Algorithms extracting linguistic relations and their evaluation

Stefan Bordag* University of Leipzig

Numerous unsupervised algorithms have been introduced with the goal of extracting knowledge about relations between words. The foundations of these are co-occurrence statistics such as mutual information or log-likelihood quotient, as well as comparison operators such as dice coefficient or euclidean distance. The aim of this study is to describe the elementary methods used throughout the literature in a coherent manner and to outline the goals with which such methods are used. For these purposes, an evaluation method has been developed which makes possible a detailed analysis and comparison of the results of the different algorithms.

Introduction

Natural language processing is concerned to a great extent with the automatic extraction of relations between words by means of statistical methods, usually measures of statistical co-occurrence. This research covers many seemingly different topics, such as lexical acquisition, computing similarity between words, word associations, synonyms, antonyms, idiosyncratic collocations, electronic thesaurus, semantic nets etc. They will be called **extraction of relations between words** throughout this study. All these topics have different applications, for instance Information Retrieval, disambiguation algorithms, speech recognition, or spellcheckers. Though there appears to be a large variety of methods and goals (often resulting in a vague and imprecise terminology), with proper classification of the methods the whole topic presents itself as fairly coherent.

The purpose of this paper is:

- to outline the research development in this area and consider it from a purely statistically based point of view;

* Natural Language Processing Department, Leipzig, PF 920

© 2005 Association for Computational Linguistics

- to review the most commonly employed methodology foundations and to show in which respects they are similar and in which they differ (Sections 1 and 2);
- to propose a suitable method of evaluation and use it to compare the various measures of association that are employed in the literature on this topic, in order to demonstrate the specific effects of these methods (Section 3).

Along with these three aims I will also attempt to compare and organize the terms used by different authors and to specify their meanings.

0.1 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 context. Since the appearance of Firth's paper, the notion of collocation has been 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 them and without the possibility of exchanging any of the elements. For example, it is possible to say 'to beat about the bush', but any other expression consisting of semantically similar words is considered wrong: 'to beat *around 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 Zelig 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 different hierarchical levels and constructed a practical conception on this topic. His own summary could very well have been the motto of this article:

> [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 semantics (focussing on 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), Church et al. (1991) and Smadja (1989). The latter two had already (Church et al., 1989) developed an interpretation of meaning in linguistics from a computational point of view. This new approach was partly derived from psycholinguistic research into 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. They produced simulated knowledge about word associations, which was used to extract lexical and grammatical collocations. He also pointed out other possible applications, especially 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 of how to compute lexical or grammatical collocations semi-automatically. Their usage of the term 'word association' indicates a broader meaning

and indeed, in their examples of automatically computed, strongly associated word pairs which stand in relations such as meronymy, hyperonymy and so forth 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) although they stand in a meronymy relation.

The much cited work of Dunning (1993) was important in two ways. On the one hand, with the log-likelihood measure he introduced an improved mathematical foundation to this field of research. On the other hand and more importantly, he abstracted from the extraction of collocations in particular (only mentioned it) and called the process 'statistical text analysis', which named the topic more precisely though more abstractly.

0.2 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) and 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, in order to avoid typical foreigners' mistakes.

The second line of development can be roughly named **extraction of word associations** and **computation of semantic similarities**. Initially mentioned by Church and also Schvaneveldt (1990), the idea is to (semi-)automatically extract pairs of 'somehow' related or similar words by statistically observing their co-occurrence patterns. The resulting pairs of words of significant co-occurrence are not solely idiosyncratic collocations. Many factors can be responsible for the frequent co-occurrence of two words, all of which could be subsumed as word associations, since this is a rather vague relation,

which 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 have therefore been applied in many different applications, such as word sense disambiguation (Agirre and Rigau, 1995), (Yarowski, 1995), (Karov and Edelman, 1998), (Pantel and Lin, 2000), (Pedersen and Bruce, 1997), word sense discrimination (Schütze, 1998), (Bordag, 2003), (Purandare, 2004) or the computation of thesauri (Grefenstette, 1994) and to a lesser extent in key word extraction (Matsumura, Ohsawa, and Ishizuka, 2003) (PAI) or (Witten et al., 1999) (Kea)), text summarization (Salton et al., 1997), (Mitra, Singhal, and Buckley, 1997) and extraction of terminology (Witschel, 2004). Manually acquired and automatically extended knowledge about word classes (using a trained part-of-speech tagger) has been often used in order to allow only pairs of words of the same word class to be taken into account. This significantly improves the perceived quality of the results, as can be seen in Grefenstette (1996). But manually acquired knowledge and therefore the algorithms based on such knowledge does not fit the otherwise fully unsupervised paradigm.

Along with this research on word associations, there is another, related line of research, which is concerned with the measuring of semantic similarity based on a manually or automatically created thesaurus: cf. Grefenstette (1994) (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 self-evident to see 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) seem to yield better results than those based on manually created thesauri. But due to the lack of comparable data such statements will remain unproven. There have been attempts to create theoretical frameworks for

these 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 termini. 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' (Lin, 1998b), (Rapp, 1996), (Jiang and Conrath, 1997). 'Word association' itself is used either in the psycholinguistic sense of association (Rapp, 2002) or in the statistical meaning of association as a synonym of correlation. 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 to the right of the observed word, to a notion of context which means the whole text in which the observed word occurs. Sometimes the set of significant co-occurrences (computed using contexts) of a given word is called 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, and therefore many different methods of indirect evaluation have been used. Psycholinguistic association experiments (priming experiments) (Burgess and Lund, 1997), to begin with, have been used to generate human-based pairs of words which are 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 most promising and most comparable evaluation is one using large manually crafted knowledge sources such as Roget's Thesaurus (Roget, 1946), WordNet (Miller, 1985), (Fellbaum, 1998) or GermaNet for German (Hamp and Feldweg, 1997) as a gold standard. Unfortunately, again, evaluations using these sources can be done in many different ways, crippling comparability. A standardized tool set or instance is needed.

0.3 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 construction of a thesaurus (Shaikevich, 1985), (Güntzer et al., 1989), has to be distinguished from the other two lines of research, because it embeds them and introduces a different methodology (second order statistics and 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, often synonyms, sometimes plain word association (play, soccer) and sometimes other linguistic relations like derivation, hyperonymy etc. In this third line of research, there is a kind of constructive awareness for the different relations and with it the means and need to differentiate between them algorithmically. Seen in this light, extraction of collocations is one possible task next 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, contrary to word sense disambiguation (Schütze, 1998), (Neill, 2002), (Tamir and Rapp, 2003), (Bordag, 2003), (Purandare, 2004), (Ferret, 2004) belongs to this area as well, since it describes just another kind of specific relations between words.

0.4 Acquisition bottleneck

All previously described algorithms are designed to be as automatic as possible: while involving a minimum manual work (ideally none) to retrieve as much knowledge as possible (recall) and to make as few mistakes as possible (precision). The ultimate goal is to construct language independent, fully unsupervised algorithms with maximized precision and recall at the same time. In reality, the less manual work (in the form of smaller

training sets for taggers or starting words for bootstrapping algorithms) involved, the worse 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 most apparent manual work usually invested is that almost all hitherto mentioned algorithms use a statistical (Brill, 1992), (Brants, 2000) or rule based tagger either to preprocess the corpus they are working on or sometimes even in order to find the units to be analysed, e.g. noun phrases in Hearst (1992) or Berland and Charniak (1999). A statistical tagger always needs a large manually annotated set of sentences for training. There are online sources available such as PennTreeBank, Susanne, Negra, for the major languages (and some minor languages e.g. the Czech National Corpus). But in order to analyze a language with the means of these algorithms, such resources must be available for this language as well. Currently there are no POS taggers which would perform this task in a fully unsupervised manner and without any training sets or a word class annotation, out of a general notion of grammar. Work on this topic by Resnik (1993), Schütze (1995), Schone and Jurafsky (2001) or Freitag (2004) shows that the quality of the results is far from adequate because, as described by Resnik (1993), when using statistic significant co-occurrences as features for clustering, the clusters tend to be both syntactically and semantically motivated. Most of the bootstrapping algorithms that have been mentioned further need a small set of knowledge to begin with, which is given manually. Moreover, they are designed to rely heavily on the quality of this initial set, and errors at this point usually produce further errors in the results.

It is necessary to divide linguistic information extraction algorithms into four classes.

Definition

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

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

Ideally it should be possible to provide an algorithmic description of Type 3 for any kind of language structure to be extracted. The methods to be described further in this paper are of this type, because they compute similarity information about words, using purely statistical means. They do not need any rules or language universals and yet are able to compute synonyms and cohyponyms. Many algorithms are however at best of Type 2, which comprises bootstrapping algorithms or those using language universals. Since using a POS-tagged corpus is the default prerequesite for most algorithms, these algorithms belong to Type 1, because POS-tagging has to be done either manually or by employing a statistical tagger that has been trained on a treebank. Algorithms of Type 0 are not much in use, since language structure comprises such a wide variety of information that encoding it all is considered too costly a solution.

0.5 Discovering structure

There can be no doubt that natural language is structured. The question is rather, which method to employ in order to discover and extract this structure. It is important to note that both syntax and semantics are considered as structured. At the first glance

there are only a few parameters which can be observed. 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 distribution statistics on the frequency of words, paragraphs and texts. It is furthermore possible to observe the significant co-occurrences of word forms within n-grams, sentences, paragraphs, texts or just within a fixed size window and the distribution statistics of these co-occurrences. The point at which the structure of language beyond simple text-, paragraph- or word form boundaries is revealed shows up 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 is 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 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 by themselves are morphologically structured units. Some groups of word forms have a meaning deviating from the combination of the individual word forms. Sentences are not just sets of words: the order of words 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 it just the first step in a long chain of methods to unveil the structure of language automatically?

It is possible to create a meaningful sentence containing any two possible word forms - for any case a context is imaginable in which this sentence would be meaningful. But

language is redundant and thus additional explanations, descriptions or seemingly unnecessary (in the sense of only delivering the intended information) adjectives and other word forms are put into a sentence in order to convey the intended meaning in a highly redundant or modified way (increasing the beauty and readability of a sentence). Note that for these reasons it is not a binary decision whether to put an additional word form into a sentence. It should rather be understood as a *continuum* where some words are more necessary than others in order to convey the information.

It is then possible to try 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 can be 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. But 'sweet' could not possibly be associated with 'terror', 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 means that a method is needed that distinguishes between first, the specific (in the meaning of only relevant for this sentence) co-occurrences of two or more words which appear together only to convey one particular piece of information, and second, co-occurrences of two or more words, which convey general world knowledge or increase readability or beauty of the sentence. There are other related issues - all of them interfering with each other: idiomatic expressions (also called collocations), multi-word lexemes, non-compositional compounds, various semantic relations between words, such as hyperonymy, antonymy, meronymy, synonymy and so on. All of them, parts of the structure

of natural language, result in the co-appearance of word forms in sentences or, on the contrary, inhibit each other's appearance.

Though the co-occurrence measure based methods to date do not attempt to distinguish between these relations (the main goal being somewhat more mathematically and less linguistically motivated), they can be employed in order to describe a more complex algorithm which reveals step by step the structure of natural language.

1 Measuring co-occurrences

As a first step in the automatic discovery of language structure through performing statistical significance tests on co-occurrences, it is important to become aware of all the parameters that can play a role. First of all let us assume 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.), although the formulae 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, which can be obtained by measuring a fixed size part of this infinite corpus.

Secondly, for each element x (usually a word) the number of occurrences n_x within this corpus is known as the frequency $f(x)$. Frequency can be either simply the number of times x occurred: n_x , which is the default, or it can be more like the physical notion of frequency: occurrences of x per text (or some other framing, 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 known as the probability $P(x)$ with which the unit x is the next one to occur if the corpus were to be enlarged by one word or one sentence:

$$P(x) = \frac{f(x)}{n} \tag{1}$$

The third parameter determines whether two specific word forms count as co-occurring or not. They are considered as co-occurring if they are near enough 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 this parameter has not yet found its way into the standard formulae used for co-occurrence measurements (except in the work by Holtsberg and Willners (2001)). All following formulae measure co-occurrence in a window which is not specified. The evaluation of the different measures will be performed on 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. Later, especially after the shift of interest towards extracting particular semantic relations *for each word*, the strongest (again in the meaning of the highest significance value) co-occurrences were 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), context vector (Curran, 2003).

1.1 Basic measures

The most straightforward way to measure the significance of the number of co-occurrences of the word A and the word B is to take the number of co-occurrences n_{AB}, though '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 must perform better:

$$sig_{count} = n_{AB} \tag{2}$$

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, whereas if the frequency of both A and B is equal to 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:

$$sig_{tanimoto} = \frac{n_{AB}}{n_A + n_B - n_{AB}} \tag{3}$$

This measure has been employed by Bensch and Savitch (1992) in order 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 take the Dice coefficient (Dice, 1945):

$$sig_{dice} = \frac{2 \cdot n_{AB}}{n_A + n_B} \tag{4}$$

These two measures have several drawbacks, that they share with the baseline. Although both take into account the possibility of insignificance of co-occurrence of one frequent word with a non-frequent one, it is still highly informative to know that a word occurring only twenty times co-occurs also twenty times with a word occurring 2000 times. In this case both measures yield very small numbers which seem insignificant. Another very important drawback is that they are not normalized against the corpus size. Two words of low frequency accidentally co-occcurring with each other might have a higher significance than two words of high frequency which co-occur quite often. In comparison to the following significance measures they have, however, the advantage of giving normalized numbers in the range [0..1].

1.2 Assumption of independence

There is, however, a statistically founded way to find the proper measurement. The standard procedure in statistics is to formulate a null hypothesis and then determine whether the observed data significantly deviates from it: in this case the null hypothesis is rejected. An optional last step then would be to quantify by how much exactly the observed data deviates from the null hypothesis. The statistical experiments can be modelled accordingly as a number of experiments with positive or negative outcomes: two words either co-occur or do not. The previous parameters have to be taken into account in order to build a statistical model.

Therefore we always have 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 to four different conditions: either both words are present $f(AB)$, only one $f(A\neg B)$ or the other $f(\neg AB)$ is present or none $f(\neg A\neg B)$. This can be put into a contingency table (or expectation table, as it is sometimes misleadingly called); see Tan, Kumar, and Srivastava (2002) for an extended discussion of the properties of measures and the role of the contingency table:

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

Table 1
Contingency table for co-occurrence of items within sentences.

If the occurrence of A is independent of the occurrence of B, then all of the 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 $p(\neg A, \neg B) = p(\neg A) \cdot p(\neg B)$. By constituting the probabilities with the counts n, n_A, n_B and n_{AB} it is possible to arrive at the following equivalence:

$$p(A,B) = p(A) \cdot p(B), \text{ thus } n_{AB} = \frac{n_A \cdot n_B}{n} \quad (5)$$

The other three statements are exactly 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, the test whether the number of co-occurrences of a word pair is significant is usually omitted. In fact, for any word A observed in a corpus there will always be a set of words co-occurring significantly with A. Instead of testing whether the number of observed co-occurrence counts is significant, a quantification is performed which gives a value stating the degree of significance. These values are then used to rank the significant co-occurrences of A according to their significance. There will also be words co-occurring with A insignificantly often, and thus there is the danger of getting insignificant words in this ranking. But since they are insignificant, they will not have a high ranking and will thus be discarded from further processing, because usually the top x words of the ranking are taken. These rankings represent a global (with respect to the corpus) meaning of the word and will be called co-occurrence vectors throughout this paper, as opposed to the similarity vectors, which will be discussed later. It is further possible to compute the similarity of words based on their co-occurrence vectors, but this is not a co-occurrence measure although they are sometimes considered the same: Terra and Clarke (2003).

The following methods perform different measures of how distant observed values are from the expected values under the independence assumption.

1.3 Mutual Information

The most common method of measuring significance of co-occurrence is measuring 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 between two (assumedly) random variables, in this case words, by comparing the probability of observing a and b together (joint probability) with the probabilities of observing a and b independently:

$$sig_{MI}(A,B) \equiv \log_2 \frac{p(A,B)}{p(A) \cdot p(B)} = \log_2 \frac{n \cdot n_{AB}}{n_A \cdot n_B} \tag{6}$$

Another way to arrive at this formula is to look at the four statements which follow from the contingency table (table 1.2) representing 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 can be done by adding a factor x which must equal exactly 1 in the case of independence:

$$n_{AB} = \frac{n_A \cdot n_B}{n} \cdot x \tag{7}$$

which can then be transformed:

$$x = \frac{n \cdot n_{AB}}{n_A \cdot n_B} \tag{8}$$

Taking the logarithm of this would give exactly 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 20 co-occurrences of A with B if A's frequency is 2000 and B's is 20 as well as 'normalizes' against the corpus size. But the normalization against corpus size only results in rewarding of high frequency words with higher significances, which is unnecessary and sometimes (in the case of trying to extract semantic relations) even unwanted, since content words belong to the lower frequency range. 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:

$$sig_{MI}(A,B) = \log_2 \frac{p(A)}{p(A) \cdot p(B)} = \log_2 \frac{1}{p(A)} = \log_2 \frac{n}{n_A} \tag{9}$$

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 sole 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 has already been pointed out by Dunning (1993). Besides, taking the logarithm in this case of application of mutual information is not really necessary, because as opposed to the true significance measure introduced below it only scales down the numbers monotonously, which makes it obvious that this measure is similar to the Dice coefficient except for the multiplication with the corpus size.

1.4 Log likelihood test

As mentioned by Dunning (1993), a proper statistical modelling 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 dependance falls off very fast with distance between the experiments, so that it can be disregarded).
- Both A and B occur in every sentence at most once, which is mostly true even for higher frequency words.

In this case it is possible to describe the distribution of the outcomes using the binomial distribution, which 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:

$$p(k) = p^k(1-p)^{n-k}\binom{n}{k} \tag{10}$$

where the mean is np and the variance is $np(1-p)$. 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:

$$np(1-p) = \frac{n_A \cdot n_B \cdot \left(n^2 - n_A \cdot n_B\right)}{n^3} \tag{11}$$

According to the Zipf distribution of word frequencies (Zipf, 1949), it is safe to say that except for the few most 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 (see also (Wikipedia, 2005)). If the null hypothesis is true, then $-2\log\lambda$ (therefore this test is often called the log-likelihood measure) will be asymptotically χ^2 distributed with degrees of freedom equal to the difference in dimensionality of the hypotheses Θ and Θ_0. Since the likelihood ratio rejects the null hypothesis if the value of this statistic is too small, taking the negative logarithm of λ gives a score for the significance which tells by how much the null hypothesis has been missed.

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:

$$\lambda = \frac{\max_{\theta \in \Theta_0} H(\theta|x)}{\max_{\theta \in \Theta} H(\theta|x)} \tag{12}$$

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 the 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:

$$H\left(p_1, p_2; k_1, n_1, k_2, n_2\right) = p_1^{k_1}\left(1 - p_1\right)^{n_1 - k_1}\binom{n_1}{k_1} p_2^{k_2}\left(1 - p_2\right)^{n_2 - k_2}\binom{n_2}{k_2} \tag{13}$$

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

$$\lambda = \frac{\max_p H\left(p, p; k_1, n_1, k_2, n_2\right)}{\max_{p_1, p_2} H\left(p_1, p_2; k_1, n_1, k_2, n_2\right)} \tag{14}$$

The maxima of the likelihood functions are achieved with $p_1 = \frac{k_1}{n_1}$, $p_2 = \frac{k_2}{n_2}$ and $p = \frac{k_1 + k_2}{n_1 + n_2}$, which reduces the ratio to:

$$\lambda = \frac{max_p L\left(p; k_1, n_1\right) L\left(p; k_2, n_2\right)}{max_{p_1,p_2} L\left(p_1; k_1, n_1\right) L\left(p_2; k_2, n_2\right)} \text{ with } L\left(p; k, n\right) = p^k (1-p)^{n-k} \tag{15}$$

Taking the logarithm of the likelihood ratio gives

$$-2 \log \lambda = 2 \left[\log L\left(p_1; k_1, n_1\right) + \log L\left(p_2; k_2, n_2\right) - \log L\left(p; k_1, n_1\right) - \log L\left(p; k_2, n_2\right)\right] \tag{16}$$

From the contingency table it 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:

$$-2 \log \lambda = 2 \left[\begin{array}{l} n \log n - n_A \log n_A - n_B \log n_B + n_{AB} \log n_{AB} \\ + (n - n_A - n_B + n_{AB}) \log (n - n_A - n_B + n_{AB}) \\ + (n_A - n_{AB}) \log (n_A - n_{AB}) + (n_B - n_{AB}) \log (n_B - n_{AB}) \\ - (n - n_A) \log (n - n_A) - (n - n_B) \log (n - n_B) \end{array} \right] \tag{17}$$

This 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.

1.5 Poisson distribution

Another approach to measure 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. 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. The question then is which these words are. This is solved by taking those with the highest significance value. Though this might be less precise on some occasions than the likelihood method, the differences are assumed to be small enough to be disregarded (this assumption will be tested in the evaluation).

The mean and variance of the Poisson distribution is $\lambda = np$. In this case p is the probability of word A and B to co-occur, 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 results in:

$$sig_{poisson}(A,B) = -\ln\left(\frac{1}{k!}\lambda^k e^{-\lambda}\right) = \ln k! - k\ln\lambda + \lambda \tag{18}$$

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

$$sig_{poisson_1}(A,B) \approx k\left(\ln k - \ln\lambda - 1\right) + \frac{1}{2}\ln 2\pi k + \lambda \text{ with } k > 10 \tag{19}$$

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

$$sig_{poisson_2}(A,B) \approx k\left(\ln k - \ln\lambda - 1\right) \text{ with } k >> 0 \tag{20}$$

There have been three approximations in the process of arriving at this formula as opposed to the 'proper' likelihood measure - using a Poisson distribution instead of a binomial distribution, then taking the negative logarithm of the probability to observe the significance of a given instead of a likelihood measure or some other statistical test and finally using approximation formulae in order to get formulae that are easy to compute.

The last step is especially problematic in this case, because often for small n_A or n_B small k will be of interest. A question arises then, whether using so many approximations could do harm in some way or other. A simple way to check this is to compare the numerical differences between the latter two approximations and show that for not too small k they indeed are asymptotically equal (corpus size 24 million, $n_A = 2000$, $n_B = 4000$) and then to compare one of the two with the likelihood ratio (same setting of corpus size and frequencies).

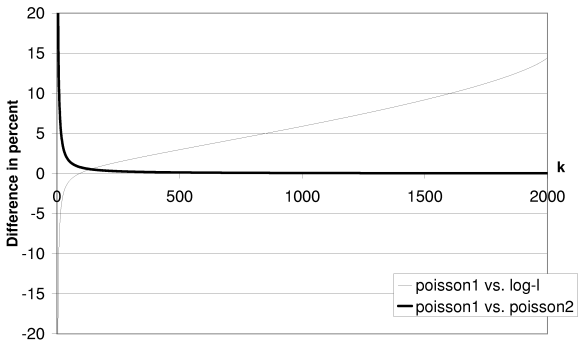

Figure 1
Comparison of a Poisson distribution approximation *poisson1* with the other approximation *poisson2* as opposed to a comparison of a Poisson approximation with the log-likelihood measure *log-l*.

Figure 1 shows that the 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 can be seen in a sample measure in Figure 2.

Therefore it is possible to formulate the hypothesis that the results obtained by the two measures should differ only slightly, especially if it is not the significance values in themselves that are of interest, but instead the ranking of the results based on these values. This hypothesis will be tested by the evaluation to follow in the next chapters.

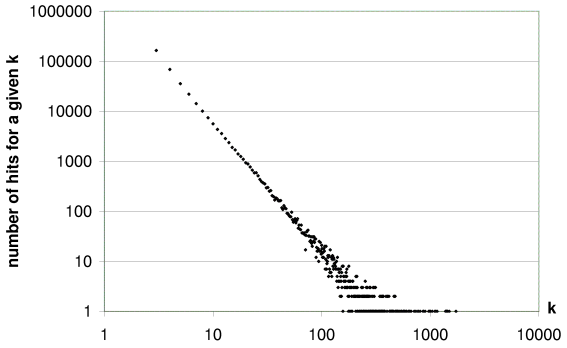

Figure 2
Depicting the distribution of k for word pairs of words where the frequency of the one word $f(A)$ is $3500 \leq f(A) \leq 4500$ and the frequency of the other word $1500 \leq f(B) \leq 2500$.

2 Similarity measures

Having computed a context representation of a word with the means of the methods described above, the next step is to use these contexts to compare the words themselves. The goal is to compute similarity relations between words. Words which are most similar are likely to be synonyms or cohyponyms, since these two relations are relations of semantic similarity. Independently of how the context of a word has been arrived at, 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 most recently Weeds, Weir, and McCarthy (2004)) and to obtain a measure 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) call it, the matching coefficient (if the vectors are represented as sets):

$$sim_{baseline}\left(\overrightarrow{A},\overrightarrow{B}\right)=\sum_{i=0}^{i=n} sgn\left(min\left(\left|a_i\right|,\left|b_i\right|\right)\right) \tag{21}$$

This measure will be taken as a baseline for evaluation, since it is the most simple one. It has the property that it computes a similarity of 0 for word pairs which have no words in common in their context representations and thus are utterly unrelated to each other. It does not take into account either the length of the vectors, nor the total number of non-zero entries in each.

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

2.1 Comparing 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:

$$\overrightarrow{A}^{bin}=\left(a_1^{bin},a_2^{bin},...,a_n^{bin}\right) \text{ with } a_i^{bin}=sgn\left|a_i\right| \text{ and } i=1,...,n \tag{22}$$

Using this representation it is possible to adapt set operations to the 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:

$$\left\|\overrightarrow{A}^{bin}\right\| = \sum_{i=0}^{i=n} {a_i}^{bin} \tag{23}$$

Definition

An intersection $\overrightarrow{A}^{bin} \cap \overrightarrow{B}^{bin}$ is a mapping 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:

$$c_i^{bin} = min\left(a_i^{bin}, b_i^{bin}\right) \text{ and } i = 1, ..., n \tag{24}$$

Definition

A union $\overrightarrow{A}^{bin} \cup \overrightarrow{B}^{bin}$ is a mapping 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:

$$c_i^{bin} = max\left(a_i^{bin}, b_i^{bin}\right) \text{ and } i = 1, ..., n \tag{25}$$

Using these adopted 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 baseline measure previously given can now be represented in the following way:

$$sim_{baseline}\left(\overrightarrow{A},\overrightarrow{B}\right) = \left\|\overrightarrow{A}^{bin} \cap \overrightarrow{B}^{bin}\right\| \tag{26}$$

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

$$sim_{overlap}\left(\overrightarrow{A},\overrightarrow{B}\right) = \frac{\left\|\overrightarrow{A}^{bin} \cap \overrightarrow{B}^{bin}\right\|}{\min\left(\left\|\overrightarrow{A}^{bin}\right\|,\left\|\overrightarrow{B}^{bin}\right\|\right)} \tag{27}$$

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:

$$sim_{dice}\left(\overrightarrow{A},\overrightarrow{B}\right) = \frac{2\left\|\overrightarrow{A}^{bin} \cap \overrightarrow{B}^{bin}\right\|}{\left\|\overrightarrow{A}^{bin}\right\| + \left\|\overrightarrow{B}^{bin}\right\|} \tag{28}$$

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

$$sim_{jaccard}\left(\overrightarrow{A},\overrightarrow{B}\right) = \frac{\left\|\overrightarrow{A}^{bin} \cap \overrightarrow{B}^{bin}\right\|}{\left\|\overrightarrow{A}^{bin} \cup \overrightarrow{B}^{bin}\right\|} \tag{29}$$

When comparing the Dice coefficient with the Jaccard coefficient, it is possible to reduce the comparison to the following inequation:

$$\left\|\overrightarrow{A}^{bin}\right\| + \left\|\overrightarrow{B}^{bin}\right\| - \left\|\overrightarrow{A}^{bin} \cap \overrightarrow{B}^{bin}\right\| \neq \frac{\left\|\overrightarrow{A}^{bin}\right\| + \left\|\overrightarrow{B}^{bin}\right\|}{\left\|\overrightarrow{A}^{bin} \cap \overrightarrow{B}^{bin}\right\|} \tag{30}$$

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 coefficient and the Dice

coefficient will always yield the same rankings of similar words for any word of the vector space. Only the globally most similar word pairs will be different when using these two measures.

Another measure is the cosine measure between binary vectors:

$$sim_{cos\_bin}\left(\overrightarrow{A},\overrightarrow{B}\right) = \frac{\left\|\overrightarrow{A}^{bin} \cap \overrightarrow{B}^{bin}\right\|}{\sqrt{\left\|\overrightarrow{A}^{bin}\right\| \cdot \left\|\overrightarrow{B}^{bin}\right\|}} \tag{31}$$

This measure is a simplification of the cosine mentioned 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.

2.2 Comparing real-valued vectors

Ignoring the real significance values raises two possibly important problems:

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

A combination of all these effects might result in too much loss of information. Real-valued similarity measures therefore have to be compared with those operating on binary vectors, as will be done in Section 3.

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

$$sim_{cos}\left(\overrightarrow{A},\overrightarrow{B}\right) = \frac{\overrightarrow{A}\cdot\overrightarrow{B}}{\left\|\overrightarrow{A}\right\|\cdot\left\|\overrightarrow{B}\right\|} \tag{32}$$

This measure has the property that it computes a value of 0 (completely dissimilar) for any word pair which has no co-occurrences in common. This is the only real-valued measure which shares 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 - the further away two points are, the less similar they are considered. However, there is an almost 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 and 2-norm, because they are the two which are most commonly employed:

$$dist_{1-norm}\left(\overrightarrow{A},\overrightarrow{B}\right) = \sum_{i=0}^{i=n}|a_i - b_i| \tag{33}$$

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 has to 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:

$$dist_{2-norm}\left(\overrightarrow{A},\overrightarrow{B}\right) = \sqrt{\sum_{i=0}^{i=n}(a_i - b_i)^2} \tag{34}$$

To sum up, it is obvious that there are a great variety of possibilities first to compute co-occurrence based context vectors and second to compare such vectors. Surprisingly, the distinction between the first and the second step is rarely drawn and the process of obtaining similar words is seldomly seen as a combination of these steps. In the next section I will therefore provide an evaluation of all the co-occurrence measures combined with all comparison operators.

3 Evaluation

One of the main goals of this study is to ascertain what the various methods described actually do and how their performance can be compared. After considering the various evaluation methods, a conclusion was drawn that for the purposes of this study the gold standard evaluation using a semantic net like WordNet is the most suitable way to evaluate the algorithms. An overview of other possible methods of evaluation with reasons of their rejection (which does not imply that the method concerned is useless) is given below.

3.1 Psycholonguistic association or priming experiments

Association or priming paradigms (Burgess and Lund, 1997) can be used to evaluate the 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 name rapidly some semantically close words after being presented with the stimulus word. The list of most frequently named words can then be compared with the lists obtained automatically. There is a vast array of possibilities to design other kinds of such experiments, but two considerations will always lead to a rejection of such a method in this context:

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 very 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 paper.
2. For the same reason it would not be easily possible for other researchers to reproduce these experiments and validate the results.

3.2 Vocabulary tests

A vocabulary test usually comprises a question and a multiple-choice answer. If both are electronically available, the test can be used quite straightforwardly to evaluate word similarity computation methods. One of the tests, that of English as a foreign language (TOEFL), has been made electronically available, and the part which tests synonyms comprises 80 test items. This kind of evaluation 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 are:

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

2. This test tests only synonymy. Since the intention is to reveal not only how good the measures are, but also what exactly they compute, measuring only one of the plethora of possible linguistic relations will not suffice.

3.3 Application-based evaluation

Application-based evaluation is the indirect method of evaluating results of a knowledge extraction algorithm by putting the extracted knowledge into use and observing how well the application using this knowledge performs. 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. The reasons for not using this kind of evaluation in this study are:

1. It would be relatively easy to use this method for testing synonymy. Any other linguistic relation would, however, be rather hard 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, as well as other sources of error, and also be indirect.

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

3.4 Evaluation by using artificial synonyms

One of the most interesting approaches to evaluating automatic extraction algorithms is that using artificial items. The idea for testing synonymy is to choose randomly one part of occurrences of a word and replace the word by a pseudoword while keeping the other part. These two words then will be perfect artificial synonyms. It is then possible to measure how often the pseudowords are extracted as synonyms of the words that have been retained. Inspired by the artificial pseudowords introduced for word sense disambiguation evaluation as in Gale, Church, and Yarowsky (1992) and Schütze (1992a), this method has been successfully used in various studies, e.g. Grefenstette (1994). Problems like 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.). The sole reason for rejecting of this evaluation method, however, was of another kind:

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

3.5 Gold-standards

The evaluation method which was finally chosen is known as evaluation using 'gold-standards'. The idea is to take a lexical knowledge base, such as a thesaurus or a dictionary, and test the results of the algorithms against this knowledge base. There are, however, many different approaches to this.

3.5.1 The global approach was applied by Grefenstette (1994) in his SEXTANT system. 20 word pairs, which the algorithm computed as most similar, were compared with the equivalent of synonym sets in Roget's Thesaurus (Roget, 1946). He then computes the probability of two randomly selected words to collide in a synonym set in Roget's Thesaurus, which is less than 1:250 attempts. He then comes to the conclusion that this shows the quality of his algorithm, which computed 8 correct pairs. This kind of interpretation of results by taking the globally most similar word pairs and evaluating them is basically the same approach as the one used by Church et al. (1989) for the extraction of collocations.

However, this kind of approach leaves many questions open. First, it does not become clear how well the evaluated algorithm performs for a larger variety of words, posing the problem of the representativity of such a small sample set. Another problem is that the quality of the globally most similar words is highly dependent on problems of the particular measure taken - whether it prefers frequent words or infrequent ones (as is the case with mutual information based algorithms).

3.5.2 The local approach, also called the information retrieval approach by Curran (2003), is to choose a set of words and for each 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 can be correct or wrong. The knowledge whether a returned word is correct can be taken from

electronically available thesarus sources such as Roget's Thesaurus or WordNet. It is therefore possible to measure directly precision (P) and recall (R) values based on how many correct words were among the first 200 similar words and how many of those were synonyms according to the knowledge source.

Since any computed similarity is associated with a similarity value, it is possible to rank the similar words according to these values, similarly to the rankings used in standard information retrieval systems. The further down a correct word is in this ranking, the less valuable this correct hit will be considered. Curran (2003) proposes to use the inverse ranks of matching synonyms (according to WordNet) in order to give a representative score for a computed ranking of similar words for the given input word. These rankings can then be averaged over all input words, finally giving a precise score of the quality of a certain algorithm. Curran selected up to 300 nouns by hand for a detailed analysis of the algorithms described. He used mainly the synonymy relation, since he was interested in finding out how well the algorithms computed similar words.

3.5.3 A modified local approach. Avoiding the preference of one relation (synonymy) over others, I reformulate the evaluation goal: instead of finding out *how well* the algorithms compute word similarity the aim is to find out *which* relations the algorithms compute and only then ask, how well they perform generally. The evaluation steps can then be described as following:

- Compare the various co-occurrence significance measures
- Compare all possible combinations of co-occurrence significance measures with similarity measures
- Give a quantification of the relations computed by the measures where our hypothesis is that similarity relations like synonymy will be dominating.

From this it follows that the results of such an evaluation are best put together into a matrix showing all combinations of co-occurrence measures with similarity measures. This is because after computing statistically significant co-occurrences using any method it is possible to use any context similarity method in order to compare words for the computation of similar words.

3.6 Experimental design

The evaluation comprises the performance of each possible measure combination based on a given corpus against a given knowledge source. First, a precise description is given of how precision and recall will be measured.

Overall precision for one input word is defined as the number of words of the x most similar words according to the algorithm standing in any relation with the input word according to the knowledge base used, divided by x [1].

Precision for a relation and one input word is defined as the number of words standing in the given relation with the input word divided by x.

Overall recall for one word is defined as the number of words found within the x most similar words divided by the number of words standing in any relation, according to the knowledge source. That is, if there are 5 synonyms, 2 antonyms and 1 hyperonym to be found, $x = 5$ and the algorithm finds 2 synonyms, then overall recall is 2/8 for this word.

Recall for a relation and one input word is the same as overall recall except that only the given relation is allowed. That is, if there are 4 synonyms, 2 antonyms and 1 hyperonym to be found, $x = 5$ and the algorithms finds 2 synonyms, then the recall for synonyms is 2/4, for antonyms 0 and for hyperonyms 0 as well.

These precision and recall values can be summed up for all input words and divided by the number of input words in order to obtain overall scores which represent the performance of the algorithm. Note that since for recall finding 2 out of 4 synonyms for one input word is 'worth more' than finding 2 out of 10 possible synonyms, it is possible for one algorithm to have higher recall and lower precision than another.

The evaluation itself was performed on a small part of the 'Projekt Deutscher Wortschatz' (Quasthoff, 1998) corpus that currently contains approximately 35 million

1 x usually has values like $x = 5, 10, 25, 50$ or even 200

German sentences. In order to make the quality of the results roughly comparable to results obtained from the British National Corpus (BNC), which contains 100 mio words, the part of the corpus for the evaluation was chosen to be of the same size. This resulted in randomly selecting about 7.1 million sentences from the main corpus. However, unlike the BNC, the corpus used consists mostly of newspaper texts from the most popular German daily newspaper. Since this evaluation is complete in that it does not try to promote absolute values, but rather relative statements like "Algorithm A' performs significantly better than 'algorithm B", the actual quality of the used corpus is not of decisive significance as long as it is possible to compare the results to the used knowledge source.

Because of the higher flectivity of the German language in comparison to English and the German knowledge source to be used contains only lemmatized words, the selected corpus has been lemmatized using a set of simple rules which introduced additional, but for this task negligible errors. Though not necessary, lemmatization of the corpus provides better comparability with the knowledge source.

Of three available knowledge sources, Dornseiff, 'Annotierter Wortschatz' (a semantic annotation initiative of the project 'Deutscher Wortschatz', not yet made public) and GermaNet, the last was chosen. The reasons for this decision were mainly the size of GermaNet and the fact that GermaNet (and therefore its quality) is wider known and better comparable to WordNet than the other two lexical knowledge sources mentioned.

All the previsously described seven possibilities of computing co-occurrences between words (baseline, mutual information, jaccard and dice coefficient, log-likelihood ratio, first poisson approximation and second poisson approximation) and eight possibilities of comparing the resulting vectors with each other (baseline, overlap factor, dice and jaccard coefficient, binary and real valued cosine and 1-norm and 2-norm distances), all combinations of these possible measures must be compared with each other. Each of the possible measures computes a ranked list of similar words for any encountered word. Since some of the words have a low frequency, it will not be possible to compute reliable information on similarity to other words. Therefore and in order to ensure that each combination of measures is evaluated on the basis of exactly the same words, a set of words has been chosen which fulfills the following conditions:

- The word must be contained in the knowledge source (GermaNet in this case)
- The word must have at least x (in this case $x = 50$) words computed by each of the measures.

As a result only 19069 different words were finally chosen for the evaluation. As there are 52251 unique word phrases in GermaNet, this corresponds to 36,5% of GermaNet. For each of these chosen words, the most similar 5, 10, 25 and 50 words have been evaluated against GermaNet by computing respectively the overall and per relation precision and recall as described above.

3.7 Results

In order to provide an initial overview of the results, the precision of the most similar 5 words ($x = 5$) for each measure combination is given in Table 2, and the recall is given in Table 3. The relations between precision and recall with an increasing x are depicted in Table 4.

	baseline	mutinf	jaccard	dice	log-l	poiss1	poiss2
co-oc. sig.	0.309	0.645	4.246	4.220	4.292	4.342	5.007
baseline	6.537	6.161	10.162^{*+}	9.022^{+}	11.027^{*}	11.280^{*}	12.089^{*}
overlap	3.126	2.937	3.109	2.782	3.211	2.272	3.194
dice	6.654	6.071	10.105^{*+}	8.979^{+}	11.392^{*}	11.516^{*}	11.806^{*}
jaccard	6.654	6.071	10.105^{*+}	8.979^{+}	11.392^{*}	11.516^{*}	11.806^{*}
cos_bin	6.705	5.675	9.411^{+}	8.546^{+}	11.146^{*}	11.164^{*}	11.106^{*}
cos	7.096	5.146	8.630^{+}	8.206^{+}	9.048^{+}	8.698^{+}	8.668^{+}
1-norm	4.432	0.921	1.357	1.445	2.465	2.064	1.531
2-norm	4.142	1.095	2.683	2.815	5.664	5.515	4.829

Table 2
Overall precision in percent for each measure combination for the 5 most similar words ($x = 5$) of each of the input words. The co-occurrence measures are placed horizontally (their evaluation tagged with co-oc. sig.), whereas the comparison operators are placed vertically. The tags * and $^{+}$ mark two of the equivalence classes, according to the Sheffé test.

In this scenario (computing similar words) achieving high precision ratios is easier than achieving high recall values. As can be seen in Table 3, recall values are generally very low. There are some factors, however, which further inhibit near-100% values.

- In GermaNet (as well as to some extent in WordNet) lexical items often consist of two or more words, for example: 'Amerikanischer Dollar'. Since the evaluated algorithms do not include a word group detection algorithm, such items cannot be found.

- Specifically in GermaNet a number of artificial words were included which do not exist but which fill 'empty places' in the hierarchy. This was done in an attempt to 'correct' the otherwise skewed hierarchy, because of missing lexicalizations which might be present in other languages. Obviously, these artificial words cannot be found using corpora approaches.

- Ideally, the measures presented should compute word similarity (as intended). Therefore the overall recall (as opposed to precision) should never reach 100%, because other relations, such as consists-of or part-of, are present in GermaNet.

	baseline	mutinf	jaccard	dice	log-l	poiss1	poiss2
co-oc. sig.	0.052	0.148	0.781	0.779	0.854	0.855	0.975
baseline	1.058	1.307	2.142	1.822	2.227	2.279	2.519
overlap	0.708	0.787	0.842	0.746	0.909	0.679	0.962
dice	1.096	1.311	2.163	1.820	2.349	2.361	2.510
jaccard	1.096	1.311	2.163	1.820	2.349	2.361	2.510
cos_bin	1.106	1.254	2.068	1.762	2.333	2.335	2.420
cos	1.095	1.287	1.895	1.782	1.841	1.861	1.926
1-norm	0.773	0.224	0.345	0.358	0.546	0.470	0.387
2-norm	0.751	0.273	0.618	0.631	1.073	1.064	0.923

Table 3
Overall recall in percent for each measure combination for the 5 most similar words of each of the input words.

Table 2 and 3 show that the baseline (counting co-occurrences and then comparing these vectors by counting mutually contained words) performs very well compared to other possible measure combinations. In fact, half of the combinations have less precision and only a roughly similar recall. The best precision is only about 2 times higher than

that of the baseline, whereas the best recall is is 2,5 times higher than that of the baseline. It is interesting that whereas for pure co-occurrence significance computation (*co-oc. sig.* in the tables) the dice and jaccard measures are obviously better than any other measure, the combination of measuring co-occurrence and then comparing the resulting vectors is better if instead of dice or jaccard the 'true' mathematically more motivated measures of log-likelihood or one of the two poisson approximations are taken. This indicates that the vectors of co-occurrence measures of dice and jaccard contain different rankings than log-likelihood or the poisson approximations.

If the ranking of most similar words is successful, then rising x should cause the precision ratio to drop and at the same time the recall value to rise, because more and more 'attempts' are made to find a correct word according to the knowledge source. But if the ranking were good, then each following word is less probable to actually be a correct one. As can be seen in Table 4 this is the case for all measure combinations.

As has been predicted in Section 2.1, dice and jaccard (as comparison measures) produce exactly the same local rankings which results in exactly the same numbers in both precision and recall. Another prediction was only partly fulfilled. In Section 1.5 the prediction was made that the two poisson approximations would not differ significantly from each other, but they would differ from log-likelihood. The poiss2 approximation generally performs better as a direct co-occurrence measure and when using the binary valued comparison operators. It performs worse than log-likelihood or poiss1 when comparing with real valued measures.

However, according to the Sheffé test the results marked with * are all in one equivalence class of means which do not differ significantly from each other as opposed to all other values. The second equivalence class, marked with $^+$, is worse and overlaps only slightly with the first. All other values differ statistically significantly from the first two classes and have much lower values. These findings show, that although the actual numbers of the best cominations differ slightly, the differences might not be significant in the long run.

	precision				recall			
$x =$	5	10	25	50	5	10	25	50
poiss2 baseline	12.089	9.567	6.433	4.586	2.519	3.716	5.682	7.562
poiss2 dice	11.806	9.357	6.385	4.578	2.510	3.741	5.805	7.777
poiss2 cos_bin	11.106	8.833	6.072	4.359	2.420	3.644	5.704	7.635
poiss2 cos	8.668	7.202	5.249	3.874	1.926	3.051	5.131	7.132
poiss2 2-norm	4.829	3.857	2.743	2.147	0.923	1.434	2.505	3.906
log-l dice	11.392	8.988	6.171	4.417	2.349	3.492	5.407	7.170
dice dice	8.979	7.144	4.897	3.571	1.820	2.716	4.273	5.888
mutinf dice	6.071	4.835	3.333	2.430	1.311	1.973	3.144	4.360
baseline dice	6.654	5.308	3.558	2.462	1.096	1.625	2.403	3.196
baseline baseline	6.537	5.151	3.412	2.375	1.058	1.534	2.268	3.052

Table 4
Precision drops with increasing x whereas recall rises.

		anton.	cohyp.	cons.-of	hyperon.	hypon.	part-of	synon.
poiss2-	P	0.247	7.019	0.324	1.751	2.528	0.393	1.350
baseline	R	0.656	1.986	0.555	1.079	3.154	0.844	2.265
poiss2-	P	0.258	6.964	0.306	1.861	2.233	0.352	1.457
dice	R	0.682	2.025	0.548	1.261	2.799	0.767	2.449
poiss2-	P	0.136	5.520	0.180	1.368	1.453	0.147	1.255
cos	R	0.465	1.676	0.289	1.040	1.863	0.322	2.175
poiss2-	P	0.063	3.205	0.090	1.122	0.404	0.072	0.672
2-norm	R	0.241	0.952	0.168	0.779	0.426	0.180	1.158
log-l-	P	0.188	6.776	0.315	1.432	2.420	0.389	1.197
dice	R	0.507	1.892	0.544	0.980	2.962	0.836	2.026
dice-	P	0.179	5.368	0.248	1.223	1.754	0.290	1.001
dice	R	0.460	1.527	0.441	0.859	2.118	0.591	1.741
mutinf-	P	0.125	4.144	0.187	0.850	0.618	0.151	0.816
dice	R	0.337	1.271	0.437	0.798	0.771	0.319	1.430
baseline-	P	0.168	4.329	0.154	0.444	1.246	0.283	0.318
baseline	R	0.316	0.882	0.209	0.182	1.454	0.515	0.522

Table 5
Absolute precision and recall for each relation for a selected set of measure combinations $x = 5$.

Surprisingly, simple counting of mutually contained words in order to compare vectors of words performs best in combination with any measure of co-occurrence. Comparisons using the binary compare operators like dice, jaccard or binary cosine fare equally well or are only slightly worse. It is not surprising, however, that the overlap factor performs so badly. This is because it prefers statistically non-representative words over represen-

tantive ones. If a word B has only two significant (no matter which significance measure is used) co-occurrences and another word A has 100, then the chances are good that the two of B will be contained in the 100 of A. This vector will get a maximum overlap factor of 1.0. On the other hand, another word C with 100 co-occurrences and 80 of them mutually contained in the vectors of A and C will receive the similarity of only 0.8, according to the overlap factor.

Another surprising aspect is that the result of taking the significance values into account by taking real valued comparison operators like 1-norm, 2-norm or even cosine is mediocre, sometimes even far worse than the simpler binary approaches. This indicates that the co-occurrence significance values must possibly be disregarded in favor of some kind of ranking based similarity measure.

Since one of the intentions of these evaluations was to show which kinds of relations are computed, the $x = 5$ precision and recall are shown on a per-relation basis for a selected set of measure combinations in Table 5. However, most relations which are sparsely represented in GermaNet, get a count of zero or near-zero and are therefore not shown in the table.

The result of comparing the precision values on a per-relation basis for each measure combination can be seen in Table 6. First, cohyponymy is always the relation with the highest precision. Second, different measure combinations clearly 'prefer' different relations. The highest relative precision for the cohyponymy relation are achieved by the direct co-occurrence measures such as *dice sig.* or *poiss2 sig.* or with the second order baseline. The highest relative precision for synonymy, on the other hand, is achieved by using a combination of *poiss2 cos*, that is, computing the co-occurrence significance using the second poisson approximation and then comparing the resulting vectors using the cosine measure. The best rating for hyperonymy is achieved by the *poiss2 2-norm* combination.

	anton.	cohyp.	con.of	hyper.	hypon.	part-of	synon.	total
dice sig.	5,285	66,710	3,617	8,664	6,779	3,379	5,566	100,0
poiss2 sig.	4,494	61,006	2,929	6,815	14,846	4,697	5,213	100,0
basel. basel.	2,420	62,360	2,218	6,396	17,949	4,077	4,581	100,0
poiss2 basel.	1,815	51,565	2,380	12,864	18,572	2,887	9,918	100,0
poiss2 dice	1,921	51,850	2,278	13,856	16,626	2,621	10,848	100,0
poiss2 cos	1,352	54,876	1,789	13,600	14,445	1,461	12,476	100,0
poiss2 2-norm	1,119	56,947	1,599	19,936	7,178	1,279	11,940	100,0
possible	0,672	67,965	0,673	25,487	3,410	0,375	1,418	100,0

Table 6
Different measure combinations vary in the preference of relations. A comparison of the distributions of relative precision values with $x = 5$.

Since usually these algorithms have been used to compute similar words with the implicit hope of obtaining synonyms, it is surprising to find the synonymy relation to be one of the weaker represented ones. As can be seen especially from Table 6, cohyponyms make up the largest part of correctly retrieved words. All the other relations depicted are to a lesser extent also similarity relations: antonymy, for example, relates two words which are very similar to each other except in one or more opposing features.

Generally the precision and recall obtained values are very low, which indicates that these basic measure combinations can only be taken as a first step in a processing chain for the extraction of lexical information. On the other hand, the world knowledge represented in GermaNet differs from the world knowledge represented by the corpus from which these algorithms 'learned'. In fact, the precision and recall values can be modified to look e.g. twice as good by evaluating not all words, but only those which are very frequent in the corpus or by evaluating a hand-picked set of 200 words which are well-represented in GermaNet, as done by Curran (2003). Other ways of artificially improving the numbers include the removal of all names or restricting similar words to those of the same word class by the use of a POS tagger. For this reason and for those given in Section 3.7 these numbers should not be taken as absolute values - they can only be used to gain relative information about the quality of the algorithms.

4 Conclusions

This study reviews the research of the unsupervised extraction of linguistic relations. First, a review of the theoretical foundations is given, by describing the historical development of structuralism. This line of research has recently been reintroduced and broadened by Rieger (1991) in a computational form. It has been continued by studies such as Mehler (2001) and most recently Bordag and Heyer (in preparation). The structuralism offers a linguistically adequate framework which fuses mathematical descriptions with linguistic hypotheses and is therefore quite straightforward to implement.

Since there are strongly varying notions of what 'unsupervised' particularly means, this study discusses this term in a manner which demonstrates the existence of different kinds of 'unsupervisedness' by introducing four classifications of algorithms that differ in which kind of knowledge they presuppose. Apparently, true unsupervisedness strongly correlates which universality and language independence. Such true unsupervised Type 3 algorithms must be based on linguistic hypotheses and, if they can be found, offer a good empirical 'proof' of the correctness of the initial hypothesis. Additionally, it seems that such algorithms extracting different kinds of relations such as semantic relations (anotnymy, synonymy, etc.), syntactic relations (word class, grammatical congruency) can boost each others performance. For example, as soon as there is information available about word classes, the precision and recall values of the algorithms presented in this study would rise significantly, because most semantic relations hold only between words of the same word class. Ideally, linguistic knowledge should be treated as a set of hypotheses which ought to be tested by providing purely statisical and language independent extraction algorithms for the phenomens described by these hypotheses.

This study demonstrates that the construction of an algorithm for extraction of collocations, synonyms, hyperonyms or other linguistic relations should be guided by an awareness of the effects of the methods employed and provides means to measure

some of these effects. It shows which relations are favoured by the most widely used methods, the co-occurrence statistics and vector comparisons, and how well they perform. Moreover, this study provides evidence that (and by how much) the present algorithms can outperform the frequency baseline. In this respect, the results differ from those of Krenn and Evert (2001) due to the use of a different evaluation method and knowledge base (GermaNet instead of collocations) and also due to a different approach towards the interpretation of the results of a co-occurrence measure (locally per word vs. globally best pairs).

The evaluation method developed enables easy comparison of new measures and algorithms. It can, for example, be used to compare algorithms using different corpora, thus alleviating the problem that different authors use different corpora to compare new algorithms with known ones. It suffices to evaluate the new algorithm alongside one or more of the known algorithms. The absolute numbers will vary, but not the relations between the performances of the algorithms.

There is a host of smaller but nonetheless important questions which have not been answered in this study and which are subject of further research. One is the question of 'How large must a corpus be?'. Using the presented evaluation method it should be possible to find an indication of how precision and recall vary with a growing corpus size. Another question is how exactly co-occurrence has to be measured - whether sentence-wide or using a window of a fixed size. Finally, it would be important to find out, whether using another language, for example English (and evaluating against WordNet), produces the same or similar results.

References

Agirre, Eneko and German Rigau. 1995. A proposal for word sense disambiguation using conceptual distance. In Tzigov Chark, editor, *Proceedings of the First International Conference on Recent Advances in NLP*, Bulgaria.

Baroni, Marco and Sabrina Bisi. 2004. Using cooccurrence statistics and the web to discover synonyms in a technical language. In *Proceedings of the ELRA 04*.

Bensch, P. A. and Walter J. Savitch. 1992. An occurrence-based model of word categorization. In *Presented at 3rd Meeting on Mathematics of Language (MOL3)*.

Benson, Morton, Evelyn Benson, and Robert Ilson. 1986. *The BBI Combinatory Dictionary of English: A Guide to Word Combinations*. John Benjamins, Amsterdam, Netherlands.

Berland, M. and Eugene Charniak. 1999. Finding parts in very large corpora. In *Proceedings of ACL 1999, College Park.*, pages 57–64.

Biemann, Christian, Stefan Bordag, Gerhard Heyer, Uwe Quasthoff, and Christian Wolff. 2004. Language-independent methods for compiling monolingual lexical data. In *Proceedings of CICLing 2004*, pages 215–228. Springer Verlag.

Bordag, Stefan. 2003. Sentence co-occurrences as small-world-graphs: A solution to automatic lexical disambiguation. In *Proceedings of CICling-03, LNCS 2588*, pages 329–333. Springer.

Bordag, Stefan and Gerhard Heyer. in preparation. *A Structuralist Framework for Quantitative Linguistics*. Studies in Fuzziness and Soft Computing. Springer, Berlin/Heidelberg/New York.

Brants, Thorsten. 2000. TnT — a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000*.

Brill, Eric. 1992. A simple rule-based part-of-speech tagger. In *Proceedings of ANLP-92*, pages 152–155.

Burgess, Curt and Kevin Lund. 1997. Modelling parsing constraints with high-dimensional context space. *Language and Cognitive Processes*, 12:177–210.

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 *nternational 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*, pages 115–164, Hillsdale, NJ. Lawrence Erlbaum.

Church, Kenneth Ward and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16:22–29.

Curran, James Richard. 2003. *From Distributional to Semantic Similarity.* Ph.D. thesis, Institute for Communicating and Collaborative Systems, School of Informatics. University of Edinburgh.

Dagan, Ido and Kenneth Ward Church. 1994. Termight: Identifying and translation technical terminology. In *Proc. of the 4th Conference on Applied Natural Language Processing (ANLP)*, pages 34–40, Stuttgart, Germany.

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 and the 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 56–63, Madrid, Spain, July.

Dagan, Ido, Shaul Marcus, and Shaul Markovitch. 1995. Contextual word similarity and estimation from sparse data. *Computer Speech and Language*, 9:123–152.

Dice, Lee R. 1945. Measures of the amount of ecologic association between species. *Journal of Ecology*, 26(3):297–302.

Doyle, Arthur Conan. 1902. The hound of the baskervilles.

Dunning, T. 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.

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*, pages 188–195, Toulouse, France.

Fellbaum, Christiane. 1998. A semantic network of english: The mother of all wordnets. *Computers and the Humanities*, 32:209–220.

Ferret, Olivier. 2004. Discovering word senses from a network of lexical cooccurrences. In *Proceedings of the ELRA 04*.

Firth, J.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. and Ricardo Baeza-Yates. 1992. *Information Retrieval*. Englewood Cliffs, NJ: Prentice Hall.

Freitag, Dayne. 2004. Toward unsupervised whole-corpus tagging. In *Proceedings of Coling 2004*, Geneva, Switzerland.

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.

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.

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. AAAI Press.

Grefenstette, Gregory. 1994. *Explorations in Automatic Thesaurus Discovery*. Boston: Kluwer Academic Press.

Grefenstette, Gregory. 1996. Evaluation techniques for automatic semantic extraction: Comparing syntactic and window-based approaches. *Corpus Processing for Lexical Acquisition*, pages 205–216.

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.

Hamp, Birgit and Helmut Feldweg. 1997. Germanet - a lexical-semantic net for german. In *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, Madrid.

Harris, Zeelig S. 1968. *Mathematical Structures of Language*. Wiley, New York.

Hatzivassiloglou, V. and K. R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of ACL/EACL-97*, pages 174–181.

Hearst, Marti A. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING 92*, pages 539–545.

Holtsberg, A. and C. Willners. 2001. Statistics for sentential co-occurrence. In *Working Papers 48*, pages 135–148.

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, Taiwan*.

Karov, Yael and Shimon Edelman. 1998. Similarity-based word sense disambiguation. *Computational Linguistics*, 24:41–59.

Krenn, Brigitte. 2000. Collocation mining: Exploiting corpora for collocation identification and representation. In *Proceedings of KONVENS 2000*, Ilmenau, Germany.

Krenn, Brigitte and Stefan Evert. 2001. Can we do better than frequency? a case study on extracting pp-verb collocations. In *Proceedings of the ACL Workshop on Collocations*, pages 39–46, Toulouse, France.

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.

Lehr, Andrea. 1993. Kollokationsanalysen. von der kollokationstheorie des kontextualismus zu einem computergestützen verfahren. *ZGL*, 21:2–19.

Lehr, Andrea. 1996. *Kollokationen und maschinenlesbare Korpora. Ein operatives Analysemodell zum Aufbau lexikalischer Netze*. Germanistische Linguistik 168. Niemeyer, Tübingen.

Lin, Dekang. 1998a. Automatic retrieval and clustering of similar words. In *Proceedings of COLING/ACL-98*, pages 768–774.

Lin, Dekang. 1998b. Extracting collocations from text corpora. In *First Workshop on Computational Terminology*.

Manning, Christopher D. and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

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.

Mehler, Alexander. 2001. *Textbedeutung. Zur prozeduralen Analyse und Repräsentation struktureller Ähnlichkeiten von Texten*, volume 5. of *Sprache, Sprechen und Computer - Computer Studies in Language and Speech*. Peter Lang, Frankfurt am Main.

Miller, George A. 1985. Wordnet: a dictionary browser. In *Proceedings of the First International Conference on Information in Data*, University of Waterloo, Waterloo.

Mitra, Mandar, Amit Singhal, and Chris Buckley. 1997. Automatic text summarization by paragraph extraction. In *Proceedings of the Workshop on Intelligent Scalable Text Summarization*, pages 39–46. Association for Computational Linguistics, July.

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, 27th May - 1st June.

Neill, Daniel B. 2002. Fully automatic word sense induction by semantic clustering. *Computer Speech*.

Pantel, P. and Dekang Lin. 2000. Word-for-word glossing with contextually similar words. In *Proc. of the 1st Annual Meeting of the North American Chapter of Association for Computational Linguistics (NAACL2000)*, pages 78–85, Seattle, USA.

Pedersen, Ted and Rebecca Bruce. 1997. Distinguishing word senses in untagged text. In *EMNLP 2*, pages 197–207.

Purandare, Amruta. 2004. *Word Sense Discrimination by Clustering Similar Contexts*. Ph.D. thesis, Department of Computer Science, University of Minnesota, August.

Quasthoff, Uwe. 1998. Projekt: Der deutsche wortschatz. In Gerhard Heyer and Christian Wolff, editors, *Tagungsband zur GLDV-Tagung*, pages 93–99, Leipzig, March. Deutscher Universitätsverlag.

Quasthoff, Uwe and Christian Wolff. 2002. The poisson collocation measure and its applications. In *Second International Workshop on Computational Approaches to Collocations*.

Rapp, Reinhard. 1996. *Die Berechnung von Assoziationen*. Hildesheim: Olms.

Rapp, Reinhard. 2002. The computation of word associations. In *Proceedings of COLING-02, Taipei, Taiwan*.

Resnik, Philip Stuart. 1993. *Selection and Information: A Class-Based Approach to Lexical Relationships*. Ph.D. thesis, University of Pennsylvania.

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.

Rieger, Burghard. 1991. Distributed semantic representations of word meanings. In *Parallelism, Learning, Evolution. (Proceedings Workshop on Evolutionary Models and Strategies / Workshop on Parallel Processing: WOPPLOT 89) Becker, J.D./ Eisele, I./ Mündemann, F.W. (eds.)*, pages 243–273. Springer.

Roget, P. M. 1946. *Roget's International Thesaurus*. Thomas Y. Crowell, New York.

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. Springer-Verlag.

Salton, Gerard, Amit Singhal, Mandar Mitra, , and Chris Buckley. 1997. Automatic text structuring and summarization. *Information Processing and Management*, 33(2):193–207.

Schone, Patrick and Daniel Jurafsky. 2001. Language-independent induction of part of speech class labels using only language universals. In *Workshop at IJCAI-2001*, Seattle, WA., August. Machine Learning: Beyond Supervision.

Schütze, Hinrich. 1992a. Context space. In *Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, Menlo Park, CA. 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 the EACL 7*, pages 141–148.

Schütze, Hinrich. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24:97–124.

Schvaneveldt, Roger. 1990. *Pathfinder Associative Networks: Studies in Knowledge Organization*. Ablex.

Seretan, Maria-Violeta. 2003. *Syntactic and Semantic Oriented Corpus Investigation for Collocation Extraction, Translation and Generation*. Ph.D. thesis, Language Technology Laboratory, Department of Linguistics, Faculty of Arts, University of Geneva.

Shaikevich, Anatole Y. 1985. Automatic construction of a thesaurus from explanatory dictionaries. *Automatic Documentation and Mathematical Linguistics*, 19:76–89.

Smadja, Frank. 1989. Macrocoding the lexicon with co-occurrence knowledge. In U. Zernik, editor, *Proceedings of the First International Lexical Acquisition Workshop*.

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.

Tamir, Raz and Reinhard Rapp. 2003. Mining the web to discover the meanings of an ambiguous word. In *Proceedings of ICDM 03*, 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.

Tanimoto, T.T. 1958. An element mathematical theory of classification. Technical report, I.B.M. Research, New York, NY USA, November.

Terra, Egidio and C. L. A. Clarke. 2003. Frequency estimates for statistical word similarity measures. In *HLT-NAACL 2003*, pages 165–172.

Turney, Peter D. 2001. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proceedings of ECML*, pages 491–502.

Turney, Peter D. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL-02*, pages 417–424.

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-2004*, Geneva, Switzerland, August.

Wikipedia. 2005. Wikipedia, the free encyclopedia. Web site: http://wikipedia.org.

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.

Witten, Ian H., Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Kea: Practical automatic keyphrase extraction. In *Proceedings of DL 99*, pages 254–256.

Yarowski, David. 1995. Unsupervised word sense disambiguation rivaling supervised methods. *ACL*, 33:189–196.

Zipf, George Kingsley. 1949. *Human Behaviour and the Principle of Least-Effort*. Addison-Wesley, cambridge ma edition.