Unsupervised Learning of Second Order Dependencies from Corpora

Reinhard Rapp
Universität Mainz, FASK
D-76711 Germersheim
rapp@mail.fask.uni-mainz.de

Abstract

It is shown that basic language processes such as the orthographic comparison of strings, the formation of word classes, and the acquisition of the lexicon can be simulated using statistical models that analyze the distribution of words in large text corpora. Whereas – as described in the literature – for the extraction of collocations and word associations first order statistics are sufficient, the successful simulation of the above tasks requires second order statistics. It turns out that the same core algorithm based on the computation of vector similarities can be successfully used for all three tasks, but that different kinds of preprocessing put the emphasis on either the level of characters, function words, or content words. The success of the simulations suggests that in human language processing an analogous kind of algorithm is used at different levels of processing.

1 Introduction

More than 2300 years ago, in the heat of ancient Greece, Aristotle assumed that the human brain serves as a cooling device for the body and that mental processes should be attributed to our heart (Brockhaus, 2002). Science has advanced since then and we know that the genius erred in this case. The fact that two millennia later it is still unclear what fundamental mechanisms underlie human information processing, however, proves the difficulties in understanding the brain.

In this paper we address questions of language acquisition and processing. We try to show that at different levels of language processing the computation of similarities between feature vectors seems to be an appropriate mechanism to simulate human behavior.[1] The specific language processing tasks we consider are human intuitions concerning
1) orthographic similarities between strings;
2) the assignment of words to parts of speech;
3) similarities in word meaning.

All these tasks have been extensively dealt with in the psycholinguistic and computational linguistic literature (Manning & Schütze, 1999; Rapp, 1996). However, since they are usually assigned to different sub-fields of linguistics (e.g. orthography, grammar, and semantics), in most cases the tasks have been considered individually and few attempts have been made to demonstrate relationships in their underlying mechanisms.

The paper is organized as follows: First, for each of the three tasks mentioned above, we describe a simulation algorithm and present some results. We then discuss what the algorithms have in common and what accounts for the different outcomes. Finally, we address shortcomings and unresolved problems.

2 Orthographic Similarity

String comparison based on orthographic similarities is used in many industrial software applications: for example, in full text information retrieval systems to deal with misspellings and inflected forms, and in translation memory systems to help the translator finding previously translated similar text patterns. The success of these systems shows that the retrieval of similar strings that match our intuitions can be considered as a more or less re-

[1] We adapted this approach from information retrieval (Salton & McGill, 1983). In other contexts, this kind of learning and generalization is also known as memory-based learning, similarity-based learning, *k* nearest neighbor classifier etc., as pointed out by Daelemans et al. (2002).

solved problem, although from an engineering point of view there is always a requirement for some fine-tuning to adapt for the peculiarities of a specific environment.

In the literature, several successful algorithms have been described (Peterson, 1980, gives an overview). A straightforward yet effective method for the computation of string similarities considers the relative number of substrings (e.g. character bi- or trigrams) two strings have in common (Angell et al., 1983). More precisely, in order to obtain the orthographic similarity *S* (expressed in percent) between the two strings *A* and *B*, the following formula is applied:

$$S = \frac{200 \cdot f(A \& B)}{f(A) \cdot f(B)}$$

Herein, *f(A&B)* is the number of *n*-grams the two strings have in common, and *f(A)* is the total number of *n*-grams in string *A*. In order to give all characters the same weight, $n - 1$ blanks should be added to both ends of the strings beforehand. Table 1 illustrates the method using an example based on bigrams. (In the table, the symbol '×' denotes a blank.)

If the words most similar to a given word are to be determined, the word must be compared to all words in a vocabulary. Table 2 shows some results based on a list of German full forms comprising about 65000 entries (see Rapp, 1996, for details). Comparisons with answers given by human subjects indicate that this method leads to results that closely agree with human judgment, but has better recall and consistency.

For the purpose of this paper it is interesting to know that an algorithm similar to the above can be easily formulated using vector similarities. To do so, we need a matrix of all strings we are interested in versus all bigrams they contain, with the entries indicating how often a certain bigram is found in the respective string (see table 3 for a small sample matrix).

Whereas in table 1 the number of connections gives a quick impression of how similar the two strings are, the matrix in table 3 does not allow estimation of the similarities at first glance. However, in this matrix representation we can more easily see that the vector similarity measures well known from information retrieval are also applicable to the problem of finding similar strings (for an overview on similarity metrics see Salton & McGill, 1983).

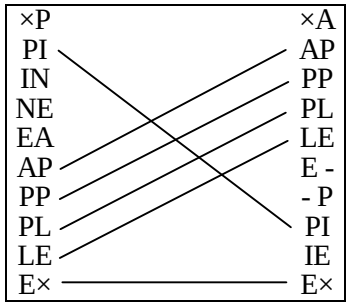

Table 1: Bigrams common to the two strings *pineapple* and *apple-pie*. Since six out of ten bigrams correspond, the similarity is 60%.

Stimulus	*Most Similar Words*
Einzelheiten	Eigenheiten, Einheiten, Einzelheit, einzuleiten, Eitelkeiten
Gericht	Gerichts, Gesicht, Gesichts, Gewicht, Gewichts, Gedicht
elektrisch	elektrische, elektrischem, elektrischen, elektronisch
einschlafen	eingeschlafen, einschlagen, geschlafen, schlafenden

Table 2: Orthographic similarities as computed from a list of 65000 German words.

	×A	×P	×N	-P	AP	AR	E×	E-	EA	IE	IN	LE	NE	PE	PI	PL	PP	R×
×APPEAR×	1	0	0	0	1	1	0	0	1	0	0	0	0	1	0	0	1	1
×APPLE×	1	0	0	0	1	0	1	0	0	0	0	1	0	0	0	1	1	0
×APPLE-PIE×	1	0	0	1	1	0	1	1	0	1	0	1	0	0	1	1	1	0
×NEAR×	0	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0	1
×PINE×	0	1	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0
×PINEAPPLE×	0	1	0	0	1	0	1	0	1	0	1	1	1	0	1	1	1	0

Table 3: Matrix of words and their bigrams.

As justified in a previous paper (Rapp, 2002) we tend to prefer the city block metric for vector comparisons, although – as experience shows – other metrics such as the cosine measure (which computes the cosine of the angle between two vectors) usually lead to similarly good results.[2]

The city-block metric computes the similarity between two vectors X and Y as the sum of the absolute differences of corresponding vector positions:

$$s = \sum_{i=1}^{n} |X_i - Y_i|$$

To determine the words most similar to a given word,[3] the vector of this word is compared to all other vectors in the matrix and the words are ranked according to the similarity values obtained. It is expected that the most similar words are ranked first in the sorted list.

If we compute the similarities of all possible pairs of the words listed in table 3, we obtain a resulting similarity matrix as shown in table 4. By inspection, the computed similarity values seem to make sense; for example, the pair *APPLE* / *APPLE-PIE* gets the lowest score (highest similarity), and the pair *APPLE-PIE* / *NEAR* gets the highest score (lowest similarity). With two scores of 11, *APPEAR* seems to be equally similar to *PINEAPPLE* and *APPLE-PIE*.

Although a more detailed evaluation of the results may be desirable, we nevertheless take our results as evidence that the vector similarity approach is appropriate for computing orthographic similarities and move on to applying the same method to word classification.

[2] As described in Rapp (2002), we usually apply some kind of normalization or significance testing to the matrix entries (e.g. the log likelihood ratio as proposed by Dunning, 1993) before computing vector similarities. However, due to the almost binary character of the values in the string/bigram matrix (mostly zeros, some ones, higher numbers only when a bigram occurs several times in a string), this is probably not worthwhile in this case.

[3] Due to the inverse characteristic of the city block metric, when we speak of a high similarity we mean a low value for the sum of the absolute differences of the matrix entries.

	APPEAR	APPLE	APPLE-PIE	NEAR	PINE	PINEAPPLE
APPEAR	0	7	11	6	12	11
APPLE	7	0	4	9	9	6
APPLE-PIE	11	4	0	15	10	8
NEAR	6	9	15	0	8	11
PINE	12	9	10	8	0	5
PINEAPPLE	11	6	8	11	5	0

Table 4: Orthographic similarities as computed.

3 Word Classification

Word classes as described in grammar books are usually defined using a mix of morphologic, syntactic, and semantic criteria. However, among others, Bergenholtz & Schaeder (1977) argue that mixing criteria leads to unsystematic results and that only one kind of criteria should be applied. They point this out by comparing the classification of some German closed class words as provided by four renowned monolingual German dictionaries. Their finding is that in many cases there is little agreement between the dictionaries.

As an alternative, they created German word classes solely based on syntactic criteria. The principal requirement for a syntactic word class system can be formulated as follows: If a word in a sentence is replaced by another word of the same class, then the syntax of the sentence must remain correct.

Applying this criterion is not trivial for two reasons: 1) many words are ambiguous and can belong to several classes; 2) it is often possible that a word in a sentence can be replaced by a word of another class, changing the syntax, but not corrupting it.[4] Nevertheless, the popularity and success of part-of-speech tagging show that it is feasible and useful to construct syntactically oriented word classes (= tag systems).

As a native speaker can easily judge whether or not a word in a given sentence can be replaced by a

[4] For example, in the sentence *"John drives."* the personal pronoun *he* can substitute for the proper noun *John*.

certain other word without affecting the grammaticality of the sentence, knowledge of word classes seems to be acquired implicitly when learning a language. Now the question is whether this process can be simulated on a machine; i.e., whether it is possible to derive word class information from corpora through unsupervised learning. Numerous papers have dealt with this problem (e.g. Brown et al., 1992; Kneser & Ney, 1993), with considerable success. We want to describe here an approach suggested in a previous publication (Rapp, 1996), with an emphasis on the vector similarity method.

The approach is based on the observation that in texts, words of the same class often have the same neighbors. For example, nouns are often preceded by articles like *the* and *a*, and succeeded by frequent verbs like *is* or *has*. Our claim is now that words belonging to the same class have similar frequency distributions of their direct neighbors.

The appropriate means of representing the co-occurrence frequencies of a word with its neighbors is a matrix. In order to illustrate this, table 5 shows a small example derived from the 1-million-word Brown corpus of present-day American English.

	left neighbors				right neighbors			
	has	is	the	to	has	is	the	to
go	0	2	7	198	0	0	3	107
have	0	0	0	339	0	3	157	274
man	0	0	184	6	14	38	3	35
car	0	0	91	0	0	8	0	3
work	0	1	110	99	7	23	4	12

Table 5: Bigram frequencies in the Brown corpus.

	left neighbors				right neighbors			
	has	is	the	to	has	is	the	to
go	0	0	4	83	0	2	28	94
have	0	0	0	91	0	0	39	77
man	0	3	98	0	41	58	0	23
car	0	0	76	0	38	63	2	18
work	23	18	68	73	24	36	9	25

Table 6: The author's guesses of bigram frequencies.

The maxima in the matrix indicate some of the more salient word bigrams. As can be expected, nouns are frequently preceded by articles, verbs by the infinitive marker *to*. *To* also often follows the verbs *have* and *go*. However, in the latter case it does not function as an infinitive marker but as a preposition. With its frequent left neighbors being *the* and *to*, the ambiguous word *work* shows the behavior of both verb and noun.

As a little digression, table 6 shows the co-occurrence estimates of the author as obtained by guessing before the counts were collected from the Brown corpus. Although there are of course differences to table 5, the maxima and minima in both tables are often at corresponding positions, which shows that in guessing the frequencies of word bigrams humans seem to be considerably better than chance.[5]

Having looked at a small sample co-occurrence matrix for word bigrams, let us now consider what results we obtain with a matrix of all words occurring in a larger corpus (see also Rapp, 1996). We compiled our corpus by putting together several corpora from the Institut für deutsche Sprache (IDS), namely the *Mannheimer Korpus* (2.2 million words of written German), the *Handbuchkorpora* (9.3 million words of newspaper text) and the *Freiburger Korpus* (0.5 million words of spoken language).

In this corpus of 12 million words we found approximately 450 000 different word tokens. In accordance with table 5, we constructed a matrix of size 900 000 × 450 000. However, unlike table 5, this matrix was not filled with absolute but with relative bigram counts. That is, each bigram count as collected from the corpus was divided by the corpus frequency of the word assigned to the respective line of the matrix. Using relative instead of absolute frequencies should eliminate the word frequency effect; i.e., comparisons between words of different corpus frequency should be improved.

Our assumption is now that application of the city block metric to pairs of lines in the matrix (as

[5] However, this observation does not necessarily mean that co-occurrence counts are stored in our memory. Another possible explanation would be that on the basis of rules we try to generate phrases that include a certain bigram, and that if this is possible we tend to give high estimates for its frequency. Although the latter view is supported by introspection, our belief is that the two mechanisms are not mutually exclusive, and that both may have their place in our brain.

described in the previous section) should lead to word similarities that reflect the syntactic behavior of a word. We expect high similarities for words that belong to the same word class, and low similarities otherwise.

Table 7 shows some results by listing the top most similar words as obtained for a small selection of German word forms belonging to different word classes. As computed, the most similar words to the preposition *auf* are other prepositions, to the participle *gegessen* other participles, to the comparative form *schneller* other comparative forms, and to the numeral *zwei* other numerals. That is, the method not only takes the part of speech of a word into account, but also a diversity of syntactic features.

Another example is given in table 8, where the top 30 most similar words to the German conjunction *bevor* (Engl. before) are listed. In the word class system suggested by Bergenholtz & Schaeder (1977), *bevor* is classified as "hypotaktische Konjunktion" (subordinating conjunction). In their book, Bergenholtz & Schaeder tried to give complete lists of the word classes they considered as closed. However, when comparing their list to table 8, it becomes apparent that *obgleich* and *wenngleich* are missing, two words which also should be classified as subordinating conjunctions.

auf	an in über mit durch unter aus für vor um nach bei gegen von auch
bitten	lernen schreiben denken holen suchen erzählen erleben lachen
gegessen	getrunken gelesen gekauft getan geredet gerettet erworben gesungen
hell	weich jung blaß dunkel hübsch naiv kalt arm grau warm dünn
Müller	Schmidt Fischer Meyer Hoffmann Wagner Schiller Schneider Koch
Thomas	Peter Michael Klaus Herbert Franz Karl Hermann Stephan Alexander
Tisches	Stücks Abends Bundes Friedens Autors Krieges Staates Alltags
schneller	besser stärker billiger größer älter leichter mehr höher schlimmer
zwei	drei vier fünf sechs acht sieben zwölf neun zehn elf zwanzig

Table 7: Similarities based on word neighborhoods.

bevor	ehe weil nachdem sobald wenn ob obwohl falls obgleich womit wo sofern worin wofür weshalb indem wovon worauf daß solange wobei woran obschon wenngleich wohin zumal was wodurch inwieweit soweit

Table 8: Similarities to *bevor*. Underlined words are missing in the list of subordinating conjunctions compiled by Bergenholtz & Schaeder (1977).

Although our results look fairly good, we of course do not claim to have solved the problem of word classification. An obvious difficulty to our method are ambiguous words such as *can* (verb) and *can* (noun). If, such as in this case, one meaning is much more frequent than the other, then this meaning tends to be dominant. If both meanings are of similar frequency, such as with *work* (verb) and *work* (noun), then the saliency of belonging to each of the two classes may be reduced. We consider this as one of the principal problems at the present state of the art in statistical natural language processing, as will be set out in section 6.

4 Word Meaning

According to Ruge (1995) the semantic similarity of two words can be computed by determining the agreement of their lexical neighborhoods. For example, the semantic similarity of the words *red* and *blue* can be derived from the fact that they both frequently co-occur with words like *color*, *flower*, *dress*, *car*, *dark*, *bright*, *beautiful*, and so forth. If for each word in a corpus a co-occurrence vector is determined whose entries are the co-occurrences with all other content words in the corpus, then the semantic similarities between words can be computed by conducting simple vector comparisons. To determine the words most similar to a given word, its co-occurrence vector is compared to the co-occurrence vectors of all other words in a vocabulary using one of the standard similarity measures; for example, the city block metric. Those words that obtain the best values are considered to be most similar. Practical implementations of algorithms based on this principle have led to excellent results as documented in publications by Grefenstette (1994), Agarwal (1995), Ruge (1995), Landauer & Dumais (1997), Schütze (1997), and Lin (1998).

We chose to extract our co-occurrence counts from the British National Corpus (BNC), a 100-million-word corpus of written and spoken language that was compiled with the intention of providing a representative sample of British English. Since function words are not required for the analysis of word meaning, to save disk space and processing time we decided to remove them from the text. This was done on the basis of a list of approximately 200 English function words. We also lemmatized the corpus (for details on the lemmatization process see Lezius et al., 1998, and Rapp, 1999). This not only reduces the sparse data problem but also significantly reduces the size of the co-occurrence matrix to be computed.

For counting word co-occurrences, as in most other studies a fixed window size is chosen and it is determined how often each pair of words occurs within a text window of this size. Choosing a window size usually means a trade-off between two parameters: specificity versus the sparse data problem. The smaller the window, the stronger the associative relation between the words inside the window, but the more severe the sparse data problem. In our case, with ±1 word, the window size appears rather small. However, this can be justified since we have reduced the effects of the sparse data problem by using a large corpus and by lemmatizing the corpus. It also should be noted that a window size of ±1 applied after elimination of the function words is comparable to a window size of ±2 without elimination of the function words (assuming that roughly every second word is a function word).

Based on the window size of ±1, we computed a co-occurrence matrix of about a million words in the lemmatized BNC. Although the resulting matrix of 10^{12} entries is extremely large, this was feasible since we used a sparse format that does not store zero entries. In order to emphasize significant word pairs, we applied the log-likelihood ratio as proposed by Dunning (1993) to all values in the matrix (see Rapp, 1999, for details). By using the city block metric as in the previous two experiments, we then computed the most similar words to a list of 80 test words. Table 9 shows the results for six of the 80 test words. As can be seen from the table, all of the most similar words seem to be of related meaning to the stimulus word. Also, as can be expected from the experiment on word classification, the most similar words are in almost all cases of the same part of speech as the stimulus word.

blue	cold	fruit	green	tobacco	whiskey
red	hot	food	red	cigarette	whisky
green	warm	flower	blue	alcohol	brandy
grey	dry	fish	white	coal	champagne
yellow	drink	meat	yellow	import	lemonade
white	cool	vegetable	grey	textile	vodka

Table 9: Most similar words to six test words.

Since it is more difficult to intuitively assess the quality of the results in the case of word meaning than it was with string similarity or word classification, we decided to conduct a quantitative evaluation of the results by comparing them to similarity estimates obtained by human subjects. Such data was kindly provided by Thomas K. Landauer, who had taken it from the synonym portion of the *Test of English as a Foreign Language* (TOEFL). Originally, the data came, along with normative data, from the Educational Testing Service (Landauer & Dumais, 1997). The TOEFL is a test required for admittance at many universities worldwide where the teaching language is English.

The TOEFL data comprises 80 test items. Each item consists of a problem word in testing parlance and four alternative words, from which the test taker is asked to choose the one with the most similar meaning to the problem word. For example, given the test sentence "*Both boats and trains are used for transporting the materials*" and the four alternative words *planes*, *ships*, *canoes*, and *railroads*, the subject would be expected to choose the word *ships*, which is the one considered most similar to *boats*. Since we had lemmatized our corpus, we also had to lemmatize the test data, so that in the above example singular instead of plural forms would be used.

For our evaluation, we assumed that the system had chosen the correct alternative if the correct word was ranked highest among the four alternatives. This was the case for 55 of the 80 test items, which gives us an accuracy of 69%. This accuracy may seem low, but it should be taken into account that the TOEFL tests the language abilities of prospective university students and therefore is relatively difficult. Actually, the performance of the average human test taker was worse than the performance of

the system. The human subjects were only able to solve 51.6 of the test items correctly, which gives an accuracy of 64.5%. Please note that in the TOEFL, average performance (over several types of tests, with the synonym test being just one of them) admits students to most universities; i.e., although the majority of the test takers does not have a native command of English, their language ability is fairly good. Another consideration is the fact that our simulation program was not designed to make use of the context of the test word, so it neglected some information that may have been useful for the human subjects.

To add some context from psycholinguistics, let us mention that – as convincingly shown by Landauer & Dumais (1997) – the ability to compute semantic similarities between words can be considered equivalent to the acquisition of the lexicon by children: a new word is added to the lexicon when its relations to other words are supported by sufficient evidence from a corpus of perceived language.[6]

5 Comparison of the Three Methods

The three methods discussed in the previous sections have a great deal in common. In all cases, normalized co-occurrence counts from a corpus are stored in a matrix and vector similarities are computed using the same similarity metric. Also, they are all of second order type. This means, to determine the similarity between two words it is not sufficient to look at a single value in the matrix (as is the case when extracting collocations or word associations from text, see Smadja, 1993, and Rapp, 1996), but that all values in their feature vectors must be taken into account.

The main difference between the first method (string comparison) and the other two is that it considers characters instead of words. The essential difference between the second method (word classification) and the third (word meaning) is that – although they both use the same window size of ±1 word from the stimulus[7] – function words were removed from the corpus only in the latter case. This little – often neglected – distinction seems to make the difference between the computation of word classes and word meaning.

Since the simulations described in this paper have been conducted over a period of several years, there are other differences in the procedures which in our view are not responsible for the different outcomes of the simulations. These concern the use of different corpora, different corpus processing (lemmatization or not), different normalization methods for the co-occurrence counts (absolute frequencies, relative frequencies, or log-likelihood ratio). We believe that had we omitted lemmatization and used relative frequencies or log-likelihood ratios in all cases, the results would not have deteriorated significantly.

6 Conclusions and Further Work

We showed that a matrix representation of features extracted from a corpus combined with the application of a vector similarity metric allowed simulating the intuitions of human subjects for three types of problems. The success of this approach is insofar surprising, as the three problems looked very different at first glance.

The essential difference in simulating the three tasks was not the core algorithm itself but the preprocessing because different statistics have been extracted from the corpus and different features were assigned to the columns of our matrix. This is analogous to connectionism, where a three layer network using the backpropagation algorithm can be used so solve almost any problem, and where the main task of the researcher is to decide what features from his problem space to choose and how to assign them to the cells in the input and output layers of the network.

Two problems that we consider essential for a deeper understanding and more successful simulation of human language processing have not been dealt with in the framework of this paper: on one hand we see a need to resolve ambiguities before computing vector similarities; on the other, the se-

[6] Landauer & Dumais even showed that the rate of a child's vocabulary acquisition, about 7 to 15 new words per day, can be explained this way. However, as described in Rapp (2002), we disagree with them insofar as they ascribe this result to their method of latent semantic indexing, which in our view only functions as a smoothing method here.

[7] Please note that only for word classification left and right neighbors obtained separate matrix entries.

quential and often recursive character of language must be better accounted for. Our attempts to solve these fundamental shortcomings of our work, to be presented in forthcoming papers, involve anticipating possible ambiguities through massive computational effort to address the first problem, and the use of replacement operators (i.e. operators that transform a word or sentence into a different but similar word or sentence) as features for our matrices to address the second problem.

Let us come back to the introductory question as to the inner workings of our brain: for us, the deeper reason for our findings is that the brain probably uses identical mechanisms at different levels of language processing, and that one of these mechanisms seems to be analogous to the computation of similarities between feature vectors. From a biological point of view this makes sense, since the process of evolution generally gives preference to the reuse, adaptation, and optimization of existing proven mechanisms over the creation of new mechanisms.[8]

Acknowledgements

I would like to thank Manfred Wettler for his help and advice, Sandra Kübler for valuable comments on memory-based learning, and the DFG for financially supporting this research.

References

Angell, R.C.; Freund, G.E.; Willett, P. (1983). Automatic spelling correction using a trigram similarity measure. *Information Processing & Management* 19(4), 255–261.

Agarwal, R. (1995). *Semantic Feature Extraction from Technical Texts with Limited Human Intervention*. Dissertation, Mississippi State University.

Bergenholtz, H.; Schaeder, B. (1977). *Die Wortarten des Deutschen.* Stuttgart: Klett.

Brockhaus (2002). Der Brockhaus in Text und Bild Edition 2002 (CD-ROM). Mannheim: Bibliographisches Institut & F. A. Brockhaus AG.

Brown, P., deSouza, P.V., Mercer, R.L., Della Pietra, V.J., Lai, J.C. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, 18(4), 467–479.

[8] A good example for this observation is – among others – the mechanism of reproduction common to most living beings.

Daelemans, W.; Zavrel, J.; van der Sloot, K.; van den Bosch, A. (2002). *TiMBL: Tilburg Memory Based Learner, Version 4.2, Reference Guide*. ILK Technical Report 02-01. http://ilk.kub.nl

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1), 61–74.

Grefenstette, G. (1994). *Explorations in Automatic Thesaurus Discovery*. Dordrecht: Kluwer.

Kneser, R.; Ney, H. (1993). Forming word classes by statistical clustering for statristical language modelling. In: R. Köhler, B.B. Rieger (eds.): *Contributions to Quantitative Linguistics.* The Netherlands: Kluwer, 221–226.

Landauer, T. K.; Dumais, S. T. (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.

Lezius, W.; Rapp, R.; Wettler, M. (1998). A freely available morphology system, part-of-speech tagger, and context-sensitive lemmatizer for German. In: *Proceedings of COLING-ACL 1998*, Montreal, Vol. 2, 743-748.

Lin, D. (1998). Automatic Retrieval and Clustering of Similar Words. *Proceedings of COLING-ACL 1998*, Montreal, Vol. 2, 768–773.

Manning, C.; Schütze, H. (1999). *Foundations of Statistical Natural Language Processing.* Cambridge, MA: MIT Press.

Peterson, J.L. (1980). Computer programs for detecting and correcting spelling errors. *Communications of the ACM* 23(12), 676–687.

Rapp, R. (1996). *Die Berechnung von Assoziationen.* Hildesheim: Olms.

Rapp, R. (1999). Automatic identification of word translations from unrelated English and German corpora. *Proceedings of ACL 1998*, College Park. 519–526.

Rapp, R. (2002). The computation of word associations: comparing syntagmatic and paradigmatic approaches. *Proceedings of COLING 2002, Taipei.*

Ruge, G. (1995). *Wortbedeutung und Termassoziation.* Hildesheim: Olms.

Salton, G.; McGill, M. (1983). *Introduction to Modern Information Retrieval*. New York: McGraw-Hill.

Schütze, H. (1997). *Ambiguity Resolution in Language Learning: Computational and Cognitive Models.* Stanford: CSLI Publications.

Smadja, F. (1993). Retrieving collocations from text: Xtract. *Computational Linguistics* 19(1), 143–177.