A hybrid WSD system using Word Space and Semantic Space

Haw-Bin Chai, Hwa-chow Hsu
Dept. of Computer Science
Swarthmore College
{chai, hsu}@cs.swarthmore.edu

Abstract

We describe a hybrid Word Sense Disambiguation (WSD) system using the context-based frameworks of both Word Space and Semantic Space. We develop confidence measures for the results generated by each model. To solve a WSD task, each classifier is run independently and the results combined using the confidence measures. The result is a more robust solution to the disambiguation task.

1. Introduction

Word Sense Disambiguation (WSD) remains a difficult problem in natural language processing. The source of the problem lies in the ambiguity of language – many words, in many languages, have different meanings in different contexts and situations. An often used example is the English word 'bank', which has the institutional sense (as in 'National Bank') and the shore sense ('river bank'), among others. Native speakers develop an intuitive ability to recognize these distinctions, but it is difficult to translate this ability into computational algorithms or models.

A variety of approaches have been attempted, ranging from statistics-based to connectionist methods. We focus on the Word Space and Structured Semantic Space approaches to WSD, two methods that develop the idea of modeling a language as a vector space. In the case of Word Space, a vector space is constructed for a particular word we wish to disambiguate, and spans all possible context words with which that word appears in a training corpus. Each instance of the ambiguous word is represented by a single point in this space. Structured Semantic Space is constructed similarly, but uses semantic information about context words rather than the words themselves, and models an entire language as opposed to a single ambiguous word.

Given the two methods' mutually independent information domains, we hypothesize that a hybrid system using both methods can take advantage of the strengths of each method while compensating for their weaknesses. One explicit way we hope to achieve this is by developing confidence measures for the results of each method and taking these into account when forming our final result.

The remainder of the paper is organized as follows: Section 2 explains the concepts of Word Space and Semantic Space in more detail. Section 3 describes our hybrid system and confidence measures for the results produced by Word Space and Semantic Space. Section 4 describes our implementation of this system. Section 5 presents our results, of which a discussion follows in Section 6. Finally, Section 7 describes possible future work.

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 46–54
Computer Science Department, Swarthmore College

2. Related Work

2.1. Word Space

A Word Space (Schütze, 1992) is an n-dimensional space of contexts of a particular word w, where n is the total number of unique words that co-occur with w in a training corpus, and each dimension of the space corresponds to one such unique word. Co-occurrence is determined by considering a window of a fixed number of characters before and after each instance of the word w in the training corpus. For example, a window of 1000 characters would include all words within 500 characters of an instance of w, both before and after.

A Word Space is built by taking every instance i of w in the training corpus. A vector representing i can be generated by considering all of the words in the context window of w along with their frequencies; the vector is non-zero in those dimensions which correspond to the words in the context window.

If the context of an ambiguous word is a good indicator of which sense it carries (this assumption is the basis for many WSD techniques), then the vectors associated with similar senses of w should have spatial locality in the Word Space for w. Vectors which are close to each other can be grouped into clusters, and the centroid of a cluster (the average of all vectors in the group) can be thought of as the "sense" for the cluster. Therefore, a small number of centroid vectors representing senses of w also exist in the Word Space of w.

WSD is accomplished by comparing the vector representing an instance of w in the test set to each of the centroid vectors determined through clustering, and assigning it the sense of the nearest vector, using cosine of the angle between the vectors as a metric. However, unless we assign a dictionary definition to each centroid vector as well, that w has been determined to carry the sense of a certain cluster means very little; we don't know what the cluster itself means! Schütze allowed assigning of real-world definitions to clusters by hand in his work with Word Space. Sense-tagged corpora can be used to automate this process of assigning definitions.

2.2. Structured Semantic Space

The Structured Semantic Space approach to WSD (Ji and Huang, 1997) is reminiscent of Word Space insofar as it involves creating context vectors and clustering them according to a similarity metric within an n-dimensional space. In the "sense space", however, similarity is measured with respect to the semantic categories of the context words rather than the context words themselves. Each of the n dimensions of the sense space corresponds to a semantic category, as defined in a dictionary resource such as Roget's Thesaurus. Additionally, a corpus tagged with semantic senses is required to construct the context vectors of monosense words, which outline the sense clusters in the space. The relevance of each particular semantic category c to the sense of the monosense word w is captured by a *salience* value, given by the formula:

$$Sal(c, w) = \frac{|\{w_i \mid c \in NC_i\}|}{k}$$

where NC_i is the set of all semantic codes for neighboring words of instance i of word w, and k is the total number of occurrences of w. Each unique w that appears in the corpus is therefore represented by a context vector of length equal to the number of semantic categories, where the c'th element of the vector is equal to the salience of semantic category c with respect to w:

$cv_w = \langle Sal(c_1,w), Sal(c_2,w),\ldots,Sal(c_k,w)\rangle$.

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 46–54
Computer Science Department, Swarthmore College

The similarity (distance) metric between two context vectors is defined as (1 - cos (cv1, cv2)), where cos(cv1, cv2) is the cosine of the angle between the two vectors. A tree-based algorithm that iteratively merges the most similar context vectors together is then used to partition the set of context vectors into a number of sense clusters. A sense cluster is characterized by its centroid.

Actual disambiguation of a word takes place in two steps. First, the words within the context window of an instance of an ambiguous word are used to create a context vector consisting only of 1's and 0's. This vector is compared to all sense clusters in the space using the distance metric described above, and all clusters within a certain threshold distance are "activated", or selected as candidates for the next step. Second, a context vector is created for each dictionary sense of the ambiguous word, based on the contents of a collocation dictionary. The distance between each dictionary sense vector and each activated cluster is calculated, and the sense minimizing the distance (maximizing similarity) is selected for the ambiguous word.

3. A Hybrid Approach

We describe a hybrid system combining features of the above mentioned methods with the following two goals in mind: 1) automatic assignment of real-world senses to sense clusters in Word Space, and 2) increased performance by employing Word Space and Semantic Space in parallel and combining the results of both methods, taking into account available confidence measures.

3.1. Automatic Tagging of Word Space Clusters

We present a method for implementing unsupervised tagging of Word Space clusters. The method requires a sense-tagged corpus, and simply involves assigning the sense with the highest representation in a cluster to that cluster. For any ambiguous word, the more consistent the mapping between senses and clusters in the Word Space, the more confidence we have in the disambiguation result. If the sense-tagged corpus is small, the clusters can first be generated from a larger, untagged corpus. The cosine similarity between context vectors for instances of each sense of the ambiguous word in the sense-tagged corpus and the cluster centroids is then computed, and the vectors are assigned to their closest clusters. We tally the number of times each particular sense was assigned to each cluster, and expect the tally to be high for only one sense per cluster, indicating that the cluster is representative of that sense.

We define a *representativeness* measurement for each sense s of ambiguous word w in cluster c, given by

$$R(c,s) = \frac{s_c / n_c}{s_t / n_t}$$

where s_c is the number of occurrences of sense s in cluster c, n_c is the total number of sense occurrences in c, s_t is the total number of occurrences of sense s in the corpus, and n_t is the total number of occurrences of w in the corpus. The numerator describes the ratio of sense s in cluster c, while the denominator normalizes according to the number of times s appears in the corpus. For word w_0, a representativeness value of 1 for sense s_0 indicates that the distribution of s_0 with respect to all senses of w_0 in cluster c_0 is the same as the distribution of s_0 with respect to all senses of w_0 in the entire corpus. Given that vectors are clustered by similar contexts, we assume that the more similar a cluster's sense distribution is to the sense distribution of the corpus, the less "unique" the context represented by the cluster is to its senses.

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 46–54
Computer Science Department, Swarthmore College

Under this assumption, cluster c_0 provides no information for disambiguating sense s_0. Thus, representativeness estimates how reliable disambiguation based on a particular cluster will be. (For convenience's sake, we take the natural log of representativeness values in our system, shifting the value of neutral representation from 1 to 0. Positive representativeness will always mean a sense is well represented in a cluster, and negative representativeness will always mean the opposite. The characteristic representativeness of a cluster is its largest positive representativeness value. We test the utility of this measurement in our experiments.)

3.2. Improving Performance by Employing Word Space and Semantic Space in Parallel

The orthogonality of different WSD techniques suggests that using multiple methods will improve overall performance. Our approach is to apply both Word Space- and Semantic Space-based disambiguation in every disambiguation event. As Ji and Huang point out, a confidence rating is implicit in the Semantic Space distance calculated between word senses and activated clusters; if this distance is large, it is very unlikely that the system will select the correct sense.

Our intuition is that an analogous confidence rating is implicit in Word Space distance calculations as well. If the distance between a word's context vector and its potential sense clusters is large, or if the sense clusters are all more or less equidistant from the context vector such that no single sense is strongly preferred over the others, we should put less faith in the system's determination.

Intelligent consideration of these confidence measures in conjunction with the results of both disambiguation methods should allow the hybrid system to show improvement over each individual system.

4. Implementation

Our implementation of Word Space involves the following steps: First, we parse a corpus, searching for occurrences of words in a list of target ambiguous words. We build context vectors for each occurrence. Second, we reduce the dimensionality of the context vectors to 100 by means of Singular Value Decomposition in order to facilitate clustering by AutoClass, a program for clustering data by modeling the data as mixture of conditionally independent classes. (For comparison, the average unreduced context vector length in our experiments was 18145.) Next, we run AutoClass to generate clusters from the vectors of reduced dimensionality. The results tell us which vectors belong to which clusters; we use this information to compute the centroids of the clusters in the original space. Finally, to perform WSD on an instance of an ambiguous word we construct its context vector, find the cluster with the highest cosine similarity to it, and assign the most representative sense of that cluster to the word.

In order to test the performance of the Word Space classifier in the absence of a sense-tagged corpus, we use *pseudowords* (Schütze, 1992). A pseudoword is a set of two or more monosense words having different senses which we consider to be a single word with multiple senses. For testing purposes, pseudowords may then substitute for a sense-tagged corpus: for example, we can pick two words, 'house' and 'dog' and treat them as one word with two senses, a 'house' sense and a 'dog' sense.

To evaluate the Word Space component, we ran four different experiments, with (CITY, HOME), (FACE, WAR), (CHILDREN, HAND), and (EYES, SYSTEM) as our pseudowords. We selected only nouns under the assumption that they possess the most distinct context vectors. Our context window of choice was 1000 characters wide,

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 46–54
Computer Science Department, Swarthmore College

which Schütze found to be ideal in his experiments. We trained on the Brown Corpus and tested on a 1 million word WSJ corpus. Since word frequencies between the two corpora are significantly different, we also test using the Brown corpus, with the idea that it can hint at the "best case" performance potential of our system. The distribution of the pseudowords in the corpora is given in Table 1:

Pseudoword	Brown corpus	WSJ corpus
CITY/HOME	415 / 547	316 / 795
FACE/WAR	314 / 310	167 / 167
CHILDREN/HAND	372 / 419	220 / 103
EYES/SYSTEM	394 / 404	36 / 479

Table 1. Frequencies of pseudowords in corpora

We test the usefulness of the representativeness measure (section 3.1) by disregarding clusters in order of increasing representativeness value and noting the effect on precision. Finally, we look for a correlation between correctness of disambiguation and the distance from the ambiguous words' context vectors to the closest cluster centroids.

We implement Semantic Space in the following manner: First, we parse the public domain 1911 version of Roget's Thesaurus and create a database of semantic categories. Second, we map word-POS pairs that appear only under a single semantic category to the headers under which they appear, discarding all word-POS pairs which appear in multiple categories. These are the monosense words we are able to identify in a POS-tagged corpus. We then semantically-tag the Brown Corpus.
To build the Semantic Space, we follow the same procedure as described in section 2.2, with the exception that we choose to try AutoClass as our clustering method instead of the trie method described by Ji and Huang. To test disambiguation, we construct a pseudoword by selecting two or more words with distinct senses from the tagged, unambiguous words. Next, we can generate the equivalent of a collocation from the context of the selected words. Because this collocation is generated from a corpus which may not be representative of all contexts in which the words may appear, it may not be as general as a collocation taken from a collocation dictionary. However, we hope it adequately reflects the semantic nature of most contexts. If the content of the test corpus is of the same genre as that of the training corpus, we expect this to be the case. A larger and more representative training corpus may obviate this problem.

To simulate disambiguation of the pseudoword in a test corpus, we follow the same procedure as described in section 2.2. We search for occurrences of the pseudoword, using the semantic-category thesaurus to tag context words, and from these build either normalized Boolean or frequency vectors. We activate nearby sense clusters in Semantic Space with this context vector, and determine which word in the pseudoword set is closest to one of the activated clusters, according to the collocation for each such word found earlier.

Unfortunately, due to time and computer hardware limitations, we have to date been unable to obtain useful data from the Semantic Space component of our system.

5. Results

Tables 2, 3, 4, and 5 summarize the results of our experiments with regard to the representativeness measure. The first row shows the precision of the system taking into account all clusters; each successive row drops the cluster with the lowest representativeness value until only one cluster is left. The second column shows the results using the WSJ corpus; the last column shows the result using the Brown corpus.

The recall value for all experiments is 100%, because our system in its current form returns an answer for every instance of the

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 46–54
Computer Science Department, Swarthmore College

# clusters dropped	WSJ Corpus	Brown Corpus
0	.455856	.587929
1	.455856	.551509
2	.35045	.547347
3	.35045	.569199
4	.410811	.578564
5	.384685	.573361
6	.317117	.562955
7	.501802	.57232
8	.547748	.619147
9	.363063	.526535
10	.361261	.539022
11	.284685	.430801
12	.284685	.430801

Table 2. Precision results for Test #1

# clusters dropped	WSJ Corpus	Brown Corpus
0	.462462	.659905
1	.477477	.677804
2	.477477	.71957
3	.468468	.720764
4	.459459	.626492
5	.435435	.610979
6	.45045	.613365
7	.465465	.713604
8	.513514	.554893
9	.498498	.557279

Table 3. Precision results for Test #2

# clusters dropped	WSJ Corpus	Brown Corpus
0	.313665	.635572
1	.31677	.631841
2	.319876	.646766
3	.60559	.656716
4	.590062	.661692
5	.590062	.655473
6	.639752	.619403
7	.649068	.61194
8	.680124	.468905
9	.680124	.468905
10	.680124	.468905

Table 4. Precision results for Test #3

# clusters dropped	WSJ Corpus	Brown Corpus
0	.680934	.818293
1	.363813	.858537
2	.363813	.859756
3	.344358	.858537
4	.392996	.868293
5	.441634	.871951
6	.929961	.510976
7	.929961	.510976
8	.929961	.510976
9	.929961	.510976
10	.929961	.510976
11	.929961	.510976
12	.929961	.510976

Table 5. Precision results for Test #4

ambiguous word – no thresholding or other means of filtering results are currently employed.

Figure 1 shows the results with respect to the distance values for the experiments using the WSJ corpus in graphical form, while Figure 2 shows the results for the same experiments using the Brown corpus. Note that the values on the vertical scale are cosine similarity values; thus, a low cosine similarity value indicates a large distance.

6. Discussion

Generally, our results do not match the reported performance in Schütze’s paper. We believe that this may be due to training data sparseness. Another reason for the low performance on the WSJ tests is the fact that we are testing on a different corpus than the one we are training on; the Brown and WSJ corpora might have sufficiently different types

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 46–54
Computer Science Department, Swarthmore College

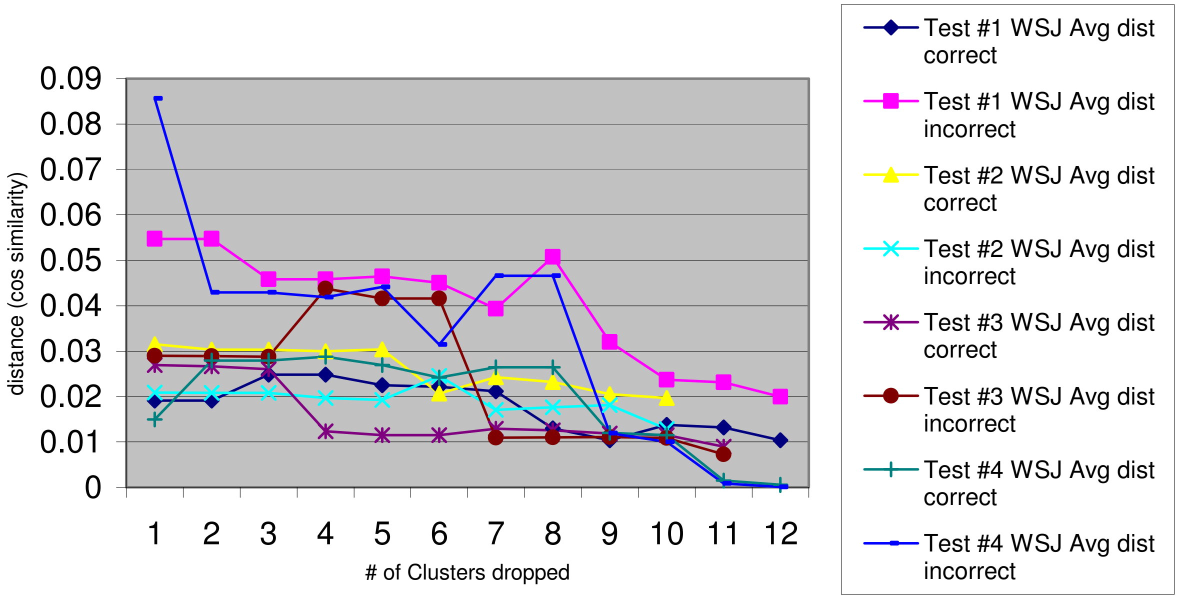

Figure 1. Distances for correct and incorrect classifications (WSJ corpus)

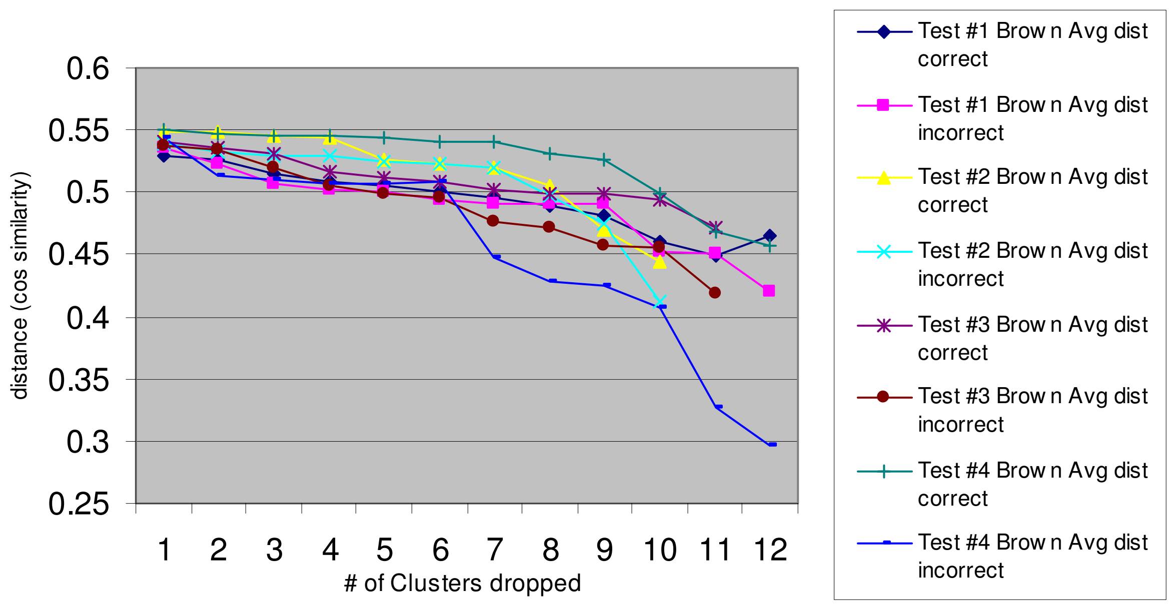

Figure 2. Distances for correct and incorrect classifications (Brown corpus)

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 46–54
Computer Science Department, Swarthmore College

of information that the context vectors are too dissimilar to produce good results. Nevertheless, despite the overall low performance, we wish to discuss several trends that we observed.

6.1. Representativeness

An interesting trend we observed is that in all four tests and with both testing corpora (with the exception of test #4 using the WSJ corpus; we provide an explanation of this anomaly later), the precision of our system is never at its peak with all clusters used. Instead, as we drop the first several clusters, a general trend of increasing precision sets in, leading up to the peak performance. A possible explanation is that because the dropped clusters have low representativeness values, they contribute little to word sense disambiguation. In fact, allowing these clusters to remain in the system impairs performance by "attracting" context vectors in their vicinities that otherwise would be assigned to sense clusters with higher representativeness values. As we drop even more clusters, we begin to lose important clusters and the system performance degrades.

Note that towards the end of each column, the precision values have a tendency to remain constant. This indicates that all remaining clusters lean towards the same sense; all instances of the ambiguous word are automatically assigned that sense, and the precision value obtained is identical to the ratio of that sense in the testing corpus. In the case of test #4, this leads to an absurdly high precision value of 93% when using the WSJ corpus for testing. Of course, no attention should be paid to these values.

As mentioned earlier, Test #4 using the WSJ corpus performs best when all clusters are considered. However, during the clustering phase of the Word Space, the most populated cluster turned out to be representative of the SYSTEM pseudosense. At the same time, this cluster's representativeness value was the lowest. We also recall that the WSJ corpus has a SYSTEM/EYES ratio of 479/36. Thus, the high initial precision can be attributed to the fact that the SYSTEM cluster described above very likely attracted many context vectors during the testing phase (since it attracted the most context vectors during the training phase), and since there were many more SYSTEM instances than EYES instances in the testing corpus, these taggings turned out to be correct. Once we dropped that cluster, some of these context vectors were assigned incorrectly to EYES clusters, thus lowering the performance.

Another exception to this trend is found in Test #3 using the WSJ corpus, which fails to exhibit the behavior of dropping precision as more clusters are dropped. This can be explained by two facts: that the highest representativeness clusters were all of the CHILDREN pseudosense, and. that CHILDREN appeared twice as many times as HAND in the test corpus.

Another interesting trend is that there seems to be some correlation between the points of highest precision when using the Brown corpus and when using the WSJ corpus. This suggests that the training corpus used to generate a Word Space can also be used to find the optimum cutoff point for dropping clusters and thus optimize actual disambiguation.

6.2. Distance value as confidence measurement

Figures 1 and 2 show that there is very little consistency in the cosine similarity values between correctly and incorrectly classified instances. The average cosine similarity of the correctly classified instances was greater than incorrectly classified instances in some cases (for example, test #2 in Figure 1), whereas in the other cases, surprisingly, the opposite was true (test #1 in Figure 1). In the Brown corpus results, the values were generally too close

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 46–54
Computer Science Department, Swarthmore College

together for any distinction between correct and incorrect classifications to be reasonable. We conclude that distance to the closest cluster is not a good confidence measure for results obtained from Word Space.

7. Future Work

Future work would of course entail completing our proposed hybrid system, followed by implementing a voting system between the Word Space and Semantic Space components of the system. Although we found the distance to the closest cluster in Word Space to be an unreliable confidence measure, perhaps the representativeness measure can in some way be used instead.

Performing additional experiments using new pseudowords would allow us test the validity of our interpretation of the relationship between the optimal cutoff point for dropping clusters in the training corpus and testing corpus.

Another way of using the representativeness measure could be to perform screening on disambiguation results. Instead of dropping clusters, we could set some minimum representativeness threshold. For disambiguation attempts that do not break that threshold, we do not return an answer, thus lowering the recall rate of the system. But since clusters with low representativeness values in general do not disambiguate well, we expect the precision to increase as a result of the thresholding.

We would also like to experiment on different languages with the hybrid system. Ji and Huang claim that the Semantic Space approach is language independent; we expect Word Space to be as well. We currently have the resources to perform tests in Chinese.

We have also discovered that the AutoClass clustering package generates some weight measures for each class found during the clustering process. This information can possibly be used to supplement existing confidence measures.

8. References

Hasan, Md Maruf; Lua, Kim Teng. Neural Networks in Chinese Lexical Classification. http://citeseer.nj.nec.com/8440.html

Ji, Donghong; Huang, Changning. 1997. Word Sense Disambiguation based on Structured Semantic Space. *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics.

Schütze, Hinrich. 1992. Dimensions of Meaning. In *Proceedings of Supercomputing '92, Minneapolis,* pages 787 – 796.

Yarowsky, David. 1992. Word-sense Disambiguation using Statistical Models of Roget's Categories Trained on Large Corpora. In *Proceedings of COLING '92,* pages 454 – 460.

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 46–54
Computer Science Department, Swarthmore College