Word Sense Discovery Based on Sense Descriptor Dissimilarity

Reinhard Rapp

Johannes Gutenberg-Universität Mainz, FASK
D-76711 Germersheim, Germany
rapp@mail.fask.uni-mainz.de

Abstract

In machine translation, information on word ambiguities is usually provided by the lexicographers who construct the lexicon. In this paper we propose an automatic method for word sense induction, i.e. for the discovery of a set of sense descriptors to a given ambiguous word. The approach is based on the statistics of the distributional similarity between the words in a corpus. Our algorithm works as follows: The 20 strongest first-order associations to the ambiguous word are considered as sense descriptor candidates. All pairs of these candidates are ranked according to the following two criteria: First, the two words in a pair should be as dissimilar as possible. Second, although being dissimilar their co-occurrence vectors should add up to the co-occurrence vector of the ambiguous word scaled by two. Both conditions together have the effect that preference is given to pairs whose co-occurring words are complementary. For best results, our implementation uses singular value decomposition, entropy-based weights, and second-order similarity metrics.

1 Introduction

Whereas programming languages are unambiguous by design, natural languages tend to be ambiguous at all levels of processing, e.g. at the phonological, the morphological, the syntactic, and the semantic levels. If we look at words, a fundamental problem is that when analyzing corpora we can only observe and study the complicated behavior of these ambiguous entities, whereas the presumably simpler behavior of some underlying unambiguous entities, i.e. the word senses, remains hidden.

Due to the importance of the problem, many publications have dealt with the task of ambiguity resolution. Concerning word semantics, important contributions have been made, for example, in the framework of SENSEVAL, a competition where a number of word sense disambiguation systems were evaluated and compared. Given an ambiguous word in context, the aim of these systems was to choose among a number of predefined senses the one that best described the semantic role of the word in the particular context.

In such systems, the sets of senses are usually taken from dictionaries such as Longman's Dictionary of Contemporary English (LDOCE), or from lexical databases such as WordNet. These dictionaries and databases are constructed manually by lexicologists and linguists.

However, only little effort has been made to derive the sets of possible senses automatically from corpora. In a recent paper, Pantel & Lin (2002) write: "To the best of our knowledge, there has been no previous work in automatic word sense discovery from text." Nevertheless, an overview on some literature has been given in a previous paper (Rapp, 2003a). We would like to supplement this overview by mentioning Neill (2002), Dorow & Widdows (2003), and Rapp (2003b).

The approach to word sense induction that we suggest here is best described by an example: Let us look at the ambiguous word *bank* with its *money* and *river* senses. We observe that the contexts of *bank* contain words representative for both meanings, e.g. *institute*, *interest*, and *account* for the *money* sense and *sand*, *water*, and *beach* for the *river* sense. We now assume that the meanings of an ambiguous word are best described by those of its significant associates whose features complement each other in an optimal way. In mathematical terms, we would expect that the co-occurrence vectors of the words describing the meaning of an ambiguous word although being dissimilar nevertheless add up to the co-occurrence vector of the ambiguous word. In the case of this example, the co-occurrence vectors of the words *money* and *river* should be as dissimilar as possible, but at the same time their (weighted) sum should be more

similar to the co-occurrence vector of *bank* than the sum of the co-occurrence vectors of any other pair of dissimilar words. It turns out that this approach works surprisingly well, as will be shown in the remainder of this paper.

2 Data

2.1 Corpus

Since our algorithm is based on a similarity measure relying on co-occurrence data, a corpus is required from which the co-occurrence counts can be derived. If – as in this case – a qualitative measure for the success of the system is the results' plausibility to human judgment, it is advisable to use a corpus that is as typical as possible for the language environment of native speakers.

We therefore chose to use 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 (Burnard & Aston, 1998).

Since function words were not considered important for our analysis of word semantics, 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 decided to lemmatize the corpus using the lexicon of full forms provided by Karp et al. (1992). This not only improves the sparse data problem but also significantly reduces the size of the co-occurrence matrix to be computed. Since most word forms are unambiguous concerning their possible lemmas, we only conducted a partial lemmatization that does not take the context of a word into account and thus leaves the relatively few words with several possible lemmas unchanged. This way we avoided the need for disambiguation which would have anticipated the purpose of this research.

2.2 Evaluation data

In order to quantitatively evaluate our results in sense induction we took the list of 12 ambiguous words used by Yarowsky (1995). Each of these words is considered to have two main senses, and for each sense he provides a word characteristic of that sense. Table 5 (first and second column) shows the list of words together with their sense descriptors.

Another kind of test data was used to evaluate our method for computing second-order word similarities. It comprises similarity estimates obtained from human subjects. This 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 an obligatory test for foreign students who would like to study at a university in an English speaking country.

The 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 most similar to *boats*. As with the corpus, the words in the test data were also lemmatized.

3 Algorithm

3.1 Distributional model of word semantics

As has been shown by Schütze (1997) and others, 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 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 the vocabulary using one of the standard vector similarity measures; for example, the cosine coefficient or the city-block metric. Those words that obtain the best scores are considered to be most similar.

3.2 Counting word co-occurrences

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 more salient the associative relations between the words inside the window, but the more severe the sparse data problem. In our case, with ±2 words, the window size looks 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 ±2 applied after elimination of the function words is comparable to a window size of ±4 applied to the original texts (assuming that roughly every second word is a function word).

Based on the window size of ±2, we computed the co-occurrence matrix for the corpus. By storing it as a sparse matrix, it was feasible to include all of the approximately 375 000 lemmas occurring in the BNC.

3.3 Computation of association strength

Although semantic similarities can be successfully computed based on raw word co-occurrence counts, the results can be improved when the observed co-occurrence-frequencies are transformed by some function that reduces the effects of different word frequencies. For example, by applying a significance test that compares the observed co-occurrence counts with the expected co-occurrence counts (e.g. the log-likelihood test; see Dunning, 1993) significant word pairs are strengthened and incidental word pairs are weakened. Other measures applied successfully include TF/IDF and mutual information (Manning & Schütze, 1999). In the remainder of this paper, we refer to co-occurrence matrices that have been transformed by such a function as association matrices. However, in order to further improve similarity estimates, in this study we are applying a *singular value decomposition* (SVD) to our co-occurrence matrices (see section 3.5). To our surprise, our experiments clearly showed that the log-likelihood test, which was the transformation function that gave very good similarity estimates without SVD, was not optimal when using SVD. Following Dumais (1990) and Landauer & Dumais (1997) we found that with SVD some entropy-based transformation function gave substantially better results than the loglikelihood test. This is the formula that we use:

$$A_{ij} = \log(1 + f_{ij}) \cdot \left(- \sum_k p_{kj} \log(p_{kj}) \right)$$

with

$$p_{kj} = \frac{f_{kj}}{c_j}$$

Hereby f_{ij} is the co-occurrence frequency of words i and j and c_j is the corpus frequency of word j. Indices i, j, and k all have a range between one and the number of words in the vocabulary n. The right term in the formula (sum) is entropy. As usual with entropy, it is assumed that 0 log(0) = 0. The entropy of a word reaches its maximum of log(n) if the word co-occurs equally often with all other words in a vocabulary, and it reaches its minimum (zero) if it co-occurs only with a single other word.

In the information retrieval literature the transformation performed by the right part of the formula (entropy) is called *global weighting* since the same value is assigned to an entire column of the co-occurrence-matrix (Dumais, 1990). This value can be interpreted as a measure for the overall importance of a word. In contrast, the transformation performed by the left part ($\log(1 + f_{ij})$) is called *local weighting*.

Let us now look at how the formula works. The important part is taking the logarithm of f_{ij} thus dampening the effects of large differences in frequency. Adding 1 to f_{ij} provides some smoothing and prevents the logarithm from becoming infinite if f_{ij} is zero. A relatively modest, but noticeable improvement[1] can be achieved by multiplying this by the entropy of a word. This has the effect that the weights of rare words that have only few (and often incidental) co-occurrences are reduced.

Please note that this is in contrast to Landauer & Dumais (1997) and Dumais (1990) who suggest not to multiply but to divide by entropy. The argument is that words with a salient co-occurrence distribution should have stronger weights than words with a more or less random distribution. However, as shown empirically, in our setting

[1] In the order of 5% when measured using the TOEFL-data, see section 4.

multiplication leads to clearly better results than division.

3.4 Computation of semantic similarity

The computation of the semantic similarities between words is based on comparisons between their co-occurrence vectors. Our experience is that the sparse data problem is usually by far not as severe for the computation of vector similarities (second-order dependency) as it is – for example – for the computation of mutual information (first-order dependency). The reason is that for the computation of vector similarities a large number of co-occurrence values are taken into account, and although each value is subject to a sampling error, these errors tend to cancel out over the whole vector. Since association measures such as mutual information usually only take a single co-occurrence-value into account, this kind of error reduction cannot be observed in this case.

For vector comparison, among the many similarity measures found in the literature (Salton & McGill, 1983) we usually use the cosine coefficient and the city block metric. The cosine coefficient computes the cosine of the angle between two vectors X and Y – both of length n – as follows:

$$a = \frac{\sum_{i=1}^{n}(X_i \cdot Y_i)}{\sqrt{\sum_{i=1}^{n} X_i^2 \cdot \sum_{i=1}^{n} Y_i^2}}$$

The city block metric computes the distance between two vectors as the sum of the absolute differences of corresponding vector positions:

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

Although the city-block metric is the simpler measure and computationally less demanding, its results are usually as good as those achieved with the cosine coefficient if vectors are normalized before comparison (sum of entries = 1). This is what we always do when using it.

3.5 Singular value decomposition

Landauer & Dumais (1997) showed that the results can be improved if before computing semantic similarities the dimensionality of the association matrix is reduced. An appropriate mathematical method to do so is singular value decomposition. As this method is rather sophisticated, we can not go into the details here. Good descriptions can be found in Landauer & Dumais (1997), Manning & Schütze (1999), and Press et al. (1992). The essence is that by computing the Eigenvalues of a matrix and by truncating the smaller ones, SVD allows to significantly reduce the number of columns, thereby (in a least squares sense) optimally preserving the Euclidean distances between the lines (Schütze, 1997:191). The resulting columns are equivalent to the so called *principal components* in the better known formalism of *principal component analysis* (which is applicable to square matrices only).

For computational reasons, we were not able to conduct the SVD for a matrix of all 374244 lemmas occurring in the BNC.[2] Therefore, we restricted our vocabulary to all lemmas occurring at least 20 times in the BNC. To this vocabulary all problem and alternative words occurring in the TOEFL synonym test were added. This resulted in a total vocabulary of 56491 words. In the association matrix corresponding to this vocabulary all 395 lines and 395 columns that contained only zeroes were removed which led to a matrix of size 56096 by 56096.

By using a version of Mike Berry's SVDPACK-Software that had been modified by Hinrich Schütze, we transformed the 56096 by 56096 association matrix to a matrix of 56096 lines and 300 columns. This smaller matrix has the advantage that all subsequent similarity computations are much faster. As discussed in Landauer & Dumais (1997), the process of dimensionality reduction by combining similar columns (relating to words with similar meanings) is believed to perform a kind of generalization that is hoped to improve similarity computations (even critics concede at least a smoothing effect).

As an example, let us assume that word *A* frequently co-occurs with *car* and *shop*, and that word *B* often co-occurs with *automobile* and *store*. Intuitively, one would say that *A* and *B* should be semantically related, since the meanings of their frequent neighbors are similar. However, since in this case it happened that the same meanings were

[2] The main reason for this is that the 32 bit operating system that we used for SVD (standard version of Windows XP) has a limitation of at most 2 GB (= 2^{31} Bytes) of memory per application.

expressed by different words, the computed similarity is zero. However, through SVD it is likely that synonyms like *car* and *automobile* or *shop* and *store* end up in the same columns, and therefore – as desired – a high similarity will be computed between *A* and *B*.

3.6 Procedure for sense induction

As stated before, our core assumption is that good descriptors for the senses of a word are those of its associates that fulfill the condition of complementarity. We express complementarity by a vector summation and by the dissimilarity of the descriptors.

This means that given a two-fold ambiguous word *A*, word *X* and word *Y* are then good descriptors for the two senses of *A* if their co-occurrence vectors are dissimilar and if the sum of the two vectors is similar to the co-occurrence vector of *A* scaled by two. An example with binary vectors is shown in table 1.

	money		river		bank
account	♦				♦
cash	♦				♦
music					
sand		+	♦	=	♦
savings	♦				♦
shore			♦		♦
water			♦		♦

Table 1: The vector of an ambiguous word can be considered as the sum of its sense descriptor vectors.

In mathematical terms, those two descriptor vectors *X* and *Y* that maximize the following expression fulfill the complementarity criterion *c* best:

$$c = \frac{d(X, Y)}{d(X + Y, 2A)}$$

Hereby *d(X,Y)* is the distance between the two vectors *X* and *Y*. Assuming the city block metric (to be applied to normalized vectors) we obtain:

$$c = \frac{\sum_{i=1}^{n} |X_i - Y_i|}{\sum_{i=1}^{n} |X_i + Y_i - 2A_i|}$$

The problem with our approach is that the sense descriptors are unknown. We therefore need to generate possible pairs of sense descriptors, then compute the complementarity score for each of them, and finally choose the one that maximizes complementarity.

In principle, given a certain vocabulary, all possible pairs of words could be generated and evaluated. However, this may not be computationally feasible for a large vocabulary. Also, there is always some risk that due to the sparse data problem the vectors of some low frequency words may incidentally fulfill the complementarity requirement.

Both problems can be avoided by limiting the choice of sense descriptors to a small number of candidates which are strongly associated with the ambiguous word. It is an interesting question whether first or second-order associations should be used as candidates. Our preference is on first-order associations because we noticed that second-order associations tend to only reflect the prevailing sense of a word, whereas first-order associations usually reflect all senses. (As an example, compare the first- and second-order associations to *poach* in tables 2 and 3.) We concede that it may be less likely for first-order associations, which are often of syntagmatic type, to fulfill the complementarity requirement; but since it can not be ruled out they should be admitted as candidates.

These are the details of our implemented algorithm for sense induction: Using the log-likelihood ratio (Dunning, 1993) with a window size of ±2 words from the stimulus we compute the top 20 first-order associations to a given word.[3] We then generate all 190 possible pairs of these words and compute the complementarity score for each pair. Please note that the complementarity computation is performed on vectors of entropy-based weights whose dimensionality had been reduced to 300 entries. The words in the pair with the highest score are considered to be optimal descriptors for the two main senses of the given word.

4 Evaluation of similarity estimates

Our method for sense induction very much depends on the quality of our estimates of the semantic similarities between words. Therefore, before looking at the results for sense induction, let

[3] As, according to human judgment, low frequency words are usually not considered as plausible associations, we introduced a frequency threshold of 100.

us first evaluate our measurements of word similarity.

AXES	BANK	DRUG	POACH
axe	banker	cocaine	salmon
axis	credit	heroin	butter
arrows	loan	psychotropic	bake
knife	banking	addiction	tuna
dagger	lend	alcohol	onion
sharpen	investment	cannabis	soup
vector	cash	addict	fillet
coordinates	Barclays	narcotic	fry
sword	deposit	antibiotic	mushroom
intersection	finance	tranquillizer	chicken
diagonal	investor	amphetamine	sauce
bladed	EIB	ecstasy	broccoli
tangent	lender	illegal	pork
flint	issuer	pills	prawn
z	Lyonnais	morphine	salad
symmetry	funds	LSD	steak
ellipse	financial	trafficker	turkey
eqn	bond	illicit	tortilla
pottery	cheque	depressant	tomato
chisel	IFC	addictive	mince

Table 2: Second-order associations as computed.

To give a first impression, table 2 shows the top 20 most similar words to some of Yarowsky's ambiguous words as computed using SVD, the cosine-coefficient, and a vocabulary of 56491 words (see section 3.5).[4] As can be seen from the table, in many cases the most similar words reflect only the prevailing sense of the ambiguous word.

Although these results look plausible, a quantitative evaluation is always desirable. For this reason we did a comparison of our results with the similarity estimates of the human subjects in the TOEFL task (section 2.2). Remember that the subjects had to choose the word most similar to a given stimulus word from a list of four alternatives.

In the simulation, we assumed that the system made the right decision if the correct word was ranked highest among the four alternatives. This was the case for 74 of the 80 test items which gives us an accuracy of 92.5%.[5] This compares to an average of 64.5% correct answers given by the human test takers who were prospective university students but – by definition – in most cases did not have a native command of English. 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. 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.

Let us compare our results to those reported by Landauer & Dumais (1997) in their seminal paper. Although we essentially used their method with only minor modifications, on exactly the same evaluation task they report an accuracy of only 64.4%.[6] We see three major reasons for the big discrepancy: First, our corpus is much larger than theirs (100 million words versus 4.7 million words) and more balanced (BNC versus Grolier's Encyclopedia). Second, we lemmatized the corpus and removed the function words whereas they did not. And third, while – as is common practice in information retrieval – their computations are based on a term/document-matrix with an average document length of 151 words, we used a co-occurrence matrix based on a much smaller window size of only ±2 words.

Of lesser importance is probably the discrepancy in the association formula used (section 3.3) and the fact that they reduced their matrix to 200 dimensions whereas we obtained best results using 300 dimensions.[7] This indicates that in order to take the richer information from a larger corpus into account, it may be advisable to use more dimensions.

Let us mention that the restriction of our vocabulary to the roughly 56000 words with a corpus frequency of at least 20 (which was necessary for performing the SVD) probably did not have any negative impact on the results. This can be concluded from the fact that even with a stronger restriction to a vocabulary of only half that size (approx. 28000 words) the results did not (yet) degrade but remained at an accuracy of 92.5%.

[4] A CD-ROM with a PC-version of the program giving comprehensive similarity lists is available on request.

[5] So far, our best result without SVD was 69%.

[6] Since this figure includes an asymmetric kind of correction for unknown words (Landauer & Dumais 1997:220) that we consider inadequate and therefore did not apply here (it corrects for unknown words only if it is advantageous to the performance), the actual accuracy that should be compared to ours is 62.5%.

[7] With 200 dimensions we obtained an accuracy of 85%.

5 Results for sense induction

Let us first look at the results for two example words taken from the list used by Yarowsky (1995), namely the words *axes* and *poach*. As listed in table 3, the 20 strongest first-order associations to these words as computed using the log-likelihood ratio are – as desired – a good mix of the two main senses for each word.[8] Our subjective choice of sense-assignments to the associations is indicated typographically.

The next steps are that we consider the computed 20 associations as sense descriptor candidates, that we generate all possible pairs of these candidates, and that by using our complementarity measure we finally evaluate each pair. Table 4 shows the ranked lists that we obtain for our example words.

The results look encouraging. Among the pairs with the top 15 complementarity scores the two words in a pair almost always belong to different senses (different typography), whereas for the 15 pairs with the lowest complementarity scores the words in a pair are semantically related and therefore usually belong to the same sense (same typography).

Let us now look at the results for all words found in the list of Yarowsky (see table 5). For each word the sense descriptor pair with the highest complementarity score is shown. Pairs that we deem to reflect the senses given by Yarowsky are printed in italics. This is the case for 7 out of 12 pairs which gives us an accuracy of 58.3%. In the other cases, the computed descriptors either both belong to the same sense, or – as for example with *tank* – do not well distinguish the senses.

AXES *grid* – tools	*Cartesian* stone *horizontal* y spear *vertical along x double z* grind *rotation graph* ice knife sword flint *principal* jade *intermediate*
POACH *steal* – boil	egg salmon *anti rhino upsurge* toast *ivory* boil *poacher* mackerel Kenya custard simmer kipper *smuggle* blanch *elephant tusk* chicken *firm*

Table 3: The top 20 strongest first-order associations to *axes* and *poach*. Typography (normal and italics) indicates our sense assignment.

[8] Please compare this to the second-order associations listed in table 2 where the sense prevailing in the corpus tends to strongly dominate.

	AXES *grid* – tools	POACH *steal* – boil
1	*z* flint	*rhino* boil
2	grind *graph*	*rhino* custard
3	spear *graph*	boil *smuggle*
4	*graph* flint	egg *smuggle*
5	*y* knife	salmon blanch
6	*y* flint	custard *smuggle*
7	*x* sword	*rhino* chicken
8	*horizontal* sword	egg *rhino*
9	*rotation* knife	*rhino* toast
10	*z* knife	*enya* simmer
11	*y* spear	egg *poacher*
12	*graph* sword	salmon toast
13	stone *graph*	salmon *smuggle*
14	*horizontal* spear	salmon *rhino*
15	*along* sword	*rhino* simmer
:	:	:
176	stone jade	salmon chicken
177	*double principal*	egg chicken
178	*vertical x*	boil chicken
179	ice *intermediate*	*Kenya elephant*
180	*x double*	toast custard
181	stone grind	*poacher tusk*
182	*double intermediate*	*ivory elephant*
183	stone flint	*rhino tusk*
184	spear knife	*rhino elephant*
185	knife sword	salmon mackerel
186	*y z*	boil custard
177	*y x*	*rhino ivory*
188	*x z*	*ivory tusk*
189	spear sword	*elephant tusk*
190	*horizontal vertical*	boil simmer

Table 4: Ranked list of the top and bottom 15 sense descriptor pairs for *axes* and *poach* as computed using the complementarity measure.

WORD	EXPECTED SENSES	COMPUTED SENSES
axes	grid – tools	*z – flint*
bass	fish – music	amp – violin
crane	bird – machine	neck – unload
drug	medicine – narcotic	*illegal – treatment*
duty	tax – obligation	fiduciary – officer
motion	legal – physical	*slow – amendment*
palm	tree – hand	*tree – outstretch*
plant	living – factory	*nuclear – shrub*
poach	steal – boil	*rhino- boil*
sake	benefit – drink	sacrifice – why
space	volume – outer	*shuttle – storage*
tank	vehicle – container	fish – petrol

Table 5: Sense induction results for the 12 ambiguous words listed by Yarowsky (1995). Sense pairs considered appropriate are printed in italics.

6 Discussion, conclusions, and prospects

The results that we presented are reasonably good, but of course far from perfect. On the one hand our system often produced word pairs that do not well reflect the two main senses of the ambiguous words as given by Yarowsky. On the other hand, the task of unsupervised sense induction is certainly difficult and research in this field is at an early stage. Unfortunately, we are not aware of related work close enough to ours that we could have compared our results with.

We see many reasons why our system did not perform as well as one might hope. First, the sense descriptors chosen by Yarowsky are somewhat arbitrary. Second, the corpus may not contain enough occurrences of one of the expected senses, as is probably the case for the *fish* meaning of *bass*. Third, it is not clear in how far SVD preserves the information required for sense induction. Finally, our approach does not take into account the syntactic usage, that is the one-sense-per-collocation-constraint formulated by Yarowsky (1995). When – as is the case for the example *sake* – one sense is much more frequent than the other, but there is a clear distinction in syntactic usage, then neglecting this distinction means neglecting crucial information.

Nevertheless, the approach looks promising, and current experiments using much larger corpora indicate that by only increasing the amount of data significant advances should be possible.

Acknowledgments

Part of this research was conducted during my stay at the Centre for Language Technology of Macquarie University, Sydney. I would like to thank Robert Dale and his research group for providing a stimulating environment and for giving valuable comments, Raz Tamir for further pursuing this research, Manfred Wettler for his continuous support, Hinrich Schütze for letting me have his version of Mike Berry's SVDPACK software, and the DFG (Deutsche Forschungsgemeinschaft) for financially supporting this work.

Bibliographical References

Burnard, L.; Aston, G. (1998). *The BNC Handbook: Exploring the British National Corpus with Sara*. Edinburgh University Press.

Dorow, B; Widdows, D. (2003): Discovering corpus-specific word senses. *EACL 2003*, Budapest, conference companion (research notes and demos), 79–82.

Dumais, S. (1990). *Enhancing Performance in Latent Semantic Indexing (LSI) Retrieval*. Technical Report TM-ARH-017527, Bellcore, Morristown (now Telcordia Technologies).

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

Karp, D.; Schabes, Y.; Zaidel, M.; Egedi, D. (1992). A freely available wide coverage morphological analyzer for English. In: *Proceedings of the 14th International Conference on Computational Linguistics*, Nantes, 950–955.

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.

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

Neill, D. B. (2002). *Fully Automatic Word Sense Induction by Semantic Cclustering*. Cambridge University, Master's Thesis, M.Phil. in Computer Speech.

Pantel, P.; Lin, D. (2002). Discovering word senses from text. In: *Proceedings of ACM SIGKDD*, Edmonton, 613–619.

Press, W.H.; Teukolsky, S.A.; Vetterling, W.T.; Flannery, B.P. (1992): *Numerical Recipes in C. The Art of Scientific Computing.* 2nd edition. Cambridge University Press.

Rapp, R. (2003a). Discovering the Meanings of an Ambiguous Word by Searching for Sense Descriptors with Complementary Context Patterns. In: *Actes des cinquiemes rencontres terminologie et intelligence artificielle,* Strasbourg, 145–155.

Rapp, R. (2003b). Die Erkennung semantischer Mehrdeutigkeiten mittels Unabhängigkeitsanalyse. In: U. Seewald-Heeg (ed.): *Sprachtechnologie für die multilinguale Kommunkation*. Sankt Augustin: Gardez, 338–355.

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.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In: *Proceedings of the 33rd Meeting of the Association for Computational Linguistics*, Cambridge, MA, 189–196.