREC)*, pages 219–230, Gaithersburg, MD, USA, November. National Institute of Standards and Technology.

Dunning, Ted E. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

Evert, Stefan. 2004. *The Statistics of Word Cooccurrences: Word Pairs and Collocations*. Ph.D. thesis, University of Stuttgart, Stuttgart, Germany.

Evert, Stefan and Brigitte Krenn. 2001. Methods for the qualitative evaluation of lexical association measures. In *Proceedings of the 39th Annual Meeting of* *the Association for Computational Linguistics (ACL)*, pages 188–195, Toulouse, France.

Faure, David and Claire Nédellec. 1998. Asium: Learning subcategorization frames and restrictions of selection. In *Proceedings of the 10th Conference on Machine Learning (ECML 98): Workshop on Text Mining*, Chemitz, Germany, July.

Fellbaum, Christiane. 1998. A semantic network of English: The mother of all WordNets. *Computers and the Humanities*, 32:209–220.

Feng, Haodi, Kang Chen, Chunyu Kit, and Xiaotie Deng. 2004. Unsupervised segmentation of chinese corpus using accessor variety. In *Proceedings of IJCNLP 2004*, pages 694–703, Hainan Island, China, March. Springer.

Ferrer i Cancho, Ramon and Ricard V. Sole. 2001. The small world of human language. In *Proceedings of The Royal Society of London. Series B, Biological Sciences.*, pages 268(1482): 2261–2265.

Ferret, Olivier. 2004. Discovering word senses from a network of lexical cooccurrences. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 1326–1332, Geneva, Switzerland, August.

Finch, Steven Paul. 1993. *Finding Structure in Language*. Ph.D. thesis, University of Edinburgh, Edinburgh, Scotland, UK.

Firth, John R. 1957. *A synopsis of linguistic theory 1930-1955.* Oxford: Philological Society., reprinted in f. r. palmer (ed), selected papers of j. r. firth 1952-1959, london: longman, 1968 edition.

Frakes, William R., 1992. *Stemming Algorithms*, chapter 8, pages 131–160. Frakes und Baeza-Yates.

Frakes, William R. and Ricardo Baeza-Yates. 1992. *Information Retrieval: Data Structures and Algorithms.* Prentice-Hall Inc., Englewood Cliffs, NJ, USA.

Fredkin, Edward. 1960. Trie memory. *Communications of the ACM*, 3(9):490–499, September.

Freitag, Dayne. 2004. Toward unsupervised whole-corpus tagging. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, Geneva, Switzerland, August.

Freitag, Dayne. 2005. Morphology induction from term clusters. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL)*, pages 128–135, Ann Arbor, MI, USA, June.

Gale, William, Kenneth Ward Church, and David Yarowsky. 1992. Work on statistical methods for word sense disambiguation. *Intelligent Probabilistic Approaches to Natural Language*, Fall Symposium Series(FS-92-04):54–60, March.

Gauch, Susan and Robert P. Futrelle. 1994. Experiments in automatic word class and word sense identification for information retrieval. In *Proceedings of 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 425–434, Las Vegas, Nevada, April.

Gaustad, Tanja. 2001. Statistical corpus-based word sense disambiguation: Pseudowords vs. real ambiguous words. In *Companion Volume to the Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, pages 61–66, Toulouse, France, July.

Gerlach, Rainer. 1982. Zur ueberpruefung des menzerath'schen gesetzes im bereich der morphologie. In W. Lehfeldt and U. Strauss, editors, *Glottometrika 4*. Brockmeyer, Bochum, Germany, pages 95–102.

Goldsmith, John. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198.

Grefenstette, Gregory. 1992. Finding semantic similarity in raw text: the deese antonyms. In Robert Goldman, Peter Norvig, Eugene Charniak, and Bill Gale, editors, *Working Notes of the AAAI Full Symposium on Probabilistic Approaches to Natural Language*, pages 61–65, Menlo Park, CA, USA, October. AAAI Press.

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

Grefenstette, Gregory. 1996. Evaluation techniques for automatic semantic extraction: Comparing syntactic and window-based approaches. *Corpus Processing for Lexical Acquisition*, pages 205–216.

Grewendorf, G. 1993. Parametrisierung der Syntax. Zur kognitiven Revolution in der Linguistik. In L. Hoffmann, editor, *Deutsche Syntax. Ansichten und Einsichten.* de Gruyter, pages 11–73.

Grewendorf, Günther, Fritz Hamm, and Wolfgang Sternefeld. 1989. *Sprachliches Wissen. Eine Einführung.* Suhrkamp.

Güntzer, Ulrich, Gerald Jüttner, Gerhard Seegmüller, and Frank Sarre. 1989. Automatic thesaurus construction by machine learning from retrieval sessions. *Information Processes Management*, 25(3):265–273.

Hafer, Margaret A. and Stephen F. Weiss. 1974. Word segmentation by letter successor varieties. *Information Storage and Retrieval*, 10:371–385.

Haghighi, Aria and Dan Klein. 2006. Prototype-driven learning for sequence models. In *Proceedings of the Human Language Technology Conference (HLT) of the NAACL*, pages 320–327, New York City, NY, USA, June. Association for Computational Linguistics (ACL).

Halle, Morris. 1997. Distributed morphology: Impoverishment and fission. In Benjamin Bruening, Yoonjung Kang, and Martha McGinnis, editors, *MIT Working Papers in Linguistics (MITWPL): Papers at the Interface*, pages 425–449, Cambridge, MA, USA.

Halle, Morris and Alec Marantz. 1993. Distributed morphology and the pieces of inflection. *The View from Building 20*, pages 111–176.

Halle, Morris and Alec Marantz. 1994. Some key features of distributed morphology. *MITWPL 21: Papers on phonology and morphology*, pages 275–288.

Hamp, Birgit and Helmut Feldweg. 1997. GermaNet - a lexical-semantic net for German. In *Proceedings of workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications at the ACL*, Madrid, Spain, July.

Happ, Heinz. 1985. *Paradigmatisch - syntagmatisch. Zur Bestimmung und Klärung zweier Grundbegriffe der Sprachwissenschaft.* Carl Winter Universitätsverlag., Heidelberg, Germany.

Harley, Trevor A. 1995. *The Psychology of Language.* Psychology Press Ltd.

Harris, Roy. 2003. *Saussure and his Interpreters.* Edinburgh University Press, 2 edition.

Harris, Zellig S. 1951. *Structural Linguistics.* University of Chicago Press, Chicago, IL, USA.

Harris, Zellig S. 1955. From phonemes to morphemes. *Language*, 31(2):190–222.

Harris, Zellig S. 1968. *Mathematical Structures of Language.* Wiley, New York City, NY, USA.

Hatzivassiloglou, V. and K. R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of The 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of Association for Computational Linguistics joint conference (ACL/EACL)*, pages 174–181, July.

Hearst, Marti A. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of The 14th International Conference on Computational Linguistics (COLING)*, pages 539–545, Nantes, France, August.

Heyer, Gerhard, Uwe Quasthoff, and Thomas Wittig. 2005. *Wissensrohstoff Text. Text Mining: Konzepte, Algorithmen, Ergebnisse.* W3L-Verlag, Bochum, Germany, 1 edition.

Heyer, Gerhard, Uwe Quasthoff, Thomas Wittig, and Christian Wolff. 2001. Learning relations using collocations. In A. Maedche, S. Staab, C. Nedellec, and E. Hovy, editors, *Proceedings of the Workshop on Ontology Learning at the IJCAI*, August.

Heyer, Gerhard, Uwe Quasthoff, and Christian Wolff. 2002. Knowledge extraction from text: Using filters on collocation sets. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC) and IICS.* Springer, Las Palmas, Spain, May, pages 153–162.

Heyer, Gerhard and Hans Friedrich Witschel. 2005. Terminology and metadata - on how to efficiently build an. In Hans Friedrich Witschel, editor, *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005*, Copenhagen, Denmark, August.

Hindle, Donald. 1990. Noun classification from predicate-argument structures. In *Meeting of the Association for Computational Linguistics (ACL)*, pages 268–275, Pittsburgh, PA, USA, June.

Hjelmslev, Louis. 1968. *Die Sprache: eine Einführung.* Wissenschaftliche Buchgesellschaft, Darmstadt, Germany.

Hjelmslev, Louis. 1974. *Aufsätze zur Sprachwissenschaft.* Klett, Stuttgart, Germany.

Holtsberg, Anders and Caroline Willners. 2001. Statistics for sentential co-occurrence. In *Working Papers 48*, pages 135–148.

Hřebíček, Ludek. 1989. A syntactic variable on the text level. In M. G. Boroda, editor, *Glottometrika 10.* Brockmeyer, Bochum, Germany, pages 205–218.

Jakobson, Roman O. 1956. Two aspects of language and two types of aphasic disturbances. In Roman Jakobson, editor, *Selected Writings II. Word and Language.* The Hague, pages 239–259.

Jiang, J. and D. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th International Conference on Research on Computational Linguistics (ROCLING), Taiwan*, Taipei, Taiwan, August.

Johanessen, Janne Bondi, Kristin Hagen, and Anders Nøklestad. 2000. A Constraint-based Tagger for Norwegian. In Carl-Erik og Steffen Nordahl, editor, *Proceedings of the 17th Scandinavian Conference of Linguistics. Odense Working Papers in Language and Communication 19*, pages 31–48, Odense, Denmark, August. Lindberg.

Jurafsky, D. and J. H. Martin. 2000. *Speech and Language Processing.* Prentice Hall.

Kamp, Hans. 1981. A theory of truth and semantic representation. In J. Groenendijk, Th. Janssen, and M. Stokhof, editor, *Formal Methods in the Study of Language.* Mathematisch Centrum, Amsterdam, Netherlands.

Kamp, Hans and Uwe Reyle. 1993. *From Discourse to Logic.* Kluwer, Dordrecht, Holland.

Kaplan, Ronald M. and Joan Bresnan. 1982. Lexical-functional grammar: A formal system for grammatical representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations.* The MIT Press, Cambridge, MA, USA, pages 173–281.

Karov, Yael and Shimon Edelman. 1998. Similarity-based word sense disambiguation. *Computational Linguistics*, 24:41–59.

Kazakov, Dimitar. 1997. Unsupervised learning of naïve morphology with genetic algorithms. In A. van den Bosch, W. Daelemans, and A. Weijters, editors, *Workshop Notes of the ECML/MLnet Workshop on Empirical Learning of Natural Language Processing Tasks*, pages 105–112, Prague, Czech Republic, April.

Kazakov, Dimitar. 2001. Unsupervised learning of word segmentation rules with genetic algorithms and inductive logic programming. *Machine Learning*, 43:121–162, April-May.

Keller, Frank, Maria Lapata, and Olga Ourioupina. 2002. Using the web to overcome data sparseness. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 230–237, Philadelphia, PA, USA, July.

Keshava, Samarth and Emily Pitler. 2006. A simpler, intuitive approach to morpheme induction. In *Proceedings of the PASCAL Challenges Workshop on Unsupervised Segmentation of Words into Morphemes*, Trento, Italy, April.

Kilgarriff, Adam. 1997. I don't believe in word senses. *Computers and the Humanities*, 31(2):91–113.

Kilgarriff, Adam and David Tugwell. 2001. WASP-bench: an MT lexicographers' workstation supporting state-of-the-art lexical disambiguation. In *Proceedings of Machine Translation Summit VIII*, pages 187–190, Compostela, Spain, September.

Kiparsky, Paul. 1982. Lexical phonology and morphology. *Linguistics in the Morning Calm*, 2:3–91.

Klein, Dan. 2005. *The Unsupervised Learning of Language Structure.* Ph.D. thesis, Stanford University, Stanford, CA, USA.

Köhler, Reinhard. 1983. Systemtheorie und Semiotik. *Zeitschrift für Semiotik*, 5(4):424–43.

Koskenniemi, Kimmo. 1983. *Two-level Morphology: A General Computational Model for Word-Form Recognition and Production.* University of Helsinki, Department of General Linguistics.

Krenn, Brigitte. 2000. Collocation mining: Exploiting corpora for collocation identification and representation. In *Proceedings of the Konferenz zur Verarbeitung natürlicher Sprache (KONVENS)*, Ilmenau, Germany, October.

Krenn, Brigitte and Stefan Evert. 2001. Can we do better than frequency? a case study on extracting pp-verb collocations. In *Proceedings of the Workshop on Collocations at the ACL*, pages 39–46, Toulouse, France, July.

Krieger, Hans-Ulrich and John Nerbonne. 1993. Feature-based inheritance networks for computational lexicons. In Ted Briscoe, Valerie de Paiva, and Ann Copestake, editors, *Inheritance, Defaults, and the Lexicon.* Cambridge University Press, Cambridge, MA, USA, pages 90 – 136.

Kucera, Henry and W. Nelson Francis. 1967. *Computational Analysis of Present-Day American English.* Brown University Press, Providence, RI, USA.

Kuhn, Thomas. 1962. *The Structure of Scientific Revolutions.* University of Chicago Press.

Kunze, C. and A. Wagner. 1999. Integrating GermaNet into EuroWordNet, a multilingual lexical-semantic database. *Sprache und Datenverarbeitung - International Journal for Language Data Processing.* Bonn.

Kurimo, Mikko, Mathias Creutz, Matti Varjokallio, Ebru Arisoy, and Murat Saraclar. 2006. *Unsupervised segmentation of words into morphemes - Challenge 2005 An Introduction and Evaluation Report.* Proceedings of the PASCAL Challenges Workshop on Unsupervised Segmentation of Words into Morphemes, Venice, Italy.

Kytö, Merja, 1996. *Manual to the Diachronic Part of the Helsinki Corpus of English Texts: Coding Conventions and Lists of Source Texts.* Department of English, University of Helsinki.

Landauer, Thomas K. and Susan T. Dumais. 1997. A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211–240.

Lee, Lillian. 1997. *Similarity-Based Approaches to Natural Language Processing.* Ph.D. thesis, Harvard University, Cambridge, MA, USA.

Lee, Lillian. 1999. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 25–32, College Park, MD, USA, June.

Lehr, Andrea. 1993. Kollokationsanalysen. Von der Kollokationstheorie des Kontextualismus zu einem computergestützen Verfahren. *Zeitschrift für germanistische Linguistik*, 21:2–19.

Lehr, Andrea. 1996. *Kollokationen und maschinenlesbare Korpora. Ein operatives Analysemodell zum Aufbau lexikalischer Netze.* Germanistische Linguistik 168. Niemeyer, Tübingen, Germany.

Lesk, Michael. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the Special Interest Group on Design of Communication Conference (SIGDOC)*, pages 24–26, Toronto, ON, Canada. Association for Computing Machinery (ACM).

Levenshtein, Vladimir I. 1965. Binary codes capable of correcting deletions, insertions, and reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848.

Levin, Beth. 1993. *English Verb Classes and Alternations.* University of Chicago Press.

Levin, Beth and Malka Rappoport Hovav. 1991. Wiping the slate clean: A lexical semantic exploration. *Cognition*, 41:123–151.

Levin, Beth, Grace Song, and Sue Atkins. 1997. Making sense of corpus data: a case study of verbs of sound. *International Journal of Corpus Linguistics*, 1(2):23–64.

Lieber, Rochelle. 1990. *On the organization of the lexicon.* Garland Publishing, Inc., New York City, NY, USA.

Lin, Dekang. 1998a. Automatic retrieval and clustering of similar words. In *Proceedings of The 17th International Conference on Computational Linguistics (COLING/ACL)*, pages 768–774, August.

Lin, Dekang. 1998b. Extracting collocations from text corpora. In *Proceedings of the First Workshop on Computational Terminology.*

Maedche, Alexander and Steffen Staab. 2004. Ontology learning. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*. Springer, pages 173–190.

Mann, William C. and Sandra A. Thompson. 1987. Rhetorical structure theory: A theory of text organization. Technical report isi/rs-87-190, Information Sciences Institute, University of Southern California.

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

Marcu, D. 2000. *The Theory and Practice of Discourse Parsing and Summarization.* The MIT Press.

Markov, Andrei A. 1913. An example of statistical investigation in the text of 'Eugene Onyegin' illustrating coupling of 'tests' in chains. *Proceedings of the Academy of Sciences*, 7 of VI:153–162.

Martinet, André. 1969. *Grundzüge der Allgemeinen Sprachwissenschaft.* Kohlhammer Verlag Stuttgart.

Matsumura, Naohiro, Yukio Ohsawa, and Mitsuru Ishizuka. 2003. PAI: automatic indexing for extracting asserted keywords from a document. *New Generation Computing*, 21(1):37–47, February.

Matthews, Peter. 1974. *Morphology: An introduction to the theory of word-structure.* Cambridge University Press.

McClelland, James L. and David E. Rumelhart. 1986. A distributed model of human learning and memory. In *Parallel Distributed Processing: Volume 2: Psychological and Biological Models.* The MIT Press, Cambridge, MA, USA, pages 170–215.

Menzerath, Paul. 1954. *Die Architektonik des deutchen Wortschatzes.* Dummler, Bonn, Germany.

Miller, George A. 1990. WordNet: An online lexical database. *International Journal of Lexicography*, 4(3):235–312.

Mitra, Mandar, Amit Singhal, and Chris Buckley. 1997. Automatic text summarization by paragraph extraction. In *Proceedings of the Workshop on Intelligent*

Scalable Text Summarization at the ACL/EACL, pages 39–46, Madrid, Spain, July. Association for Computational Linguistics (ACL).

Moldovan, Dan, Adriana Badulescu, Marta Tatu, Daniel Antohe, and Roxana Girju. 2004. Models for the semantic classification of noun phrases. In *Proceedings of the Computational Lexical Semantics Workshop at HLT/NAACL*, pages 60–67, Boston, MA, USA, May.

Moore, Robert C. 2004. On log-likelihood-ratios and the significance of rare events. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Barcelona, Spain, July. Association for Computational Linguistics (ACL).

Morrison, David R. 1968. Patricia - practical algorithm to retrieve information coded in alphanumeric. *Journal of the ACM*, 15(4):514–534, October.

Nakov, Preslav I. and Marti A. Hearst. 2003. Category-based pseudowords. In *Companion Volume to the Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics.*, pages 70–72, Edmonton, Alberta, Canada, May-June.

Neill, Daniel B. 2002. Fully automatic word sense induction by semantic clustering. Master's thesis, Cambridge University, Cambridge, UK.

Oliva, Karel and Pavel Kveton. 2002. Corpus representativity, bigrams, and pos-tagging quality. Tr-2002-26, Österreichisches Forschungsinstitut für Artificial Intelligence, Wien, Austria.

Pantel, P. and Dekang Lin. 2000. Word-for-word glossing with contextually similar words. In *Proceedings of the 1st Annual Meeting of the North American Chapter of Association for Computational Linguistics (NAACL)*, pages 78–85, Seattle, WA, USA, April-May.

Pantel, Patrick and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD)*, pages 613–619, Edmonton, Alberta, Canada, July.

Pantel, Patrick, Deepak Ravichandran, and Eduard Hovy. 2004. Towards terascale knowledge acquisition. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, Geneva, Switzerland, August.

Pedersen, Ted and Rebecca Bruce. 1997. Distinguishing word senses in untagged text. In *Proceedings of the EMNLP-2*, pages 197–207, Providence, RI, USA, August.

Peirce, Charles Sanderson. 1986. *Semiotische Schriften. Bd. 1. 1865-1903.* Suhrkamp, Frankfurt am Main, Germany.

Pereira, Fernando, Naftali Z. Tishby, and Lillian Lee. 1993. Distributional clustering of English words. In *30th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 183–190, Columbus, OH, USA, June. Association for Computational Linguistics (ACL).

Pinker, Steven. 1994. *The Language Instinct: the new science of language and mind.* William Morrow, New York City, NY, USA.

Pollard, Carl and Ivan A. Sag. 1994. *Head-driven Phrase Structure Grammar.* University of Chicago Press and CSLI Publications, Chicago, IL, USA.

Popper, Karl. 1959. *The Logic of Scientific Discovery.* Routledge.

Purandare, Amruta. 2004. Word sense discrimination by clustering similarity contexts. Master's thesis, University of Minnesota, Duluth, MN, USA.

Quasthoff, Uwe and Christian Wolff. 2002. The poisson collocation measure and its applications. In *Second International Workshop on Computational Approaches to Collocations*, Vienna, Austria, July.

Quasthoff, Uwe. 1998. Projekt: Der Deutsche Wortschatz. In Gerhard Heyer and Christian Wolff, editors, *Tagungsband zur GLDV-Tagung*, pages 93–99, Leipzig, Germany, March. Deutscher Universitätsverlag.

Quasthoff, Uwe, Matthias Richter, and Christian Biemann. 2006. Corpus portal for search in monolingual corpora. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 1799–1802, Genoa, Italy, May.

Quirk, Randolph, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language.* Addison Wesley Publishing Company.

Rapp, Reinhard. 1996. *Die Berechnung von Assoziationen.* Hildesheim, Olms.

Rapp, Reinhard. 2002. The computation of word associations. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, Taipei, Taiwan, August-September.

Rapp, Reinhard. 2004. Mining text for word senses using independent component analysis. In *Proceedings of SIAM International Conference on Data Mining*, Lake Buena Vista, FL, USA, April.

Rapp, Reinhard. 2005a. On the relationship between word frequency and word familiarity. In Bernhard Fisseni, Hans-Christian Schmitz, Bernhard Schröder, and Petra Wagner, editors, *Proceedings of GLDV-conference*, pages 249–263, Bonn, Germany, March-April. Peter Lang.

Rapp, Reinhard. 2005b. A practical solution to the problem of automatic part-of-speech induction from text. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 77–80, Ann Arbor, MI, USA, June. ACL.

Resnik, Philip Stuart. 1993. *Selection and Information: A Class-Based Approach to Lexical Relationships.* Ph.D. thesis, University of Pennsylvania, Philadelphia, PA, USA.

Resnik, Philip Stuart. 1998. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130.

Richardson, Stephen D. 1997. *Determining Similarity and Inferring Relations in a Lexical Knowledge Base.* Ph.D. thesis, The City University of New York, New York City, NY, USA.

Richter, Matthias. 2005. Analysis and visualization for daily newspaper corpora. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, Borovets, Bulgaria, September. Incoma.

Rieger, Burghard. 1979. Repräsentativität: von der Unangemessenheit eines Begriffs zur Kennzeichnung eines Problems linguistischer Korpusbildung. In *Empirische Textwissenschaft. Aufbau und Auswertung von Text-Corpora [Monographien Linguistik und Kommunikationswissenschaft 39]. Bergenholtz, H./ Schaeder, B. (eds.).* Königstein / Taunus (Scriptor), pages 52–70.

Rieger, Burghard. 1989. *Unscharfe Semantik. Die empirische Analyse, quantitative Beschreibung, formale Repräsentation und prozedurale Modellierung vager Wortbedeutungen in Texten.* Peter Lang, Bern / Frankfurt / New York / Paris.

Rieger, Burghard. 1991. Distributed semantic representations of word meanings. In *Proceedings of the Workshop on Evolutionary Models and Strategies / Workshop on Parallel Processing: WOPPLOT 89. Becker, J.D./ Eisele, I./ Mündemann, F.W. (eds.)*, Parallelism, Learning, Evolution., pages 243–273. Springer, July.

Rieger, Burghard. 1995. Situations, language games, and SCIPS. modeling semiotic cognitive information processing systems. In *Proceedings of the ISIC-Workshop, Monterey: the 10th International IEEE-Symposium on Intelligent Control. Albus, J./ Meystel, A./ Pospelov, D./ Reader, T. (eds.)*, Architectures for Semiotic Modeling and Situation Analysis in Large Complex Systems, pages 130–138. Bala Cynwyd, PA. (AdRem Inc.).

Riloff, Ellen and Jessica Shepherd. 1997. A corpus-based approach for building semantic lexicons. In C. Cardie and R. Weischedel, editors, *Proceedings of* *the Second Conference on Empirical Methods in Natural Language Processing (EMNLP 1997)*, pages 117–124, Somerset, NJ, USA, August. Association for Computational Linguistics (ACL).

Roark, Brian and Eugene Charniak. 1998. Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. In *Proceedings of The 17th International Conference on Computational Linguistics (COLING/ACL)*, pages 1110–1116, Montreal, Quebec, Canada, August.

Roget, P. M. 1946. *Roget's International Thesaurus*. Thomas Y. Crowell, New York City, NY, USA.

Rohwer, Richard and Dayne Freitag. 2004. Towards full automation of lexicon construction. In *Proceedings of Computational Lexical Semantics Workshop at the HLT/NAACL*, Boston, MA, USA, May.

Rothe, Ursula. 1983. Wortlänge und Bedeutungsmenge. Eine Untersuchung zum Menzerathschen Gesetz an drei romanischen Sprachen. In R. Köhler and J. Boy, editors, *Glottometrika 5*. Brockmeyer, Bochum, Germany, pages 101–112.

Ruge, Gerda. 1997. Automatic detection of thesaurus relations for information retrieval applications. In C. Freksa, M. Jantzen, and R. Valk, editors, *Foundations of Computer Science: Potential - Theory - Cognition*, pages 499–506, Heidelberg, Germany. Springer.

Sahlgren, Magnus. 2006. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Ph.D. thesis, Swedish Intitute of Computer Science, Stockholm, Sweden.

Salton, Gerard, Amit Singhal, Mandar Mitra, , and Chris Buckley. 1997. Automatic text structuring and summarization. *Information Processing and Management*, 33(2):193–207.

Sampson, Geoffrey. 2005. *The 'Language Instinct' Debate*. Continuum, London, UK.

Sánchez, A. and P. Cantos. 1997. Predictability of word forms (types) and lemmas in linguistic corpora. a case study based on the analysis of the cumbre corpus: An 8-million-word corpus of contemporary spanish. *International Journal of Corpus Linguistics*, 2(2):259–280.

Sanderson, Mark. 1994. Word sense disambiguation and information retrieval. In *Proceedings of the 17th ACM Special Interest Group on Information Retrieval Conference (SIGIR)*, pages 142–151, Dublin, Ireland, July. Springer-Verlag New York, Inc.

Schiller, A., S. Teufel, and C. Thielen. 1995. Guidelines für das Taggen deutscher Textcorpora mit STTS. Technical report, IMS-CL, Univ. Stuttgart and SfS, Universität Tübingen, Germany.

Schone, Patrick and Daniel Jurafsky. 2001a. Knowledge-free induction of inflectional morphologies. In *Proceedings of the 2nd Annual Meeting of the North American Chapter of Association for Computational Linguistics*, Pittsburgh, PA, USA, June.

Schone, Patrick and Daniel Jurafsky. 2001b. Language-independent induction of part of speech class labels using only language universals. In *Workshop at the IJCAI-2001*, Seattle, WA, USA, August. Machine Learning: Beyond Supervision.

Schütze, Hinrich. 1992a. Context space. In *Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, pages 113–120, Menlo Park, CA, USA, October. AAAI Press.

Schütze, Hinrich. 1992b. Dimensions of meaning. In *Proceedings of the 1992 conference on Supercomputing*, pages 787–796, Minneapolis, MN, USA, November.

Schütze, Hinrich. 1995. Distributional part-of-speech tagging. In *Proceedings of 7th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 141–148, Dublin, Ireland, March.

Schütze, Hinrich. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24:97–124.

Schvaneveldt, Roger. 1990. *Pathfinder Associative Networks: Studies in Knowledge Organization.* Ablex.

Sciullo, Anna-Maria Di and Edwin Williams. 1987. *On the definition of word.* The MIT Press, Cambridge, MA, USA.

Sejnowksi, Terrence J. and Charles R. Rosenberg. 1987. Parallel networks that learn to pronounce english text. *Complex Systems*, pages 145–168.

Selkirk, Elizabeth. 1982. *The syntax of words.* The MIT Press, Cambridge, MA, USA.

SENSEVAL 2. 2001. *Second International Workshop on Evaluating Word Sense Disambiguation Systems.* Toulouse, France, July.

Seretan, Maria-Violeta. 2003. *Syntactic and Semantic Oriented Corpus Investigation for Collocation Extraction, Translation and Generation.* Ph.D. thesis, University of Geneva, Geneva, Switzerland.

Shaikevich, Anatole Y. 1985. Automatic construction of a thesaurus from explanatory dictionaries. *Automatic Documentation and Mathematical Linguistics*, 19:76–89.

Sjöberg, Jonas and Viggo Kann. 2004. Automatic indexing based on Bayesian inference networks. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, Lisbon, Portugal, May.

Smadja, Frank. 1989. Macrocoding the lexicon with co-occurrence knowledge. In U. Zernik, editor, *Proceedings of the First International Lexical Acquisition Workshop*, Detroit, MI, USA, August.

Smadja, Frank. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):43–177.

Smadja, Frank, Kathleen R. McKeown, and Vasileios Hatzivassiloglou. 1996. Translating collocations for bilingual lexicons: A statistical approach. *Computational Linguistics*, 22:1–38.

Sproat, Richard. 1992. *Morphology and Computation.* The MIT Press, Cambridge, MA, USA.

Steyvers, Mark and Josh Tenenbaum. 2005. The large scale structure of semantic networks: Statistical analyses and a model of semantic growth. *Cognitive Science*, 29(1):41–78.

Svartvik, Jan. 1990. *The London Corpus of Spoken English: Description and Research.* Lund Studies in English 82. Lund University Press.

Tager-Flusberg, Helen. 1997. Putting words together: Morphology and syntax in the preschool years. In J. Berko Gleason, editor, *The development of language.* Allyn and Bacon, Boston, MA, USA, pages 159–209.

Tamir, Raz and Reinhard Rapp. 2003. Mining the web to discover the meanings of an ambiguous word. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM)*, pages 645–648.

Tan, Pang-Ning, Vipin Kumar, and Jaideep Srivastava. 2002. Selecting the right interestingness measure for association patterns. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 32–41, Melbourne, FL, USA, December.

Tanimoto, T.T. 1958. An element mathematical theory of classification. Technical report, I.B.M. Research, New York City, NY, USA, November.

Terra, Egidio and C. L. A. Clarke. 2003. Frequency estimates for statistical word similarity measures. In *Proceedings of the Human Language Technology Conference (HLT) of the NAACL*, pages 165–172, Edmonton, Canada, May-June.

Trubetzkoy, Nikolai S. 1939. *Grundzüge der Phonologie (Travaux du Cercle Linguistique de Prague. 7)*. Prague, Czech Republic.

Turney, Peter D. 2001. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the 12th European Conference on Machine Learning (ECML)*, pages 491–502, Freiburg, Germany, September.

Turney, Peter D. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 417–424, July.

Udani, Goldee, Shachi Dave, Anthony Davis, and Tim Sibley. 2005. Noun sense induction using web search results. In *Proceedings of 28th ACM Special Interest Group in Information Retrieval (SIGIR)*, pages 657–658, Salvador, Brazil, August.

Velldal, Erik. 2005. A fuzzy clustering approach to word sense discrimination. In *Proceedings of the 7th International conference on Terminology and Knowledge Engineering (TKE)*, Copenhagen, Denmark, August.

Vihman, Marilyn May. 1982. The acquisition of morphology by a bilingual child: A whole-word approach. *Applied Psycholinguistics*, 3:141–160.

Vossen, P. 1998. Introduction to EuroWordNet. *Special Issue on EuroWordNet of Computers and the Humanities*, 32(2-3):73–89.

Weeds, Julie and David Weir. 2005. Co-occurrence retrieval: A flexible framework for lexical distributional similarity. *Computational Linguistics*, pages 439–475, December.

Weeds, Julie, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of the 20th International Conference of Computational Linguistics (COLING)*, Geneva, Switzerland, August.

Weissenborn, Jürgen and Barbara Höhle. 2001. *Approaches to Bootstrapping - Phonological, lexical, syntactic and neurophysiological aspects of early language acquisition.* John Benjamins Publishing, Philadelhpia, PA, USA.

Widdows, Dominic. 2003. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *Proceedings of the Human Language Technology Conference (HLT) of the NAACL*, pages 276–283, Edmonton, Canada, May-June.

Widdows, Dominic and Beate Dorow. 2002. A graph model for unsupervised lexical acquisition. In *Proceedings of 19th International Conference on Computational Linguistics*, pages 1093–1099, Taipei, Taiwan, August.

Witschel, Hans Friedrich. 2004. Terminologie-Extraktion: Möglichkeiten der Kombination statistischer und musterbasierter Verfahren. In *Content and Communication: Terminology, Language Resources and Semantic Interoperability*. Ergon Verlag, Würzburg.

Witschel, Hans Friedrich. 2005. Using decision trees and text mining techniques for extending taxonomies. In *Proc. of Learning and Extending Lexical Ontologies by using Machine Learning Methods, Workshop at the ICML*, Bonn, Germany, August.

Witschel, Hans Friedrich and Chris Biemann. 2005. Rigorous dimensionality reduction through linguistically motivated feature selection for text categorisation. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA)*, Joensuu, Finland, May.

Witten, Ian H., Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Kea: Practical automatic keyphrase extraction. In *Proceedings of The Fourth ACM Digital Libraries Conference (DL)*, pages 254–256, Berkeley, CA, USA, August.

Yarowski, David. 1995. Unsupervised word sense disambiguation rivaling supervised methods. *Association for Computational Linguistics (ACL)*, 33:189–196.

Zipf, George Kingsley. 1949. *Human Behaviour and the Principle of Least-Effort.* Addison-Wesley, Cambridge MA edition.

Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation selbständig und ohne zulässige fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialien oder erbrachten Dienstleistungen als solche gekennzeichnet.

Leipzig, den

Stefan Bordag

Wissenschaftlicher Werdegang

1997 Immatrikulation an der Universitt Leipzig für das Fach Diplom Informatik mit Nebenfach Linguistik

1999 Vordiplom im Fach Diplom Informatik mit Nebenfach Linguistik

2000 Einsemestriges Auslandsstudium an der University of Arizona, in Tucson, USA

2001 Praktikum und anschließende Anstellung bei der Firma Bios s.r.o. in Prag, Tschechische Republik

2002 Abschluss Studium Diplom Informatik mit Nebenfach Linguistik an der Universität Leipzig

seit 2003 Mitarbeiter der Universität Leipzig, Institut für Informatik, Abteilung Automatische Sprachverarbeitung

Teilnahme an Konferenzen und Wettbewerben, unter anderem

- 4th International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City (CICLing 2003)
- Text, Speech and Dialogue 2003, České Budějovice (TSD 2003)
- Recent Advances in Natural Language Processing, Borovets (RANLP 2005)
- Unsupervised Segmentation of Words into Morphemes – Challenge 2005 (EU Network of Excellence PASCAL Challenge Program), Venecia
- 11th Conference of the European Chapter of the Association for Computational Linguistics, Trento (EACL 2006)

Lehre in den folgenden Fächern (Seminare, Praktika oder vorlesungsbegleitend):

- Algorithmen und Datenstrukturen I & II
- Text Mining
- Information Retrieval I & II

- Textdaten und Korpuslinguistik
- Modelle und Verfahren in der Computerlinguistik
- Automatische Akquisition linguistischen Wissens
- Semantische Analyse im Internet