Information Retrieval Based on Word Senses

Hinrich Schütze and Jan O. Pedersen
Xerox Palo Alto Research Center
3333 Coyote Hill
Palo Alto, CA 94304
internet: {schuetze,pedersen}@parc.xerox.com
URL: ftp://parcftp.xerox.com/pub/qca/SenseIR.DAIR95.ps

Abstract

This paper proposes an algorithm for word sense disambiguation based on a vector representation of word similarity derived from lexical co-occurrence. It differs from standard approaches by allowing for as fine grained distinctions as is warranted by the information at hand, rather than supposing a fixed number of senses per word, and by allowing for more than one sense to be assigned to a given word occurrence.

The algorithm is applied to the standard vector-space information retrieval model and an evaluation is performed over the Category B TREC-1 corpus (WSJ subcollection). Results show that this sense disambiguation algorithm improves performance by between 7% and 14% on average.

1 Introduction

An ambiguous term is a word with multiple senses, where a *sense* is a group of similar usages of a word dissimilar from other usages. Ambiguity varies in degree from homographs (i.e., two different words that happen to be spelled identically, such as "bank" in the sense of financial institution as opposed to the geographic feature) to the more subtle graded sense distinctions of words such as "space" (line space vs. office space vs. exhibition space). Out of context it is impossible to determine which sense is intended. However, in context the sense is determined by various cues, such as syntactic role, nearby words, and semantics. For example, "river bank" is clearly distinguished from "Bank of New York".

Automatic word sense disambiguation is a central problem in the computational treatment of language. However, the role of sense disambiguation in IR is unclear. In the standard "bag of words" analysis of text, each word occurrence is treated as a separate isolated feature; word order, and, hence, local context is not preserved. One would think therefore that ambiguity would be a problem. However, experiments have not shown appreciable improvements in overall system performance through the application of word sense disambiguation algorithms [25]. This is partly due to the fact that retrievals typically do not depend on a single term, but rather on a set of related terms expressed in the query. In order for a document to score highly it must match several of these terms simultaneously; often, this is enough to achieve disambiguation. However, it seems intuitively clear that a good sense disambiguation algorithm should, at least, not decrease performance and might potentially increase it in some cases (especially for short queries). Recent work [34] casts doubt on this intuition.

We will comment on that work and other related work, describe an automatic word sense disambiguation algorithm based on a vector representation of word similarity derived from lexical co-occurrence, apply it to information retrieval, and present experimental results that show substantial improvement over a baseline system.

2 Related Work

Different sources of information have been employed to discriminate senses computationally: Kelly and Stone [22] consider hand-constructed disambiguation rules, Lesk [27], Krovetz and Croft [24], and Guthrie et al. [15] use online dictionaries, Hirst [20] constructs knowledge bases, Cottrell [4] uses syntactic and semantic structure encoded in a connectionist net, Brown et al. [1] and Church and Gale [3] exploit bilingual corpora, Dagan et al. [7] use a bilingual dictionary, Hearst [18] and Leacock et al. [26] exploit a hand-labeled training set, and Yarowsky [41] performs a computation based on Roget's thesaurus. McRoy [28] investigates how multiple knowledge sources can be combined for disambiguation. Considerable effort is necessary to construct or obtain some of these information sources, for example hand-constructed disambiguation rules and hand-labeled training sets. All these approaches share the problem of coverage: specialized domains tend to exhibit rare words and specialized meanings, which are not covered by generic lexical resources. The cost of customizing the resources is often prohibitively high.

If a limited degradation in disambiguation performance is acceptable, then a solution to the problem of coverage is to derive the knowledge needed for disambiguation directly from the text collection of interest. This paper presents an algorithm for learning this knowledge from text by inducing a *thesaurus.* Each word is represented by a vector formed from counts of its near neighbors in the text corpus. To the extent that related meanings are expressed by similar words, two semantically related words have similar neighbors. Hence, their vectors will have similar entries. This similarity can be measured in the usual way by the cosine between vectors. An occurrence of a word can then be classified as belonging to one sense or another by considering the co-occurrence patterns of neighboring words as represented by their vectors.

The work presented here differs in another respect from previous approaches. In general, two assumptions have been made:

- Only one sense of a word is used in each occurrence.
- Each word has a fixed number of senses.

The first assumption has been recently challenged by Kilgarriff [23]: "Sometimes two senses of a word are mutually exclusive, but more often they are not ...". He finds that a large proportion of the words he investigates is used with several senses in one context. We will test both hypotheses in our information retrieval experiment and find that the many-sense approach improves performance.

Kilgarriff also expresses reservations about the second claim, that words have a determined number of senses: "Often, the senses as identified in the dictionary identify points on a continuum of possibilities for how the word is used and dictionary senses might equally have been written which divided up the space differently." In the same vein, Geeraerts [13] argues convincingly that the criteria for distinguishing between vagueness and ambiguity are not consistent, and that this distinction is vague itself. In lexicography, difficulties with sense individuation are well known. For example, Morton quotes Philip Gove, the editor of Webster's third, as saying [30]:

> Rather grotesquely, after centuries of lexicography and language study of one sort or another, it appears that no one has answered the question of how we may know with sharp clarity and definitive exactness when a word has one meaning alone ... and when it has two or more quite discrete meanings.

Our working hypothesis will therefore be that the individuation of senses is to a large extent arbitrary and distinctions of any grain can be justified if enough information is available. Therefore, we will make fine distinctions wherever there is sufficient information to do so.

Various attempts have been made to apply automatic sense disambiguation to information retrieval with indifferent results. In the following we analyze in greater detail two representative experiments [39, 34] and suggest possible reasons why they are less successful than the method presented in this paper.

2.1 Using WordNet

Voorhees [39] uses the WordNet [29] online thesaurus to perform disambiguation. Roughly, WordNet is first transformed into a mapping from words to one or more classes (called "hoods"). If many words from a particular class c occur with an ambiguous word w, then w is disambiguated as belonging to the sense that is associated with c.

This algorithm is less robust than the one presented here since in some cases none of the possible senses can be chosen. For example, if a word has two senses, but its neighbors in the context in question are not members of either class, then no disambiguation decision can be made. In contrast, because we disambiguate based on a continuous measure of word similarity defined for *all* words in the corpus of interest, sufficient data exists to make a disambiguation decision for every occurrence.

However, we believe the main difficulty with using WordNet directly is its lack of coverage. The ontology that organizes the hierarchy (or heterarchy), the senses that are defined for individual words, and the coverage of the vocabulary of English are all chosen for maximal generality (which for other applications is a strength). For example, proper names (such as "Steffi Graf") are often excellent disambiguation clues ("Graf" disambiguates "court" as "a quadrangular space for playing tennis"), but proper names are not covered in WordNet. Again, we address this difficulty by deriving a thesaurus directly from the corpus of interest ensuring that all words will be considered as indicators for or against the presence of a particular sense.

The sense partition chosen for a particular word can be a problem if the text collection invokes additional senses that are not covered in WordNet. For example, "dl" is only listed as "deciliter", "derby" only as "bowler hat". The meanings "disabled list" and "horse race" that may come up in a newspaper collection are missing. Any specialized text will have senses that are not defined in a general lexicon.

Finally, the ontology that is pre-defined in WordNet is not tuned to distinctions that may be crucial in a specialized text collection. For example, the distinction between number theory and discrete algebra may be important in a collection of mathematics texts, but it is not defined in WordNet. WordNet-based disambiguation seems to be successful when the ontology reflects the relevant domain correctly (for example, "distribution" in statistics vs. computer science, a distinction that is reflected in WordNet's ontology). On the other hand, to the extent that they are characterized by different co-occurrence patterns the sense partitioning scheme presented in this paper can discriminate between subtopics in specific and general text collections equally well.

We are not arguing that WordNet should not be used for disambiguation. On the contrary, it is a valuable source of information that may well improve the results presented here if the problems outlined above are overcome. For example, Hearst and Schütze [19] consider a possible way to address the lack of specificity.

2.2 Testing disambiguation with pseudowords

Sanderson [34] uses pseudowords [36, 42] to test the utility of disambiguation for information retrieval. A pseudoword is created by assigning two or more types, for example "banana" and "door", to a new type, "banana-door". In the subsequent evaluation, the retrieval system has no access to information that would distinguish the two words. One would expect that if two words are conflated that contribute important information to a query, performance would decrease. Sanderson describes several experiments in which this is unexpectedly not the case: performance decreases only marginally or not at all with pseudo-word creation.

However, several factors in the experimental setup suggest that the model created by Sanderson is not representative of typical adhoc information retrieval settings. First, an optimized query representation is computed via feedback. The n ($1 \leq n \leq 30$) terms that rank highest in the feedback process form the query for the experiments. This means that the n best discriminators for the particular query make up the query. This is not typical of natural queries which often contain few good discriminators. Degrading the discriminatory potential of query terms may be more harmful in an experiment with few good discriminators than one with many.

Secondly, the queries used in the experiment are broad topic codes, rather than precise statements of information need. It seems plausible that degrading the query has little effect if the information need is of the unfocussed nature that is typical of topic codes like "metals" or "grain".

	frequency range			total number	percentage other senses ≤ 1%	≤ 5%	≤ 10%	≤ 20%	≤ 50%
tokens	(all)			15762291	55	76	82	88	94
types	1	–	24	82947	0	0	0	0	5
	25	–	29	1474	0	0	0	10	40
	30	–	38	1908	0	0	0	13	45
	39	–	50	1648	0	0	3	20	52
	51	–	69	1756	0	0	7	28	60
	70	–	98	1698	0	2	15	35	64
	99	–	146	1736	0	8	24	45	69
	147	–	245	1718	0	17	34	52	75
	246	–	473	1699	1	28	47	62	81
	474	–	1394	1710	12	50	64	75	89
	1395	–	80000	1706	49	76	85	91	97

Table 1: In random pseudoword generation, few low-frequency word types, but many high-frequency word types are the majority sense of a pseudoword.

A final point concerns the type of ambiguity that is modelled in Sanderson's work. Pseudowords in the experiment are created by selecting five word types at random and conflating them. We will argue that this technique tends to generate ambiguous word types that don't have an adverse effect on retrieval performance even though other types of ambiguity can result in serious performance degradation.

Table 1 reports an experiment in which 20,000 pseudowords with 5 senses each were generated for the Tipster Wall Street Journal subcollection [16]. Statistics were gathered for all tokens involved and for 11 word type frequency ranges: word types with fewer than 25 occurrences, and 10 more frequency ranges, each with roughly the same number of word types. The table shows for each range what percentage of word types were the majority sense of a pseudoword whose other senses contribute less than 1%, less than 5%, less than 10%, less than 20%, and less than 50% of the tokens for that pseudoword. For example, word types in the first column "≤ 1%" are the majority sense of their pseudoword and account for more than 99% of its tokens with the four other senses making up less than 1% of the tokens. The row labelled "tokens" shows what percentage of all tokens are instances of the majority sense of a pseudoword in the category indicated by the column head ("≤ 1%" etc.). It is apparent from the table that the discriminatory power of low-frequency word types is obliterated by pseudoword creation. Only 5% of the least frequent word types (frequency range 1 – 24) are the majority sense of a pseudoword whose tokens account for more than 50% of the tokens of that pseudoword. However, conflation does little damage to medium- and high-frequency word types. For example, 49% of the most frequent word types are hardly affected at all: less than 1% of their tokens are contaminations from other senses. In fact, more than half of all tokens in the corpus (55%) belong to the majority sense of a pseudoword with less than 1% contamination from other senses. From this perspective, it is not surprising that overall retrieval performance increases only marginally when this type of ambiguity is resolved.

There is some evidence that many ambiguous words behave like pseudowords (i.e., have a majority sense) if only coarse sense distinctions are made. For example, looking at a coarse-grained notion of ambiguity, Gale et al. [10] find that for a random sample of words choosing the most frequent sense results in a disambiguation performance of 92% (averaged over types). This result could be interpreted as showing that most ambiguous words are similar to randomly created pseudowords in that they have one clearly dominating sense (so that picking that sense in all cases does a surprisingly good job at disambiguation). However, Gale et al.'s experiment has not been repeated for finer sense distinctions. For example, "capacity", "marine", and "stretch" are categorized as unambiguous in their experiment, but a large dictionary would identify several senses for each of these words. It therefore remains unclear what proportion of ambiguous words have a majority sense according to a more fine-grained notion of ambiguity.

In summary, we take Sanderson's work as indicating that the successful application of disambiguation to information retrieval is problematic in the following circumstances:

- The queries are topic codes, rather than more specific statements of information need.
- The queries contain many terms that are optimal discriminators between relevant and non-relevant documents.
- The words that are disambiguated are either low-frequency words or medium and high frequency words with a majority sense that is much more frequent than the other senses.

The experimental setting chosen in this paper is more typical of the standard ad-hoc information retrieval environment in that natural queries (topics from the Tipster collection) are used. In addition, only medium and high frequency words will be disambiguated and we will bias our disambiguation algorithm towards discovering senses with a similar number of tokens.[1]

3 Thesaurus construction

Our sense disambiguation algorithm is based on a notion of word similarity computed from lexical co-occurrence statistics gathered directly from the corpus of interest. We refer to this similarity measure as a thesaurus since it relates words to other similar words, but it should not be confused with hand-built lexical resources such as WordNet.

The goal of the lexical co-occurrence based thesaurus is to associate with each term a vector that represents its pattern of local co-occurrences. This vector can then be compared with others to measure the co-occurrence similarity, and hence semantic similarity of terms.

The starting point of the computation is to collect a (symmetric) term-by-term matrix C, such that element c_{ij} records the number of times that words i and j co-occur in a symmetric window of total size k that is centered on word i (k is forty-one words in our experiments). Topical or semantic similarity between two words can then be defined as the cosine between the corresponding columns of C. The assumption is that words with similar meanings will occur with similar neighbors if enough text material is available. Qiu and Frei [31] use a similar scheme, although the matrix in their case is documents vs. terms.

However, simple resource calculations suggest that this direct approach is not workable. The matrix C has $v^2/2$ distinct entries where v is the size of the vocabulary. Although this matrix is sparse, we can expect v to be very large, and hence the overall storage requirement to be unworkable. For example, the category B TREC-1 corpus[2] [17], which is the subject of our experiments, has over 450,000 unique terms.

Even if enough memory were found to represent C directly, the thesaurus vectors associated with each word (columns of C) would be v-dimensional. Although these vectors are somewhat sparse, this implies that word comparisons are an order v operation, which is prohibitively expensive for large scale applications.

We address these issues by reducing the dimensionality of the problem to a workable size. The key dimensionality reduction tool is a singular value decomposition [14] of the matrix of co-occurrence counts (see [8] for a different use of SVD in information retrieval). However, this matrix must be constructed in a series of steps to keep the computations tractable at each stage. A more detailed discussion of this computation can be found in [38]. While other automatic thesaurus construction algorithms exploit similar information [5, 9, 21], they use it

[1] Mark Sanderson recently applied pseudoword creation to two collections with natural queries (CACM and Cranfield) and found effects "broadly similar" to those of the Reuters experiment [35]. If no consistent differences between natural and artificial queries are found, one would have to conclude that the different biases of the two disambiguation methods (for vs. against strongly dominating senses) are responsible for their different impact on IR performance.

[2] A standard ARPA text corpus of roughly 170,000 documents and 500MB of text from the Wall Street Journal and a set of 25 evaluation queries (Topics 51-75).

accident	repair faulty personnel accidents exhaust equipped MISHAPS injuries sites
advocates	passage PROPONENTS argument address favoring compromise congress favors urge
litigation	LAWSUITS audit lawsuit file auditors auditor suit sued proceedings
tax	taxes income;tax new;tax income;taxes taxpayers incentives LEVIES taxpayer corporate;taxes
treatment	drugs syndrome administered administer study administering PROCEDURE undergo aids

Table 2: Synonyms are often nearest neighbors of each other in the SVD-based thesaurus.

for vector expansion instead of word sense disambiguation.

3.1 Sample Synonyms

The net effect of this computation is to produce for each unique term a dense p-dimensional vector that characterizes its co-occurrence neighborhoods. These thesaurus vectors then define a thesaurus by associating each word with its nearest neighbors. Table 2 displays some of the associations found in our experiment over the category B TREC-1 corpus. Each row displays a word and its nine nearest neighbors. For example, "repair" is the nearest neighbor of "accident". Word pairs used as terms are displayed as couples separated by semicolon. Words in upper case are hand selected synonyms as might be found in a manually constructed thesaurus. They are particularly interesting because they are unlikely to co-occur with their mates and hence illustrate that this thesaurus construction effectively uses second-order co-occurrence (sharing neighbors in the corpus) rather than simple first-order co-occurrence (occurring near to each other) to find synonyms.

4 The Disambiguation Algorithm

An individual occurrence t of a word can be characterized by summing the thesaurus vectors of the words that occur close to it to produce a *context vector* [40, 12]. Since the direction of a word's vector corresponds to its main topic, if several words with the same main topic occur close to t, then that topic will dominate in the computation of the context vector. On the other hand, topics that are represented by only one word in the environment of t will not influence the direction of the context vector significantly. Since not all words are equally useful for defining topic, the computation of context vectors weights each word vector according to its discriminating potential as measured by inverse document frequency [32]:

$$w_i = log(\frac{N}{n_i})$$

where N is the total number of documents in a text collection (or any other meaningful unit, e.g. paragraphs) and n_i is the number of documents that the word w_i occurs in. All content words in a symmetric 41-word neighborhood centered on t are summed to compute the context vector.

The context vector characterizes the local topic of an individual word occurrence. The remaining problem is to find directions in the space that correspond to different senses of a word. The approach taken here is to consider the context vectors corresponding to all the occurrences of a particular word in a training set, partition them (through a clustering algorithm) into regions of high density, and posit a sense for each such region. For the partitioning we typically employ Buckshot, an efficient approximation of group agglomerative clustering [6][3]. A vector is included in a region if it is closer to that region's cluster centroid than to any other region's cluster centroid. Word occurrences can be disambiguated by computing the context vector of each occurrence, finding the closest centroid and assigning the occurrence the sense of that centroid.

In summary, the disambiguation algorithm proposed here has three phases. First, a number of context vectors of a target word are computed from a training set. The context vectors are clustered, with each cluster ideally correspond-

[3] AutoClass [2] is used for some of the words in section "Case Studies".

word	senses	% rare senses	# clusters	% correct 1	2	3	total
capital/s	goods/seat of government	5	2	96	92		95
interest/s	special attention/financial	15	3	94	92		93
motion/s	movement/proposal	0	2	92	91		92
plant/s	factory/living being	4	13	94	88		92
ruling	decision/to exert control	14	2	90	91		90
space	area, volume/outer space	0	10	89	90		90
suit/s	legal action/garments	18	2	94	95		95
tank/s	combat vehicle/receptacle	15	8	97	85		95
train/s	railroad cars/to teach	2	10	94	69		89
vessel/s	ship/blood vessel/hollow utensil	10	7	93	91	86	92

Table 3: Accuracy for the disambiguation algorithm averages about 90%.

ing to one sense of the word. Finally, disambiguation consists in assigning the occurrence that is to be disambiguated to the sense of its closest cluster.

5 Case Studies

Before considering a large scale systemic information retrieval evaluation, to gain insight into the disambiguation algorithm we consider in this section some case studies of its detailed behavior over selected words[4]. For each word the number of classes was determined from a clustering algorithm; for Autoclass this number was determined fully automatically, for Buckshot clustering into 2, 5, 7 and 10 classes was performed and the satisfactory result with the lowest number of classes was selected. The clusters were inspected and labeled as belonging to one sense or another, thus conflating some clusters. We will only consider the most frequent senses here. Rare senses such as the "tank top" sense of "tank" are not included. A test set that was not used in training was then processed by assigning the context vector of each word to its closest centroid. Table 3 summarizes the results. On average, the performance of the algorithm is about 90%. For each word, the table lists the senses, the number of clusters used, the percentage of tokens not covered by a common sense, and disambiguation results for the common senses as well as for the word as a whole.

The following two contexts of "suit" were correctly classified as belonging to the senses "garment" and "lawsuit" respectively:

> Gene, long haired and laid back, preferring jeans and blazer to a suit, went to film school ...

> The suit was filed in federal court ...

On the other hand, the following context was misclassified as being an instance of the sense "lawsuit":

> Sharpton said, "I have been on the attorney general's case, and I will be on his assistants like a suit jacket throughout the arraignment and the trial."

The analysis of these three instances is obvious. In all three cases, there were relatively reliable indicators for one of the two senses (words like "jeans", "filed", and "attorney"), but in the third case these indicators were misleading. This analysis suggests that the algorithm will

[4]The thesaurus induction algorithm used for the experiment described in this section differs in details from the one presented above, but uses essentially the same method. See [37] for details.

do well if the sense distinctions correspond to distinctions in topic. There is a clear difference between the topics "law" and "clothes". Hence, the algorithm performs well for "suit".

The case of "interest" demonstrates that the algorithm works well even if only one of the senses is clearly topically distinguished. The "interest rate" sense has as many good indicators as the two senses of "suit". However, the general sense of "interest" is not topically well distinguished. Still disambiguation is very reliable.

A much harder case is "space". Both the "outer space" meaning and the more general meaning are fragmented over many different topics. Here are some of the classes that were found in clustering the context vectors:

- NASA space program: the shuttle, satellites, space capsules (class 1)
- space and scientific research (class 3)
- space and weapons, Star Wars (class 8)
- space in art (exhibition space, stage space) (class 4)
- line space (as in "or use the space below to write me") (class 6)
- office and living space (class 7)

The fact that classes 4, 6, and 7 are instances of a more general sense was not recognized. However, this example supports the claim that the distinction between senses is really a continuum. While "office space" and "exhibition space" do not differ enough to justify a sense distinction in the ordinary sense of the word, making such a distinction can be quite useful for an application. From a cognitive perspective, different concepts may well be associated with the two interpretations since the way they are experienced is quite different for most people.

Of the other words in Table 3 "train" is noteworthy for exhibiting a noun/verb ambiguity. The algorithm also distinguishes the three senses of "vessel" correctly.

Table 3 suggests some similarity with the above-mentioned disambiguation method proposed by Yarowsky [41]. However, his method classifies contexts as belonging to one of a predefined set of thesaurus classes (taken from Roget's thesaurus) rather than grouping contexts according to similarity as we do. A strength of Yarowsky's method is that it can incorporate prior knowledge in form of thesaurus classes. One would expect it to do better than a purely corpus-based method if the thesaurus accurately describes semantic properties of words in the corpus of application. On the other hand, it can only make distinctions that are predefined in the thesaurus, so a corpus-based method is likely to be superior if the thesaurus doesn't cover the subject matter of the corpus well.

6 Application to Information Retrieval

The sense disambiguation algorithm described above can be applied directly to information retrieval by replacing the words in a standard "bag of words" text representation by word senses. That is the text is analyzed into words and each word occurrence annotated by a sense as suggested by the disambiguation algorithm. These annotated, and hence differentiated, words may then be used as the basic features for retrieval matching and scoring.

In this way, the disambiguation algorithm was tested on queries 51–75 of the Category B TREC-1 collection, which contains about 170,000 documents from the Wall Street Journal. The queries contain 1013 different terms (excluding stop words). In keeping with the discussion above on fine distinctions that can potentially be exploited, we divided each term into as many senses as could be reliably distinguished from the available data. Our experience indicates that 50 occurrences per sense is a reasonable estimate of the minimum amount of data necessary to distinguish between senses. Therefore, the number of senses for word w was defined to be $\frac{f}{50}$ where f is the number of occurrences of w in the corpus. Only 20 senses were considered even for words that occurred more often than 1000 times. In the experiment, the occurrences of each of the 1013 query terms in the corpus were clustered into the predetermined number of clusters.

For disambiguation, context vectors for each occurrence of one of the query terms in the corpus were computed and assigned to the closest centroid. Each document in the collection was indexed with the automatically assigned senses rather than the terms themselves. Fi-

	word-based	sense-based		combined	
at 0.00	0.693	0.788	+13.7	0.854	+23.2
at 0.10	0.540	0.629	+16.5	0.645	+19.4
at 0.20	0.453	0.503	+11.0	0.506	+11.7
at 0.30	0.385	0.427	+10.9	0.434	+12.7
at 0.40	0.315	0.340	+7.9	0.347	+10.2
at 0.50	0.264	0.260	-1.5	0.291	+10.2
at 0.60	0.206	0.198	-3.9	0.229	+11.2
at 0.70	0.165	0.145	-12.1	0.174	+5.5
at 0.80	0.118	0.110	-6.8	0.129	+9.3
at 0.90	0.085	0.076	-10.6	0.091	+7.1
at 1.00	0.061	0.057	-6.6	0.059	-3.3
average precision (non-interpolated) over all rel docs					
	0.299	0.321	+7.4	0.342	+14.4

Table 4: Disambiguation markedly improves retrieval performance (3 senses per occurrence).

nally, the 25 queries were processed in the same way.[5]

A modification of the standard vector similarity model that represents documents and queries as vectors [33] was then applied for senses instead of words. In the standard model, documents are ranked according to the number of words they share with the query. In the modified model, documents are ranked according to the number of senses (disambiguated words) they share with the query. We used term frequency/inverse document frequency weighting [32] for the senses as well as for the words. We achieved a measurable improvement for sense-based retrieval when compared to word-based retrieval: Average precision for 11 points of recall increased by 4% (from 0.299 to 0.311).

Two modifications were introduced to further improve these results. First, a combination of word-based and sense-based retrieval proved to be better than either on its own. To combine results from both methods, we ranked documents according to word-based and sense-based similarity (ranks r_i^{word} and r_i^{sense} respectively) and combined the ranks for each document. The final ranking was then the ranking of the combined ranks r_i'. This improved average precision by 11% compared with the baseline (from 0.299 to 0.311, see Table 5).

$$r_i' = r_i^{\text{word}} + r_i^{\text{sense}}$$

The second modification concerns our claim that several senses of a word can be used at the same time. This proposal was implemented by coding each word occurrence as having each of the n senses whose clusters were closest to the context vector (the experiment was performed for $n = 2, 3, 4, 5$). Table 5 shows that precision improves when each occurrence is coded with several senses. The best performance was achieved for 3 senses per occurrence using combined ranking.

Table 4 shows relative improvement over word-based retrieval. For each level of recall ("at 0.00" etc.), the table displays the relative improvement in percent of sense-based retrieval and combined retrieval over word-based retrieval.

Figure 1 shows precision for 11 recall points for coding with three senses per occurrence. Average precision for combined ranking is 0.342. This means that the system incorporating word-sense disambiguation achieved a relative improvement of 14% over the basic vector similarity model.[6]

[5] Note that since the queries in the Tipster collection are long (about 150 words), the context vectors for disambiguating query words are in most cases based on 41 words. It is known that disambiguation performance decreases when less context is available [11, 36]. Further research is necessary to determine how much such a deterioration in disambiguation affects performance of the information retrieval model presented here.

[6] Our result is also better than the best result

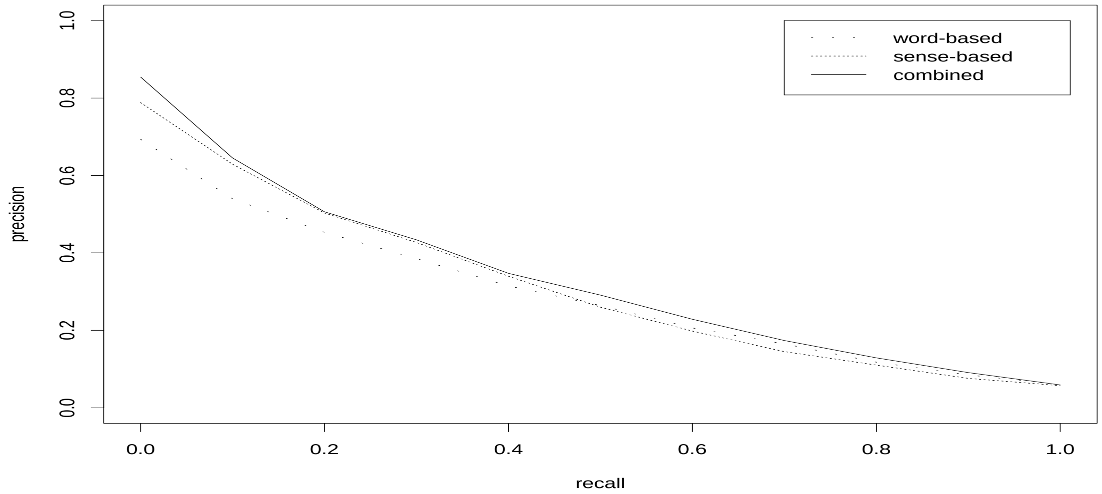

Figure 1: The figure gives precision for 11 recall points for word-based, sense-based, and combined retrieval.

word-based retrieval
av. precision: 0.299

sense-based retrieval

# senses	1	2	3	4	5
av. precision	0.311	0.324	0.321	0.324	0.323

word-based and sense-based combined

# senses	1	2	3	4	5
av. precision	0.324	0.328	0.342	0.339	0.340

Table 5: Coding occurrences with several senses improves average retrieval precision.

7 Conclusion

This paper presents a system that performs sense-based information retrieval. The underlying disambiguation algorithm is completely automatic, but still text-specific since it derives necessary lexical knowledge from the text itself. Its success is partly due to the fact that two assumptions present in most work on word sense disambiguation are avoided: the "one sense per occurrence" and the "fixed number of senses per word" hypotheses. In contrast, our method assumes that multiple senses of a word can be used simultaneously and that even fine sense distinctions can be made if enough data is available.

The disambiguation algorithm is not without limitations. As indicated in several case studies, the method works well only when sense distinctions correspond to distinctions in topic. Still, for the first time it has been demonstrated that disambiguation, even if imperfect, can substantially improve text retrieval performance.

for any of the systems in TREC-1, Category B (0.3219), which indicates that the improvement is not due to an artificially low baseline.

8 Acknowledgment

We would like to thank Marti Hearst, David Hull, Mark Sanderson, John Tukey and two anonymous reviewers for their comments and Michael Berry for SVDPACK.

References

[1] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. Word-sense disambiguation using statistical methods. In *Proceedings of ACL 29*, 1991.

[2] Peter Cheeseman, James Kelly, Matthew Self, John Stutz, Will Taylor, and Don Freeman. AutoClass: A Bayesian classification system. In *Proceedings of the Fifth International Conference on Machine Learning*, 1988.

[3] Kenneth W. Church and William A. Gale. Concordances for parallel text. In *Proceedings of the Seventh Annual Conference of the UW Centre for the New OED and Text Research*, pages 40–62, England: Oxford, 1991.

[4] Garrison W. Cottrell. *A Connectionist Approach to Word Sense Disambiguation.* Pitman, London, 1989.

[5] C. J. Crouch. An approach to the automatic construction of global thesauri. *Information Processing & Management*, 26(5):629–640, 1990.

[6] Douglas R. Cutting, Jan O. Pedersen, David Karger, and John W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of SIGIR '92*, pages 318–329, 1992.

[7] Ido Dagan, Alon Itai, and Ulrike Schwall. Two languages are more informative than one. In *Proceedings of ACL 29*, pages 130–137, Berkeley CA, 1991.

[8] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[9] David A. Evans, Kimberly Ginther-Webster, Mary Hart, Robert G. Lefferts, and Ira A. Monarch. Automatic indexing using selective nlp and first-order thesauri. In *Proceedings of the RIAO*, volume 2, pages 624–643, 1991.

[10] William Gale, Kenneth Ward Church, and David Yarowsky. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proceedings of ACL 30*, pages 249–256, 1992.

[11] William A. Gale, Kenneth W. Church, and David Yarowsky. A method for disambiguating word senses in a large corpus. Technical report, AT&T Bell Laboratories, Murray Hill NJ, 1992.

[12] Stephen I. Gallant. A practical approach for representing context and for performing word sense disambiguation using neural networks. *Neural Computation*, 3(3):293–309, 1991.

[13] Dirk Geeraerts. Vagueness's puzzles, polysemy's vagaries. *Cognitive Linguistics*, 4:223–272, 1993.

[14] Gene H. Golub and Charles F. van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore and London, 1989.

[15] Joe A. Guthrie, Louise Guthrie, Yorick Wilks, and Homa Aidinejad. Subject-dependent co-occurrence and word sense disambiguation. In *Proceedings of ACL 29*, pages 146–152, 1991.

[16] D.K. Harman, editor. *The Second Text REtrieval Converence (TREC-2).* U.S. Department of Commerce, Washington DC, 1994. NIST Special Publication 500-215.

[17] Donna Harman. Overview of the first trec conference. In *Proceedings of SIGIR '93*, 1993.

[18] Marti A. Hearst. Noun homograph disambiguation using local context in large text corpora. In *Seventh Annual Conference of the UW Centre for the New OED and Text*

Research, pages 1–22, England: Oxford, 1991.

[19] Marti A. Hearst and Hinrich Schütze. Customizing a lexicon to better suit a computational task. In Branimir Boguraev and James Pustejovsky, editors, *Corpus Processing for Lexical Acquisition*. MIT Press, Cambridge MA, 1995. To appear.

[20] Graeme Hirst. *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge University Press, Cambridge, 1987.

[21] Yufeng Jing and W. Bruce Croft. An association thesaurus for information retrieval. In *Proceedings of RIAO*, pages 146–160, Rockefeller University, New York, 1994.

[22] Edward Kelly and Phillip Stone. *Computer Recognition of English Word Senses*. North-Holland, Amsterdam, 1975.

[23] Adam Kilgarriff. Dictionary word sense distinctions: An enquiry into their nature. *Computers and the Humanities*, 26:365–387, 1993.

[24] Robert Krovetz and W. Bruce Croft. Word sense disambiguation using machine-readable dictionaries. In *Proceedings of SIGIR '89*, pages 127–136, 1989.

[25] Rovert Krovetz and W. Bruce Croft. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems*, 10(2):115–141, 1992.

[26] Claudia Leacock, Geoffrey Towell, and Ellen Voorhees. Corpus-based statistical sense resolution. In *Proceedings of the ARPA Workshop on Human Language Technology*, Plainsboro NJ, 1993.

[27] Michael Lesk. Automatic sense disambiguation: How to tell a pine cone from an ice cream cone. In *Proceedings of the 1986 SIGDOC Conference*, New York, 1986. Association for Computing Machinery.

[28] Susan W. McRoy. Using multiple knowledge sources for word sense disambiguation. *Computational Linguistics*, 18(1):1–30, 1992.

[29] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. Introduction to WordNet: An on-line lexical database. *Journal of Lexicography*, 3(4):235–244, 1990.

[30] Herbert Charles Morton. *The story of Webster's third: Philip Gove's controversial dictionary*. Cambridge University Press, Cambridge, England, 1994.

[31] Yonggang Qiu and H.P. Frei. Concept based query expansion. In *Proceedings of SIGIR '93*, 1993.

[32] Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288–297, 1990.

[33] Gerard Salton and Michael J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, New York, 1983.

[34] Mark Sanderson. Word sense disambiguation and information retrieval. In *Proceedings of SIGIR '94*, pages 142–151, 1994.

[35] Mark Sanderson, 1995. Electronic mail message to the authors, 26 Jan 1995.

[36] Hinrich Schütze. Dimensions of meaning. In *Proceedings of Supercomputing '92*, pages 787–796, Minneapolis MN, 1992.

[37] Hinrich Schütze. Word space. In Stephen José Hanson, Jack D. Cowan, and C. Lee Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 895–902. Morgan Kaufmann Publishers, San Mateo CA, 1993.

[38] Hinrich Schütze and Jan O. Pedersen. A cooccurrence-based thesaurus and two applications to information retrieval. In *Proceedings of RIAO*, pages 266–274, Rockefeller University, New York, 1994.

[39] Ellen M. Voorhees. Using wordnet to disambiguate word senses for text retrieval. In *Proceedings of SIGIR '93*, pages 171–180, 1993.

[40] Yorick A. Wilks, Dan C. Fass, Cheng ming Guo, James E. McDonald, Tony Plate,

and Brian M. Slator. Providing machine tractable dictionary tools. *Journal of Computers and Translation*, 2, 1990.

[41] David Yarowsky. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of Coling-92*, 1992.

[42] David Yarowsky. One sense per collocation. In *Proceedings of ARPA Human Language Technology Workshop*, 1993.