Word Sense Induction: Triplet-Based Clustering and Automatic Evaluation

Stefan Bordag
Natural Language Processing Department
University of Leipzig
Germany
sbordag@informatik.uni-leipzig.de

Abstract

In this paper a novel solution to automatic and unsupervised word sense induction (WSI) is introduced. It represents an instantiation of the 'one sense per collocation' observation (Gale et al., 1992). Like most existing approaches it utilizes clustering of word co-occurrences. This approach differs from other approaches to WSI in that it enhances the effect of the one sense per collocation observation by using triplets of words instead of pairs. The combination with a two-step clustering process using sentence co-occurrences as features allows for accurate results. Additionally, a novel and likewise automatic and unsupervised evaluation method inspired by Schütze's (1992) idea of evaluation of word sense disambiguation algorithms is employed. Offering advantages like reproducability and independency of a given biased gold standard it also enables automatic parameter optimization of the WSI algorithm.

1 Introduction

The aim of word sense induction[1] (WSI) is to find senses of a given target word (Yarowski, 1995) automatically and if possible in an unsupervised manner. WSI is akin to word sense disambiguation (WSD) both in methods employed and in problems encountered, such as vagueness of sense distinctions (Kilgarriff, 1997). The input to a WSI algorithm is a target word to be disambiguated, e.g. *space*, and the output is a number of word sets representing the various senses, e.g. *(3-dimensional, expanse, locate)* and *(office, building, square)*. Such results can be at the very least used as empirically grounded suggestions for lexicographers or as input for WSD algorithms. Other possible uses include automatic thesaurus or ontology construction, machine translation or information retrieval. But the usefulness of WSI in real-world applications has yet to be tested and proved.

2 Related work

A substantial number of different approaches to WSI has been proposed so far. They are all based on co-occurrence statistics, albeit using different context representations such as co-occurrence of words within phrases (Pantel and Lin, 2002; Dorow and Widdows, 2003; Velldal, 2005), bigrams (Schütze, 1998; Neill, 2002; Udani et al., 2005), small windows around a word (Gauch and Futrelle, 1993), or larger contexts such as sentences (Bordag, 2003; Rapp, 2004) or large windows of up to 20 words (Ferret, 2004). Moreover they all employ clustering methods to partition the co-occurring words into sets describing concepts or senses. Some algorithms aim for a global clustering of words into concepts (Yarowski, 1995; Pantel and Lin, 2002; Velldal, 2005). But the majority of algorithms are based on a local clustering: Words co-occurring with the target word are grouped into the various senses the target word has. It is not immediately clear which approach to favor, however aiming at global senses has the inherent property to produce a uniform granularity of distinctions between senses that might not be desired (Rapp, 2004).

Graph-based algorithms differ from the majority of algorithms in several aspects. Words

[1] Sometimes called word sense discovery (Dorow and Widdows, 2003) or word sense discrimination (Purandare, 2004; Velldal, 2005)

can be taken as nodes and co-occurrence of two words defines an edge between the two respective nodes. Activation spreading on the resulting graph can be employed (Barth, 2004) in order to obtain most distinctly activated areas in the vicinity of the target word. It is also possible to use graph-based clustering techniques to obtain sense representations based on sub-graph density measures (Dorow and Widdows, 2003; Bordag, 2003). However, it is not yet clear, whether this kind of approach differs qualitatively from the standard clustering approaches. Generally though, the notion of sub-graph density seems to be more intuitive compared to the more abstract clustering.

There are different types of polysemy, the most significant distinction probably being between syntactic classes of the word (e.g. *to plant* vs. *a plant*) and conceptually different senses (e.g. *power plant* vs. *green plant*). As known from work on unsupervised part-of-speech tagging (Rohwer and Freitag, 2004; Rapp, 2005), the size of the window in which words will be found similar to the target word plays a decisive role. Using most significant direct neighbours as context representations to compare words results in predominantly syntactical similarity to be found. On the other hand, using most significant sentence co-occurrences results in mostly semantical similarity (Curran, 2003). However, whereas various context representations, similarity measures and clustering methods have already been compared against each other (Purandare, 2004), there is no evidence so far, whether the various window sizes or other parameters have influence on the *type* of ambiguity found, see also (Manning and Schütze, 1999, p. 259).

Pantel & Lin (2002) introduced an evaluation method based on comparisons of the obtained word senses with senses provided in WordNet. This method has been successfully used by other authors as well (Purandare, 2004; Ferret, 2004) because it is straightforward and produces intuitive numbers that help to directly estimate whether the output of a WSI algorithm is meaningful. On the other hand, any gold standard such as WordNet is biased and hence also lacks domain-specific sense definitions while providing an abundance of sense definitions that occur too rarely in most corpora. For example in the British National Corpus (BNC), the sense #2 of *MALE* (*[n] the capital of Maldives*) from WordNet is represented by a single sentence only. Furthermore, comparing results of an algorithm to WordNet automatically implies another algorithm that matches the found senses with the senses in WordNet. This is very similar to the task of WSD and therefore can be assumed to be similarly error prone. These reasons have led some researchers to opt for a manual evaluation of their algorithms (Neill, 2002; Rapp, 2004; Udani et al., 2005). Manual evaluation, however, has its own disadvantages, most notably the poor reproducability of results. In this work a pseudoword based evaluation method similar to Schütze's (1992) pseudoword method is employed. It is automatic, easy to reproduce and adapts well to domain specificity of a given corpus.

3 Triplet-based algorithm

The algorithm proposed in this work is based on the *one sense per collocation* observation (Gale et al., 1992). That essentially means that whenever a pair of words co-occurs significantly often in a corpus (hence a collocation), the concept referenced by that pair is unambiguous, e.g. *growing plant* vs. *power plant*. However, as also pointed out by Yarowsky (1995), this observation does not hold uniformly over all possible co-occurrences of two words. It is stronger for adjacent co-occurrences or for word pairs in a predicate-argument relationship than for arbitrary associations at equivalent distance, e.g. *a plant* is much less clear-cut. To alleviate this problem, the first step of the presented algorithm is to build triplets of words (target word and two of it's co-occurrences) instead of pairs (target word and one co-occurrence). This means that *a plant* is further restricted by another word and even a stop word such as *on* rules several possibilities of interpretation of *a plant* out or at least makes them a lot less improbable.

The algorithm was applied to two types of co-occurrence data. In order to show the influence of window size, both the most significant sentence-wide co-occurrences and direct neighbour co-occurrences were computed for each word. The significance values are obtained using the log-likelihood measure assuming a binomial distribution for the unrelatedness hypothesis (Dunning, 1993). For each word, only the 200 most significant co-occurrences were kept. This threshold and all others to follow were chosen after experiment-

ing with the algorithm. However, as will be shown in section 4, the exact set-up of these numbers does not matter. The presented evaluation method enables to find the optimal configuration of parameters automatically using a genetic algorithm.

The core assumption of the triplet-based algorithm is, that any three (or more) words either uniquely identify a topic, concept or sense. Using the previously acquired most significant co-occurrences (of both types), the lists of co-occurrences for all three words of a triplet are intersected to retain words contained in all three lists. If the three words cover a topic, e.g. *space, NASA, Mars*, then the intersection will not be empty, e.g. *launch, probe, cosmonaut, ...* If the three words do not identify a meaningful topic, e.g. *space, NASA, cupboard*, then the intersection will most likely contain few to no words at all. Intersections of triplets built from function words are very likely to contain many co-occurrences even if they do not identify a unique topic. These so-called 'stop words' are thus removed both from the co-occurrences from which triplets are built and from the co-occurrences which are used as features.

It is straightforward then to create all possible triplets of the co-occurrences of the target word w and to compute the intersection of their co-occurrence lists. Using these intersections as features of the triplets, it is possible to group triplets of words together that have similar features by means of any standard clustering algorithm. However, in order to 'tie' the referenced meanings of the triplets to the target word w, the resulting set of triplets can be restricted only to those that also contain the target word. This has the useful side-effect that it reduces the number of triplets to cluster. To further reduce the remaining number of $\binom{200}{2} = 19900$ items to be clustered, an iterative incremental windowing mechanism has been added. Instead of clustering all triplets in one step, 30 co-occurrences beginning from the most significant ones are taken in each step to build $\binom{30}{2} = 435$ triplets and their intersections. The resulting elements (triplets and intersections of their respective co-occurrences as features) are then clustered with the clusters remaining from the previous step.

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

- Target word is w
- for each step take the next 30 co-occurrences of w
  - Build all possible pairs of the 30 co-occurrences and add w to each to make them triplets
  - Compute intersections of co-occurrences of each triplet
  - Cluster the triplets using their intersections as features together with clusters remaining from previous step
    - Whenever two clusters are found to belong together, both the words from the triplets and the features are merged together, increasing their counts
- Cluster results of the loop by using the merged words of the triplets as features
- Classify unused words to the resulting clusters if possible

In order to reduce noise, for example introduced by triplets of unrelated words still containing a few words, there is a threshold of minimum intersection size which was set to 4. Another parameter

worth mentioning is that after the last clustering step all clusters are removed which contain less than 8 words. Keeping track of how many times a given word has 'hit' a certain cluster (in each merging step) enables to add a post-processing step. In this step a word is removed from a cluster if it has 'hit' another cluster significantly more often.

There are several issues and open questions that arise from this entire approach. Most obviously, why to use a particular similarity measure, a particular clustering method or why to merge the vectors instead of creating proper centroids. It is possible that another combination of decisions of this kind would produce better results. However, the overall observation is that the results are fairly stable with respect to such decisions whereas parameters such as frequency of the target word, size of the corpus, balance of the various senses and others have a much greater impact.

4 Evaluation

Schütze (1992) introduced a pseudoword-based evaluation method for WSD algorithms. The idea is to take two arbitrarily chosen words like *banana* and *door* and replace all occurrences of either word by the new pseudoword *bananadoor*. Then WSD is applied to each sentence and the amount of correctly disambiguated sentences is measured. A disambiguation in this case is correct, if the sentence like *I ate the banana* is assigned to sense #1 (banana) instead of #2 (door). In other words all sentences where one of the two words occurs are viewed as one set and the WSD algorithm is then supposed to sort them correctly apart. This, in fact, is very similar to the WSI task, which is supposed to sort the set of words apart that co-occur with the target word and refer to its different meanings. Thus, again it is possible to take two words, view their co-occurrences as one set and let the WSI algorithm sort them apart. For example, the word *banana* might have co-occurrences such as *apple, fruit, coconut, ...* and the word *door* co-occurrences such as *open, front, locked, ...* The WSI algorithm would therefore have to disambiguate the pseudoword *bananadoor* with the co-occurrences *apple, open, fruit, front, locked, ...*

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

It is further informative to measure **retrieval recall** (rR) - the amount of words that have been correctly retrieved into the correct sense. If, e.g., two words are merged into a pseudoword and the meaning of each of these two words is represented by 200 co-occurring words, then it could happen that one of the senses has been correctly found by the WSI algorithm containing 110 words with an overlap similarity of 0.91. That means that only 100 words representing the original sense were retrieved, resulting in a 50% retrieval recall. This retrieval recall also has an upper bound for two reasons. The average overlap ratio of the co-occurrences of the word pairs used for the evaluation was 3.6%. Another factor lowering the upper bound by an unknown amount is the fact that some of the words are ambiguous. If the algorithm correctly finds different senses of one of the two original words, then only one of the found senses will be chosen to represent the original 'meaning' of the original word. All words assigned to the other sense are lost to the other sense.

Using terms from information retrieval makes sense because this task can be reformulated as follows: Given a set of 400 words and one out of several word senses, try to retrieve all words belonging to that sense (retrieval recall) without retrieving any wrong ones (retrieval precision). A sense is then defined as correctly found by the WSI algorithm, if its retrieval precision is above 60% and retrieval recall above 25%. The latter number implies that at least 50 words have to be retrieved correctly since the initial co-occurrence sets contained 200 words. This also assumes that 50 words would be sufficient to characterize a sense if the WSI algorithm is not only used to evaluate itself. The reason to set the minimum retrieval precision to any value above 50% is to avoid a too strong

baseline, see below.

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

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

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

There is at least one possible baseline for the four introduced measures. One is an algorithm that does nothing, resulting in a single set of 400 co-occurrences of the pseudo-word. This set has a retrieval Precision rP of 50% compared to either of the two original 'senses' because for any of the two senses only half of the 'retrieved' words match. This is below the allowed 60% and thus does not count as a correctly found sense. This means that also retrieval Recall rR, Recall R are both 0% and Precision P in such a case (nothing correctly retrieved, but also nothing wrong retrieved) is defined to be 100%.

As mentioned in the previous sections, there are several parameters that have a strong impact on the quality of a WSI algorithm. One interesting question is, whether the quality of disambiguation depends on the type of ambiguity: Would the WSI based on sentence co-occurrences (and hence on the bag-of-words model) produce better results for two syntactically different senses or for two senses differing by topic (as predicted by Schütze (1992)). This can be simulated by choosing two words of different word classes to create the pseudoword, such as the (dominantly) noun *committee* and the (dominantly) verb *accept*.

Another interesting question concerns the influence of frequency of either the word itself or the sense to be found. The latter, for example, can be simulated by choosing one high-frequent word and one low-frequent word, thus representing a well-represented vs. a poorly represented sense.

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

- high frequent nouns (N_h): picture, average, blood, committee, economy
- medium frequent nouns (N_m): disintegration, substrate, emigration, thirst, saucepan
- low frequent nouns (N_l): paratuberculosis, gravitation, pharmacology, papillomavirus, sceptre
- high frequent verbs (V_h): avoid, accept, walk, agree, write
- medium frequent verbs (V_m): rend, confine, uphold, evoke, varnish
- low frequent verbs (V_l): immerse, disengage, memorize, typify, depute
- high frequent adjectives (A_h): useful, deep, effective, considerable, traditional
- medium frequent adjectives (A_m): ferocious, normative, phenomenal, vibrant, inactive
- low frequent adjectives (A_l): astrological, crispy, unrepresented, homoclinic, bitchy

These nine groups were used to design fours tests, each focussing on a different variable. The high frequent nouns are around 9000 occurrences, medium frequent around 300 and low frequent around 50.

4.1 Influence of word class and frequency

In the first run of all four tests, sentence co-occurrences were used as features. In the first test, all words of **equal word class** were viewed as one set of 15 words. This results in $\binom{15}{2} = 105$ possibilities to combine two of these words into

a pseudoword and test the results of the WSI algorithm. The purpose of this test is to examine whether there is a tendency for senses of certain word classes to be easier induced. As can be seen from Table 1, sense induction of verbs using sentence co-occurrences performs worse compared to nouns. This could be explained by the fact that verbs are less semantically specific and need more syntactic cues or generalizations - both hardly covered by the underlying bag-of-words model - in order to be disambiguated properly. At the same time, nouns and adjectives are much better distinguishable by topical key words. These results seem to be in unison with the prediction made by Schütze (1992).

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

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

In the second test, all three types of possible **combinations of the word classes** are tested, i.e. pseudowords consisting of a noun and a verb, a nouns and an adjective and a verb with an adjective. For each combination there are $15 \cdot 15 = 225$ possibilities of combining a word from one word class with a word from another word class. The purpose of this test was to demonstrate possible differences between WSI of different word class combinations. This corresponds to cases when one word form can be both a nound and a verb, e.g. *a walk* and *to walk* or a noun and an adjective, for example *a nice color* and *color TV*. However, the results in Table 2 show no clear tendencies other than perhaps that WSI of adjectival senses from verb senses seems to be slightly more difficult.

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

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

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

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

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

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

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

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

Finally it is possible to provide the averages for the entire test runs comprising 1980 tests. The macro averages over all tests are $P = 85.42\%$, $R = 72.90\%$, $rP = 86.83\%$ and $rR = 62.30\%$, the micro averages are almost the same. Using the same thresholds but only pairs instead of triplets

results in $P = 91.00\%$, $R = 60.40\%$, $rP = 83.94\%$ and $rR = 62.58\%$. Or in other words, more often only one sense is retrieved and the F-measures of $F = 78.66\%$ for triplets compared to $F = 72.61\%$ for pairs confirm an improvement by 6% by using triplets.

4.2 Window size

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

Although these results are insufficient to show the influence of context representations on the type of induced senses as they were supposed to, they allow several other insights. Firstly, corpus size does obviously matter for WSI as more data would probably have alleviated the sparseness problem. Secondly, while perhaps one context representation might be theoretically superior to another (such as neighbor co-occurrences vs. sentence co-occurrences), the effect various representations have on the data richness were by far stronger in the presented tests.

4.3 Examples

In the light of rather abstract, pseudoword-based evaluations some real examples sometimes help to reduce the abstractness of the presented results. Three words, *sheet*, *line* and *space* were chosen arbitrarily and some words representing the induced senses are listed below.

- sheet
  - beneath, blank, blanket, blotting, bottom, canvas, cardboard
  - accounts, amount, amounts, asset, assets, attributable, balance
- line
  - angle, argument, assembly, axis, bottom, boundary, cell, circle, column
  - lines, link, locomotive, locomotives, loop, metres, mouth, north, parallel
- space
  - astronaut, launch, launched, manned, mission, orbit, rocket, satellite
  - air, allocated, atmosphere, blank, breathing, buildings, ceiling, confined

These examples show that the found differentiations between senses of words indeed are intuitive. They also show that the found senses are only the most distinguishable ones and many futher senses are missing even though they do appear in the BNC, some of them even frequently. It seems that for finer grained distinctions the bag-of-words model is not appropriate, although it might prove to be sufficient for other applications such as Information Retrieval. Varying contextual representations might prove to be complementary to the approach presented here and enable the detection of syntactic differences or collocational usages of a word.

5 Conclusions

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

There are several questions that remain open. As the frequency of a word has a great impact on the possibility to disambiguate it correctly using the presented methods, the question is to what extent corpus size plays a role in this equation as compared to balancedness of the corpus and therefore the senses to be found. Another question is connected to the limitation of the presented algorithm which requires that any sense to be induced has to be representable by a rather large amount of words. The question then is, whether this (or any other similar) algorithm can be improved to discern 'small' senses from random noise. A combination with algorithms finding collocational usages of words probably offers a feasible solution.

The evaluation method employed can be used for automatic optimization of the algorithm's own parameters using genetic algorithms. Moreover, it would be interesting to employ genetic programming in order to let an optimal word sense induction algorithm design itself.

References

Michael Barth. 2004. Extraktion von Textelementen mittels "spreading activation" für indikative Textzusammenfassungen. Master's thesis, University of Leipzig.

Stefan Bordag. 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.

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

Beate Dorow and Dominic Widdows. 2003. Discovering corpus-specific word senses. In *Proceedings of EACL 2003*, pages 79–82, Budapest, Hungary.

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

Olivier Ferret. 2004. Discovering word senses from a network of lexical cooccurrences. In *Proceedings of Coling 2004*, pages 1326–1332, Geneva, Switzerland, August.

William Gale, 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.

Susan Gauch and Robert P. Futrelle. 1993. Experiments in automatic word class and word sense identification for information retrieval. In *Proceedings of 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 425–434.

Adam Kilgarriff. 1997. I don't believe in word senses. *Computers and the Humanities*, 31(2):91–113.

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

Daniel B. Neill. 2002. Fully automatic word sense induction by semantic clustering. Master's thesis, Cambridge University.

Patrick Pantel and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of ACM SIGKDD*, pages 613–619, Edmonton.

Amruta Purandare. 2004. Word sense discrimination by clustering similarity contexts. Master's thesis, Department of Computer Science, University of Minnesota, Duluth.

Reinhard Rapp. 2004. Mining text for word senses using independent component analysis. In *Proceedings of SIAM International Conference on Data Mining 2004*.

Reinhard Rapp. 2005. A practical solution to the problem of automatic part-of-speech induction from text. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 77–80, Ann Arbor, June. ACL.

Richard Rohwer and Dayne Freitag. 2004. Towards full automation of lexicon construction. In *Proceedings of HLT-NAACL 04: Computational Lexical Semantics Workshop*, Boston, MA.

Hinrich Schütze. 1992. Context space. In *Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, pages 113–120, Menlo Park, CA. AAAI Press.

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

Goldee Udani, Shachi Dave, Anthony Davis, and Tim Sibley. 2005. Noun sense induction using web search results. In *Proceedings of 28th ACM SIGIR*, pages 657–658, Salvador, Brazil.

Erik Velldal. 2005. A fuzzy clustering approach to word sense discrimination. In *Proceedings of the 7th International conference on Terminology and Knowledge Engineering*, Copenhagen, Denmark.

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