Using WordNet™ to Disambiguate Word Senses for Text Retrieval

Ellen M. Voorhees
Siemens Corporate Research, Inc.
755 College Road East
Princeton, NJ 08540
ellen@learning.scr.siemens.com

Abstract

This paper describes an automatic indexing procedure that uses the "IS-A" relations contained within WordNet and the set of nouns contained in a text to select a sense for each polysemous noun in the text. The result of the indexing procedure is a vector in which some of the terms represent word senses instead of word stems. Retrieval experiments comparing the effectiveness of these sense-based vectors vs. stem-based vectors show the stem-based vectors to be superior overall, although the sense-based vectors do improve the performance of some queries. The overall degradation is due in large part to the difficulty of disambiguating senses in short query statements. An analysis of these results suggests two conclusions: the IS-A links define a generalization/specialization hierarchy that is not sufficient to reliably select the correct sense of a noun from the set of fine sense distinctions in WordNet; and missing correct matches because of incorrect sense resolution has a much more deleterious effect on retrieval performance than does making spurious matches.

1 Introduction

Retrieval systems that employ automatic indexing techniques to create text representatives from natural language must deal with the problems of polysemy and synonymy. Polysemy, a single word form having more than one meaning, depresses precision by causing false matches, while synonymy, multiple words having the same meaning, depresses recall by causing true conceptual matches to be missed. In principle, polysemy and synonymy can be handled by assigning different senses of a word different *concept identifiers* and assigning the same concept identifier to synonyms. In practice, this requires procedures that are capable of recognizing synonyms, and that can not only detect uses of different senses of a word but can also resolve which meaning is intended in each case.

This paper describes an experiment in which a completely automatic indexing procedure attempts to detect and resolve the senses of the polysemous nouns occurring in the texts of documents and queries. In particular, the procedure selects a single WordNet synonym set as the meaning of each noun. A synonym set is selected on the basis of the difference between the pattern of synonym sets that are visited for the given text and the pattern produced for the collection as a whole. The result of the indexing procedure is a vector in which some of the concepts represent word senses (the synonym sets) instead of word stems. The efficacy of the disambiguation procedure is tested by comparing the retrieval effectiveness of the resulting sense-based vectors to the effectiveness of stem-based vectors for five standard test collections.

The disambiguation procedure was developed as a possible method for exploiting the semantics contained within WordNet to improve retrieval effectiveness. Accordingly, the next section of the paper describes WordNet in some detail to provide

[†]WordNet is a trademark of Princeton University.

ACM-SIGIR'93-6/93/Pittsburgh, PA, USA
© 1993 ACM 0-89791-605-0/93/0006/0171...$1.50

the appropriate background. Section 3 motivates and explains the disambiguation procedure itself. The following section presents the results of the retrieval runs, including the results of two variations of the basic procedure. The final section summarizes the results.

2 WordNet

WordNet is a manually-constructed lexical system developed by George Miller and his colleagues at the Cognitive Science Laboratory at Princeton University [4]. Originating from a project whose goal was to produce a dictionary that could be searched conceptually instead of only alphabetically, WordNet evolved into a system that reflects current psycholinguistic theories about how humans organize their lexical memories. The basic object in WordNet is a set of strict synonyms called a *synset*. By definition, each synset in which a word appears is a different sense of that word.

There are four main divisions in WordNet, one each for nouns, verbs, adjectives, and adverbs. Within a division, synsets are organized by the lexical relations defined on them. For nouns, the only division of WordNet used in this study, the lexical relations include antonymy, hypernymy/hyponymy (IS-A relation) and three different meronym/holonym (PART-OF) relations. The IS-A relation is the dominant relation, and organizes the synsets into a set of approximately ten hierarchies[1]. As an example, Figure 1 shows the IS-A hierarchy relating the eight different senses of the noun 'board'. The synsets with the heavy border are the actual senses of 'board', and the remaining synsets are either ancestors or descendents of one of the senses. Additional relations, such as (one sense of) 'director' is a MEMBER-OF the committee sense of 'board', are included in WordNet but are not shown in the figure. The synsets {*group, grouping*} and {*entity, thing*} in the figure are examples of heads of hierarchies. Other heads include {*act, human_action, human_activity*}, {*abstraction*}, {*possession*}, and {*psychological_feature*}.

For any lexical system to be usable with real text, it must provide a means to map morphological variants of a word to the information for that word. WordNet provides such access code, but we developed our own routine to access the WordNet information that differs from the official code[2]. In our version, the access routine takes a word (a string of characters), converts it to lower case, and checks if the converted string occurs in the noun portion of WordNet. If the string is found, the routine returns either the synsets in which the string appears, the fact that the string is a known irregular morphological variant of a member of a synset (e.g., 'women' is an inflection of 'woman'), or both (e.g., 'media' is both a member of {*media, mass_media*} and an inflection of 'medium'). If the string is not found, several simple regular morphological variants of the word are tried. If none is found, the routine reports the string as not found. Otherwise, the routine returns the base form. An unintended and usually unfortunate consequence of this simple strategy is that regular plural forms that are members of their own synsets do not return the synsets of the base word since once the plural form is found in WordNet no variants are tried. For example, 'arms' returns the synsets {*coat_of_arms, arms, blazon, blazonry*} and {*weaponry, arms, implements_of_war*}, but not the four synsets for 'arm'.

WordNet 1.2 (April, 1992), the version of WordNet used in this study, contains 35,155 synonym sets and 67,293 senses[3] in the noun division. Because synsets contain only strict synonyms, the majority of synsets are quite small. Similarly, the average number of senses per word is close to one. (Table 1 gives statistics about the distributions of synset sizes and senses per word.) These figures seem to suggest that polysemy and synonymy occur too infrequently to be a problem for retrieval, but they are misleading. The more frequently a word is used, the more polysemous it tends to be [11]. The more common words also tend to appear in the larger synsets. Thus it is precisely those nouns that actually get used in documents

[1]The actual structure is not quite a hierarchy since some synsets have more than one parent.

[2]Our version of the morphology code was developed before the official morphology code was available. Recognized as a quick-and-dirty approach, it continues to be used as a result of inertia.

[3]This number and all other numbers that pertain to senses include 6829 irregular morphological variants our access code treats as "senses" even though they appear in no synonym set.

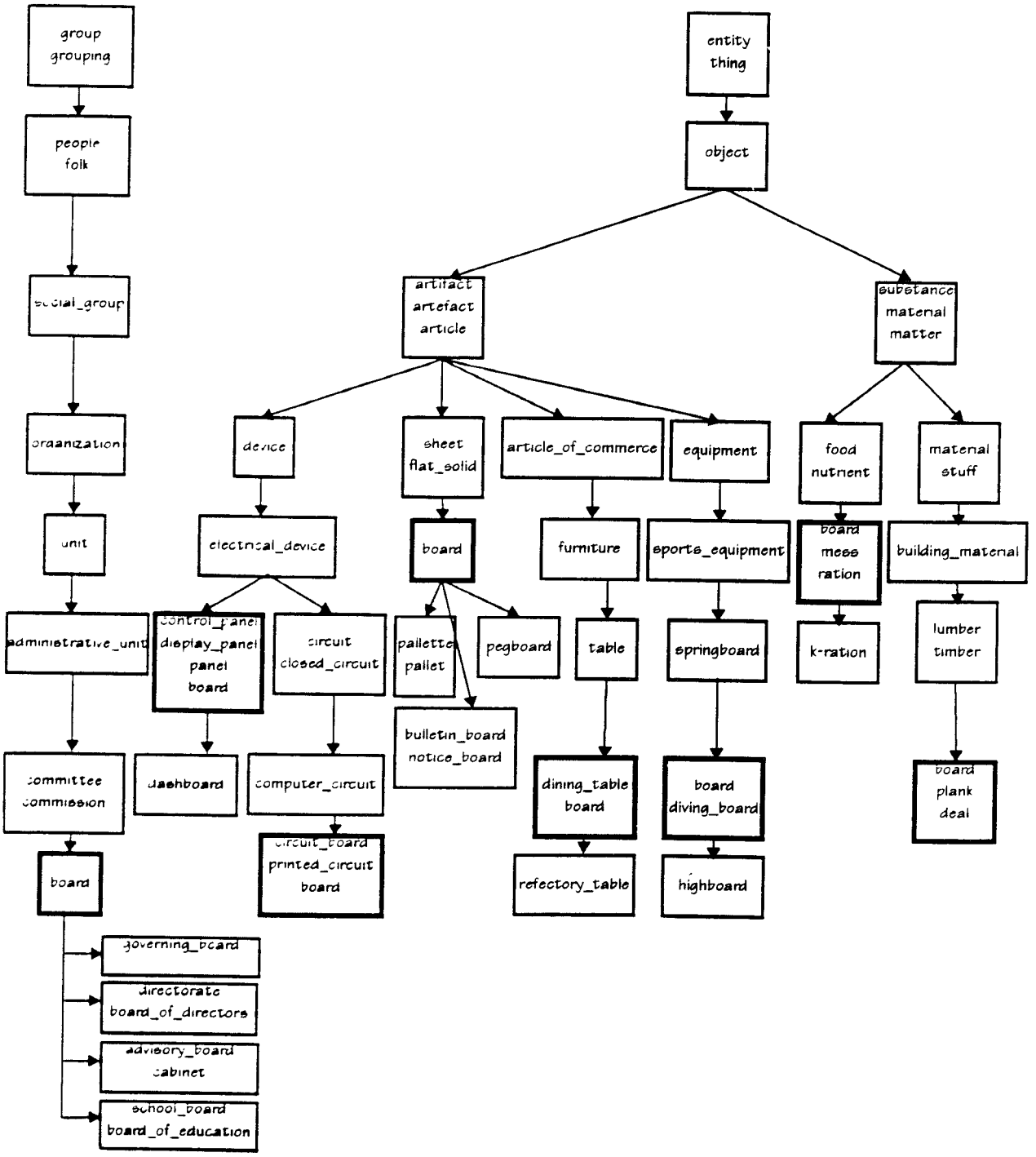

Figure 1: The IS-A hierarchy for eight different senses of the noun "board".

Size	# synsets	% synsets
1	18114	52
2	11361	32
3	3694	11
4	1292	3.6
5	396	1.1
6	171	.49
7	53	.15
8	32	.09
9	18	.05
10	5	.01
> 10	19	.05

Largest synset size: 38
Mean synset size: 1.74

Senses	# words	% words
1	45686	85
2	5067	9
3	1508	2.8
4	638	1.2
5	321	.6
6	164	.3
7	83	.15
8	55	.1
9	30	.05
10	27	.05
> 10	19	.04

Most # of senses: 27
Mean # of senses: 1.26

Table 1: Synset size (number of words in synset) and number of senses per word for nouns

and query statements that are most likely to have many senses and synonyms.

3 Sense Disambiguation

Research into the automatic resolution of word senses has been going on for at least forty years, and thus there is a large literature describing a variety of different techniques. This paper will not attempt a review of all these techniques; Krovetz and Croft [3] and Voorhees, Leacock, and Towell [10] review many of the methods. While general sense disambiguation techniques have been studied, little has been published about the direct application of sense disambiguation in IR. A notable exception is the recent study by Krovetz and Croft [3]. They examine two test collections (the CACM and TIME collections) to study both the amount of lexical ambiguity in the collections and its effect on retrieval performance. They find that even these relatively small, specialized collections contain words used in multiple senses, but that retrieval effectiveness is not strongly affected by ambiguity, in part because documents with many words in common with a query (and are thus ranked highly with regard to that query) tend to use the words in the same senses as the query. They hypothesize that collections that contain more diverse subject matter, and high-recall searches that depend on matches of single concepts will benefit more from disambiguation. They further observe that disambiguation is a critical step in exploiting relation among words (e.g., expanding a concept by its synonyms) since senses, not word forms, are the true participants in these relations. These conjectures are supported by retrieval results obtained on the large TREC collection [9].

The particular disambiguation technique used in this work is based on the idea that a set of words occurring together in context will determine appropriate senses for one another despite each individual word being multiply ambiguous. A common example of this effect (see [6]) is the set of nouns *base*, *bat*, *glove*, and *hit*. While most of these words have several senses, when taken together the intent is clearly the game of baseball. To exploit this idea automatically, a set of categories representing the different senses of words needs to be defined. Once such categories are defined, the number of words in the text that have senses that belong to a given category is counted. The senses that correspond to the categories with the largest counts are selected to be the intended senses of the ambiguous words. Obviously, the category definitions are a critical component of this procedure. Sedelow and Mooney report on a study in which the categories are defined by *Roget's Third International Thesaurus* classes [7]. In

an independent study Slator uses the subject codes available in the machine-readable version of the *Longman Dictionary of Contemporary English* [8]. Both of these studies report some success at disambiguating word senses, but neither applies the method to the retrieval problem.

Like *Roget*'s and *Longman*'s, WordNet also groups noun senses into IS-A hierarchies, but it contains no obvious division to use to define categories. Using each separate hierarchy as a category is well defined but too coarse grained. For example, in Figure 1 seven of the eight senses of board are in the {*entity, thing*} hierarchy. Similarly, using individual synsets is well defined but too fine grained. Given the small mean size of a synset, no distinctions can be made between different senses if only the words in the synset itself are counted: the synsets for both the committee sense of board and the flat solid sense of board contain only the word 'board'. On the other hand, defining an appropriate middle level is difficult. For example, the very specific noun 'k-ration' and the fairly general collocation 'administrative unit' are each six nodes down from the root of their respective hierarchies.

A new construct called a *hood* is used to resolve this difficulty[4]. To define the hood of a given synset, s, consider the set of synsets and the hyponymy links in WordNet as the set of vertices and directed edges of a graph. Then the hood of s is the largest connected subgraph that contains s, contains only descendents of an ancestor of s, and contains no synset that has a descendent that includes another instance of a member of s as a member. A hood is represented by the synset that is the root of the hood. For example, in Figure 1 the hood of the synset for the committee sense of board is rooted at the synset {*group, grouping*} (and thus the hood for that sense is the entire hierarchy in which it occurs), the hood for the computer circuit sense of board is rooted at {*circuit, closed_circuit*}, and the hood for the panel sense of board is rooted at the synset itself. Because some synsets have more than one parent, synsets can have more than one hood. A synset has no hood if the same word is a member of both the synset and one of its descendents.

[4]The general idea of a hood as the area in WordNet in which a word is unambiguous was suggested by George Miller.

The hoods of the WordNet synsets can be used as sense categories in much the same manner as *Longman* subject codes or *Roget* classes. A word that occurs in the hood of a sense of an ambiguous word is evidence for that sense. The sense of an ambiguous word in a given text can be selected by counting the number of other words in the text that occur in each of the different sense's hoods and choosing the hood with the largest number. Of course, since the text is likely to contain other ambiguous words, the disambiguation process must be capable of resolving multiple ambiguous words simultaneously.

The senses of the nouns in a text of a given collection are selected by the following two stage process. A marking procedure that visits synsets and maintains a count of the number of times each synset is visited is fundamental to both stages. Given a word, the procedure finds all instances of the word in (the noun portion of) WordNet. For each identified synset, the procedure follows the IS-A links up to the root of the hierarchy incrementing a counter at each synset it visits. In the first stage the marking procedure is called once for each occurrence of a content word (i.e., a word that is not a stop word) in all of the documents in the collection. The number of times the procedure was called and found the word in WordNet is also maintained. This produces a set of *global counts* (relative to this particular collection) at each synset. In the second stage, the marking procedure is called once for each occurrence of a content word in an individual text (document or query). Again the number of times the procedure was called and found the word in WordNet for the individual text is maintained. This produces a set of *local counts* at the synsets. Given the local and global counts, a sense for a particular ambiguous word contained within the text that generated the local counts is selected as follows:

- The difference

$$\frac{\#\text{ local visits}}{\#\text{ of calls in stage 2}} - \frac{\#\text{ global visits}}{\#\text{ of calls in stage 1}}$$

  is computed at the root of the hood for each sense of the word. If a sense does not have a hood or if the local count at its hood root is less than two, that difference is set to zero. If a sense has multiple hoods, that difference

is set to the largest difference over the set of hoods.

- The sense corresponding to the hood root with the largest positive difference is selected as the sense of the word in the text. If no sense has a positive difference, no WordNet sense is chosen for the word.

The idea behind this disambiguation procedure is to select senses from the areas of the WordNet hierarchies in which document-induced (local) activity is greater than the expected (global) activity. The hood construct is designed to provide a point of comparison that is broad enough to encompass markings from several different words yet narrow enough to distinguish among senses. A possible disadvantage of the method is that it does not make use of the prior probability of a sense. That is, no preference is given to the sense that occurs most frequently.

4 Retrieval Experiments

4.1 Retrieval Environment

Creating sense-based vectors is a straight-forward application of the disambiguation procedure. The retrieval model used is the extended vector space model of information retrieval introduced by Fox in which each vector is comprised of subvectors of different concept types (called *ctypes*) [2]. In this experiment, a vector may contain three ctypes: stems of words that do not appear in WordNet, synonym set id's of disambiguated nouns, and stems of the disambiguated nouns. The second and third ctypes are alternative representations of the text in that the same text word causes an entry in both ctypes. The noun word stems are kept to act as a control in the experiment. The members of the first ctype include the words that are not nouns, nouns that are not in WordNet (e.g., proper nouns and technical terms), and nouns that could not be disambiguated (because no sense had a positive difference).

In the extended vector space model the similarity between a pair of document and query vectors is computed as the weighted sum of the similarities between ctypes:

$$sim(D,Q) = \sum_{\text{ctype } i} \alpha_i sim_i(D_i, Q_i)$$

where sim_i is the similarity function for ctype i, D_i and Q_i are the ith subvectors of vectors D and Q, and α_i, a real number, reflects the importance of ctype i relative to the other ctypes.

For the current experiment, each of the concepts is weighted using a tf×idf weight. The tf×idf weights are then normalized by the square root of the sum of the squares of the weights *of the entire vector*. This weight differs from the "tfc" weights described by Salton and Buckley [5] in that the tfc weights normalize each ctype independently. The ctypes used here vary widely in length both across ctypes and across vectors within the same ctype. In such cases, cosine normalization by ctype can place undue emphasis on terms in short ctypes. Normalizing by the entire vector avoids this problem. The inner product is used as the similarity measure for each ctype.

4.2 Sense-based Retrieval

To judge the effectiveness of the sense resolution procedure in retrieval, the performance of sense vectors is compared to the performance of a standard run. In the standard run, both document and query vectors contain one ctype that consists of word stems for all content words. The terms in the vectors are weighted using the "ntc" weight of Salton and Buckley [5], which is a variant of a cosine-normalized, tf×idf weight that emphasizes the tf component; Salton and Buckley found this to be a good weight. Since the weights are normalized, an inner product similarity measure is used. The resulting similarity is identical to a cosine measure used on unnormalized weights.

Table 2 gives performance figures for the standard run and three different sense-based vector runs for five collections. The three sense-based vector runs differ in the ctype weights (the α's in the similarity function above) that are used. The run labeled '110' gives equal weight to the non-noun word stems (ctype 1) and the synset id's (ctype 2) and ignores the noun word stems (ctype 3). The run labeled '211' gives the non-noun word stems twice the weight given to each of the synset

Collection	Standard		110			211			101		
	#	3-pt	#	3-pt	%	#	3-pt	%	#	3-pt	%
CACM	228	.3291	139	.1994	-39.4	163	.2594	-21.2	199	.2998	-8.9
CISI	368	.2426	271	.1401	-42.3	318	.1980	-18.4	344	.2225	-8.3
CRAN	822	.4246	570	.2729	-35.7	680	.3261	-23.2	718	.3538	-16.7
MED	260	.5527	228	.4405	-20.3	246	.4777	-13.6	241	.4735	-14.3
TIME	267	.6891	241	.6044	-12.3	253	.6462	-6.2	256	.6577	-4.6

Table 2: Number relevant retrieved and 3-point average precision for baseline runs

id's and the noun word stems. The final run ('101') is a control run. All of the word stems get equal weight and the synset id's are ignored. This is *not* equivalent to a single-ctype word stem run since matches between query and document vectors must occur within the same ctype. The performance measures used are the total number of relevant documents retrieved over all queries and the mean precision[5] at recall points .2, .5, and .8. For the sense-based vector runs, the percentage difference of the 3-point average over the standard run's 3-point average is also given.

Since many words that are used as verbs and adjectives in the texts are nonetheless found in the noun division of WordNet[6], the sense-vector retrieval runs are also performed using text that has parts of speech tagged. The text of the collections is processed using Eric Brill's stochastic tagger [1], and is then indexed. The marking procedure of the disambiguation process is called only if the word is tagged as being a noun. All other processing is the same as above. The results of these runs are given in Table 3.

Clearly the effectiveness of the sense-based vectors is worse than that of the stem-based vectors, sometimes very much worse; part of speech tagging makes little difference in terms of retrieval effectiveness. Examination of the individual query results shows that most of this degradation is caused by matches between documents and queries that are made in the standard run but missed in the sense-based runs. The missed matches have several causes: different senses of a noun being chosen for documents and queries when in fact the same sense is used; the inability to select any senses in some queries due to lack of context; and adjectives and verbs that conflate to the same stem as a noun in the standard run but are maintained as separate concepts in the sense-based runs.

As an example of these effects, consider query 16 of the MED collection. The query, requesting documents on separation anxiety in infant and preschool children, retrieves 7 relevant documents in the top 15 for the standard run and only 1 relevant document in the top 15 for the '110' run (without part-of-speech tagging). The problem is selecting the sense of 'separation' in the query. WordNet contains eight senses of the noun 'separation'. With few other words to use in making a selection, the indexing procedure selects a sense that is not selected for any document. The separation concept is dropped from the query, and retrieval performance suffers accordingly.

The importance of finding matches between document and query terms is underscored by the degradation in performance of the control run '101' compared to the standard run. The only major difference between the control run, which ignores the senses and just uses the word stems, and the standard run is the introduction of ctypes. Ctype 1 receives all words that do not occur in the noun division of WordNet and any noun that cannot be disambiguated. Ctype 3 receives all word stems of nouns that can be disambiguated. Since the similarity measure only looks for matches within ctypes, adjectives and verbs that conflate to the same stem as a noun match that noun in the stan-

[5]Although the tables present only 3-point averages, entire recall-precision graphs were computed. There are essentially no differences in the relative performance of the different methods at the high-precision or high-recall ends of the scale from that shown by the average performance.

[6]For example, when indexing a query about fatty acids in the MED collection, the indexing procedure found 'fatty' in the WordNet nouns as an obese person.

Collection	Standard		110			211			101		
	#	3-pt	#	3-pt	%	#	3-pt	%	#	3-pt	%
CACM	228	.3291	133	.1941	-41.0	165	.2502	-24.0	195	.2852	-13.3
CISI	368	.2426	265	.1601	-34.0	315	.2042	-15.8	340	.2282	-5.9
CRAN	822	.4246	599	.2924	-31.1	682	.3361	-20.8	713	.3584	-15.6
MED	260	.5527	231	.4515	-18.3	246	.4855	-12.2	245	.4835	-12.5
TIME	267	.6891	244	.6333	-8.1	256	.6765	-1.8	260	.6698	-2.8

Table 3: Number relevant retrieved and 3-point average precision for part-of-speech tagged runs

dard run but do not match in the 101 run. Nouns that cannot be disambiguated in a query due to the lack of context have an even greater impact because they happen more frequently. Some sense is almost always selected for these nouns in the documents, so no match is found between the document and query for this term.

There *are* a few queries that are helped by the disambiguation procedure. CACM query 23, for example, requests documents on distributed computing structures and algorithms. In the standard run, no relevant documents are retrieved in the top 15 because most of the documents retrieved discuss algorithms for computing statistical distribution functions. The '110' run retrieves four relevant documents in the top 15 because the statistical distribution documents are no longer retrieved. The overall retrieval results indicate, however, that the damage caused by missing correct matches more than offsets the gain from eliminating false matches, at least for these small, homogenous collections.

4.3 Disambiguating in Documents Only

Since most of the degradation in retrieval performance occurs because short query statements are difficult to disambiguate, it is interesting to investigate the retrieval performance when only the nouns in documents are disambiguated. Instead of selecting a single sense of an ambiguous noun for a query, *all* synset id's of the noun are added to the query vector. Synset id's are weighted such that the term frequency component of each id's weight is the term frequency of the word in the query divided by the number of senses of the word in WordNet (i.e., $tf_{id} = \frac{tf_{word}}{\#\text{ of senses of word}}$). The retrieval results for document-only disambiguation with tagged parts of speech are given in Table 4.

For most of the collections, the retrieval results exhibit the same sort of degradation as the control runs in the sense-based retrieval. This is consistent with the cause of the degradation being mismatches in the ctypes of terms. Note that the MED collection has a minimal effectiveness improvement for the runs that include synset id's compared to the standard run. This improvement is attributable to the gains for one query, query 20, and demonstrates the potential benefits of concept, as opposed to word form, indexing. Query 20 requests documents that discuss the effects of *somatotropin*, a human growth hormone. Many of the relevant documents use the variant spelling 'somatotrophin' for the hormone and thus are not retrieved in the standard run. However, the synset that represents the hormone includes both spellings as members of the set. Documents that use either spelling are indexed with the same synset identifier and match the query.

4.4 Disambiguation

So far nothing has been said about how good this disambiguation procedure is at selecting the correct sense of an ambiguous term except to say that it does not do well for short query statements. No systematic evaluation of the disambiguation technique itself has been done since such an evaluation requires the knowledge of which WordNet sense is the correct sense for all the nouns in all the collections' texts. A subjective evaluation obtained while looking at the individual query re-

Collection	Standard		110			211			101		
	#	3-pt	#	3-pt	%	#	3-pt	%	#	3-pt	%
CACM	228	.3291	186	.2722	-17.3	202	.3073	-6.6	207	.3052	-7.3
CISI	368	.2426	318	.2101	-13.4	342	.2314	-4.6	349	.2337	-3.7
CRAN	822	.4246	722	.3667	-13.6	775	.3984	-6.2	784	.4042	-4.8
MED	260	.5527	266	.5631	1.9	269	.5670	2.6	262	.5457	-1.3
TIME	267	.6891	259	.6784	-1.6	262	.6878	-0.2	266	.6879	-0.2

Table 4: Number relevant retrieved and 3-point average precision for document-only disambiguation

trieval results suggests that the technique is not a reliable method for choosing among the fine sense distinctions WordNet makes. As an example of why this is true, consider the 'board' example of Figure 1. The nouns 'nail', 'hammer', and 'carpenter' are all good hints that the intended sense of board is the lumber sense. However, 'nail' is a fastener, which in turn is a device. Thus 'nail' would help select the control panel sense of board. 'Hammer' is a tool, which is an implement, which is an article of commerce. Thus 'hammer' would help select the dining table sense of board. Finally, 'carpenter' is a worker, which is a person, which is both an agent and a life form, which are both 'thing's. So 'carpenter' would not help select any sense of board. This analysis indicates that specialization/generalization relations are unlikely to contain sufficient information to choose among fine sense distinctions.

5 Conclusion

The retrieval experiments described in this paper attempt to exploit the semantics contained within WordNet to improve retrieval effectiveness by indexing with word senses instead of word stems. The results show that the effectiveness of the vectors produced by this disambiguation technique is worse than word stem vectors for all five collections. Much of the degradation is due to the difficulty of disambiguating word senses in the short query statements: with little context to use in disambiguating, the indexing procedure either does not attempt to resolve the disambiguity or selects an incorrect sense. In either case, the query no longer matches documents in which the sense is correctly resolved. These results demonstrate that missing matches between the documents and query degrades performance more than eliminating spurious matches helps retrieval for small, homogenous collections. Nevertheless, some queries do exhibit the performance improvements that concept-based retrieval suggests is possible.

As mentioned earler in the paper, this disambiguation procedure ignores the prior probability of a given sense occurring. Yet short query statement are intelligible to humans because the statement is seen in the context of the particular domain covered by the document collection. A possible solution to the problem of disambiguating short query statements, therefore, may be simply to select the most frequent sense unless there is strong evidence to the contrary.

Although the stem-based vectors are more effective overall than the sense-based vectors in this experiment, I concur with Krovetz and Croft's opinion that sense resolution is an important component for future retrieval systems [3]. Collections more diverse than those studied here will have more lexical ambiguity that will affect retrieval effectiveness. In a separate study of the effects of vector expansion in the large TREC collection, we found at least two queries in which the ambiguity of a highly-weighted query term caused poor retrieval performance: road salt vs. the strategic arms limitation treaty (SALT); and demographic shifts vs. automatic transmission shifting mechanisms vs. work periods (i.e., "the night shift"). The same study concluded that some sort of sense-resolution process is necessary for vector expansion to be effective [9]. As a result, we are investigating other ways of automatically disambiguating word senses. An important lesson learned from this study is that unless such a method can cope

with very small contexts (short query statements) it will not be useful for retrieval.

Acknowledgements

Chris Buckley, Chris Darken, Gary Kuhn, George Miller, and Geoff Towell reviewed drafts of this paper and made suggestions that improved its presentation and clarity. The anonymous referees also made insightful comments. I gratefully acknowledge their efforts.

References

[1] Eric Brill. A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Computational Linguistics (ACL)*, 1992.

[2] Edward A. Fox. *Extending the Boolean and Vector Space Models of Information Retrieval with P-norm Queries and Multiple Concept Types.* PhD thesis, Cornell University, 1983. University Microfilms, Ann Arbor, MI.

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

[4] George Miller. Special Issue, WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4), 1990.

[5] Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24:513–523, 1988.

[6] Gerard Salton and Michael E. Lesk. Information analysis and dictionary construction. In Gerard Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, chapter 6, pages 115–142. Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1971.

[7] Sally Yeates Sedelow and Donna Weir Mooney. Knowledge retrieval from domain-transcendent expert systems: II. research results. In *Proceedings of the 51st Annual Meeting of the American Society of Information Science*, pages 209–212, 1988.

[8] Brian Michael Slator. *Lexical Semantics and Preference Semantics Analysis.* PhD thesis, New Mexico State University, Las Cruces, NM, December 1988.

[9] Ellen M. Voorhees and Yuan-Wang Hou. Vector expansion in a large collection. In *Proceedings of the First Text Retrieval Conference*, 1992. Proceedings to appear.

[10] Ellen M. Voorhees, Claudia Leacock, and Geoffrey Towell. Learning context to disambiguate word senses. In *Proceedings of the 3rd Computational Learning Theory and Natural Learning Systems Conference*, 1992. Proceedings to appear. Also available as Siemens technical report.

[11] G.K. Zipf. The meaning-frequency relationship of words. *Journal of General Psychology*, 3:251–256, 1945.