Word Sense Disambiguation for Free-text Indexing Using a Massive Semantic Network

Michael Sussna
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, California 92093-0114

sussna@cs.ucsd.edu

Abstract

Semantics-free, word-based information retrieval is thwarted by two complementary problems. First, search for relevant documents returns irrelevant items when all meanings of a search term are used, rather than just the meaning intended. This causes low precision. Second, relevant items are missed when they are indexed not under the actual search terms, but rather under related terms. This causes low recall. With semantics-free approaches there is generally no way to improve both precision and recall at the same time.

Word sense disambiguation during document indexing should improve precision. We have investigated using the massive WordNet semantic network for disambiguation during indexing. With the unconstrained text of the SMART retrieval environment, we have had to derive our own content description from the input text, given only part-of-speech tagging of the input.

We employ the notion of semantic distance between network nodes. Input text terms with multiple senses are disambiguated by finding the combination of senses from a set of contiguous terms which minimizes total pairwise distance between senses. Results so far have been encouraging. Improvement in disambiguation compared with chance is clear and consistent.

Keywords: Information retrieval, indexing, word sense disambiguation, semantic networks, free-text.

1 Introduction

Semantics-free, word-based information retrieval is thwarted by two complementary problems. First, search for relevant documents returns irrelevant items when all meanings of a search term are used, rather than just the meaning intended. This is the polysemy/false positives/low precision problem. Second, relevant items are missed when they are indexed not under the actual search terms, but rather under related terms. This is the synonymy/false negatives/low recall problem. With semantics-free approaches there is generally no way to improve both precision and recall at the same time. Increasing one is done at the expense of the other [Salton and McGill, 1983; van Rijsbergen, 1983]. For example, casting a wider net of search terms to improve recall of relevant items will also bring in an even greater proportion of irrelevant items, lowering precision.

There is a many-to-many mapping between word forms and word meanings. A single word form can have multiple meanings, and a single meaning can be expressed by multiple word forms. Both of these multiplicities cause problems for any approach to content search based on word forms. We believe that in order to do near-human level retrieval we must go beyond words and get at meanings. Text disambiguation during indexing should improve precision by combating polysemy [Krovetz and Croft, 1992]. We are looking into reducing the ambiguity of word forms during indexing by taking advantage of semantic networks. A number of these networks already exist and their implementation is fairly straightforward.

As part of a larger research project exploring the exploitation of explicit semantics for overcoming both the polysemy and synonymy problems, we have performed preliminary investigations of document indexing using a massive semantic network, WordNet. WordNet is a network of word meanings connected by a variety of lexical and semantic relations. Over 35,000 word senses are represented in the noun portion of WordNet alone. We have been working with WordNet in the SMART information retrieval environment. In the unconstrained text of the SMART environment, no index terms have been assigned [Buckley, 1985]. We have had to derive our own content description from the input text, given only part-of-speech tagging of the input.

Employing the notion of semantic distance between network nodes, we have run a series of experiments. Input text terms with multiple senses have been disambiguated by finding the combination of senses from a set of contiguous terms which minimizes total pairwise distance between senses. Results so far have been encouraging. Improvement in disambiguation compared with chance is clear and consistent, strongly suggesting that semantics-based indexing is worth pursuing further for transcending the polysemy problem. It is competitive with word-based approaches. A number of these have focused on only a few fixed terms whose senses were to be distinguished, rather than on unconstrained text [Lesk, 1986; Wilks *et al.*, 1989; Voorhees *et al.*, 1992].

In the following sections we will discuss the research environment, network-based disambiguation, the experiments performed and results obtained.

CIKM '93 - 11/93/D.C., USA
© 1993 ACM 0-89791-626-3/93/0011 ....$1.50

2 Research Environment

The current project uses the SMART information retrieval environment. In SMART, documents do not have keyword descriptors. Instead, one must do one's own indexing of content. We are investigating using semantics for word sense disambiguation during document indexing.

Semantics are supplied by the WordNet lexical/semantic database developed at the Cognitive Science Laboratory at Princeton University [Miller *et al.*, 1990; Miller, 1990]. Of particular relevance and usefulness for our research is the noun portion of WordNet, which contains over 35,000 word meanings represented as network nodes called "synsets" (synonym sets). Each sense of a word maps to a distinct synset. For example, one sense of the noun "strike" maps to (hit rap strike tap) which IS-A (impact bump thump blow); another maps to (strike work_stoppage) which IS-A (direct_action).

We work with the *Time Magazine* article collection, since it is the least specialized and technical, because WordNet is a general English lexicon.

With SMART, the words in the documents are converted to lower case and parsed into strings. They can be stemmed down to base forms; e.g., "stemmed" and "stems" both become "stem." Input words can also be labeled by part of speech, which is a feature that we took advantage of. Although the part-of-speech tagger employed was not infallible, it was accurate enough to give us a good working set of nouns to serve as input to semantic processing.

One aspect of this input editing process which is a source for limiting the effectiveness of our efforts is the filtering out of terms. SMART uses a list of "stopwords," words to be ignored as "contentless." For example, prepositions, conjunctions, and articles are considered extraneous. After stopwords have been removed, and non-nouns removed from what remains, very little of the original article is left. So, we are working with a sparse sample of the original text by the time we get to decide which sense of each noun is intended. Nouns found in WordNet are the final distillation that we begin to work with during disambiguation.

The following example illustrates the filtering process. It uses an excerpt from *Time* document 1, shown after successive filtering steps.

After conversion to lowercase (part-of-speech tagging is omitted for readability; the first four words are actually the title):

the allies after nassau in december 1960, the u.s . first proposed to help nato develop its own nuclear strike force . but europe made no attempt to devise a plan . last week, as they studied the nassau accord between president kennedy and prime minister macmillan, europeans saw emerging the first outlines of the nuclear nato that the u.s . wants and will support . it all sprang from the anglo-u.s . crisis over cancellation of the bug-ridden skybolt missile, and the u.s . offer to supply britain and france with the proved polaris (time, dec . 28).

After stopword removal:

allies . proposed nato develop nuclear strike force made attempt devise plan . week studied accord president kennedy prime minister macmillan emerging outlines nuclear nato . support sprang anglo crisis cancellation bug ridden skybolt missile offer supply britain france proved polaris time dec

Nouns in WordNet:

allies strike force attempt plan week accord president prime minister outlines support crisis cancellation bug missile france polaris time

WordNet's noun portion has fairly rich connectivity as well as obvious comprehensiveness. The WordNet noun nodes are connected by nine relations. Eight of these form four pairs of complementary or inverse relations, while one is its own inverse. There is actually a tenth relation that is implicit in the network structure, but does not label any net edges because it is intranode rather than internode. The relations are:

`synonymy`	`(has same meaning as; intranode)`
`hypernymy`	`(is a)`
`hyponymy`	`(has instance)`
`holonymy`	`(is part of, is substance in, is member of; 3 relations)`
`meronymy`	`(has part, contains substance, has member; 3 relations)`
`antonymy`	`(is complement of; self-inverse)`

Hypernymy and hyponymy are the strictly hierarchical links. The holonymy/meronymy relations can also be considered "vertical" relations. Vertical relations are asymmetrical and order items. Synonymy and antonymy are "horizontal," symmetrical, non-ordering relations (and of course are non-hierarchical).

3 Net-based Disambiguation

We have tried a variety of approaches to term disambiguation, all based on minimizing an objective function utilizing semantic distance between topics in WordNet. It is outside the scope of this paper to explain the distance determination logic. We will, however, describe the salient aspects of the network edge weighting scheme because this background is necessary for discussion of the experiments where the network weights were varied.

3.1 Edge weighting

Each edge consists of two inverse relations. Each relation type has a weight range between its own *min* and *max*. The point in the range for a particular arc depends on the number of arcs of the same type leaving the node. This is the *type-specific fanout* (TSF) factor. TSF reflects dilution of the *strength of connotation* between a source and target node as a function of the number of like relations that the source node has.[1] The two inverse weights for an edge are averaged. The average is divided by the depth of the edge within the overall "tree." This process is called *depth-relative scaling* and it is based on the observation that only-siblings deep in a tree are more closely related than only-siblings higher in the tree.

Definition 1

The edge between adjacent nodes A and B has distance or weight

[1] This factor takes into account the possible asymmetry between two nodes, where the strength of connotation in one direction differs from that in the other direction [Tversky, 1977].

$$w(A,B) = \frac{w(A \rightarrow_r B) + w(B \rightarrow_{r'} A)}{2d}$$

$$given \quad w(X \rightarrow_r Y) = max_r - \frac{max_r - min_r}{n_r(X)}$$

where $\rightarrow_r$ is a relation of type r, $\rightarrow_{r'}$ is its inverse, d is the depth of the deeper of the two nodes, max_r and min_r are the maximum and minimum weights possible for a relation of type r respectively, and $n_r(X)$ is the number of relations of type r leaving node X. □

The synonym relation gets a weight of zero, while the nine internode relation types have preliminary weight ranges as follows: hypernymy, hyponymy, holonymy, and meronymy all have weights ranging from 1 to 2. Antonymy arcs all get the value 2.5 (there is no range).

3.2 Total distance minimization

We utilize semantic distance between network nodes, captured by the weights on the edges along the shortest path connecting the nodes, as a measure of relatedness between the topics represented by the nodes. The shorter the distance, the greater the relatedness. For disambiguation the hypothesis is that, given a set of terms occurring near each other in the text, each of which might have multiple meanings, by picking the senses that minimize distance we select the correct senses.

Overall distance minimization works as follows. For a given set of terms $T = \{t_1, t_2, ..., t_n\}$, each with possibly more than one candidate sense, each combination of n senses across the terms is tried, with one sense chosen at a time for each term. For example, given three terms t_1, t_2, t_3, with 2, 1, and 3 senses respectively, each of the $6 = 2 \cdot 1 \cdot 3$ combinations of senses is tried. For each combination of n senses, the pairwise distances between each pair of senses is found. The $\frac{n(n-1)}{2}$ pairwise distances are summed to arrive at an overall value, $H(T)$. The combination of senses which minimizes this sum is the "winning" combination.

Definition 2

For a set of neighboring terms $T = \{t_1, t_2, ..., t_n\}$, let $\mathcal{S}$ be the set of all combinations of term senses, which has cardinality $\prod_{i=1}^{n} |t_i|$, where $|t_i|$ is the number of senses of term i, and let $S \in \mathcal{S}$ be a particular combination of senses $\{s_1, s_2, ..., s_n\}$, where each s_j is a sense of t_j.

The winning combination is the $S \in \mathcal{S}$ which produces the minimal "energy"

$$H_{\min}(T) = \min_{S} \sum distance(x,y) \qquad \forall x, y \in S. \square^2$$

We call this technique *mutual constraint* among terms. There is a special case of mutual constraint where all terms except the one being disambiguated have had their senses determined and "frozen." Thus they have only one sense to work with now. When we are trying to disambiguate a term and work with previous frozen terms only, we speak of using a *frozen past* approach.

We have experimented with pure mutual constraint, pure frozen past, and a combination of the two. In all cases there is a *moving window* of terms currently in focus as we move from the beginning of a document towards its end. In the pure cases there is only a moving window. In the case where there is both mutual constraint and frozen past, a small set of initial text terms is processed with mutual constraint. This sets up a bias in semantic space for the processing of subsequent terms. The later terms are then processed with a moving frozen past window.

Mutual constraint is more appealing conceptually than frozen past but is exponential in the number of combinations of term senses that need to be tried. Frozen past avoids this combinatoric explosion by reducing the problem to essentially linear-time processing, since there are only as many "combinations" to try as there are senses of the single term being disambiguated.

Which term(s) gets its winning sense assigned varies depending on the type of window used. When working with a frozen past window of size n, only the $(n+1)$st term is assigned its sense. Each of the n window terms has already had its sense frozen. When working with a moving mutual constraint window, just the middle term is assigned its sense. Record is kept of the winning sense, but when that term plays a role other than "middle term," its senses are allowed to fully vary. This gives a middle term full benefit of both previous and subsequent context. All senses of surrounding terms are considered, not just their winning senses. For initial (as opposed to moving) mutual constraint windows, all of the terms in the window are assigned their senses at the same time.

4 Experiments

We have performed a number of disambiguation experiments with the *Time* collection. One series of experiments varied window size and type, and a second series varied network weighting schemes. Before discussing our experimental results, we need to cover the subject of measuring performance during disambiguation.

4.1 Performance evaluation

How do we measure success in disambiguation? We need to know what the "right" answer is for each term being disambiguated. This knowledge is provided by manual analysis and disambiguation of the terms. Because this is tedious and problematic work, we originally only hand-disambiguated the first five *Time* documents.

During that process it became evident that there are a number of situations that can arise when considering the input to the disambiguator. Seven situations can be distinguished:

1. There are multiple "good" senses — more than one sense of the input term is applicable in the context in which the term appears.
2. There is exactly one good sense.
3. There are no applicable senses. This has five variations:
   - **3a.** The item is not actually a noun here (e.g. "prime" in "prime minister")
   - **3b.** The item is a noun, but not the one the program sees (e.g. "cent" from "per cent")

[2] $distance(x,y) = \frac{distance(x \rightarrow y) + distance(y \rightarrow x)}{2} = distance(y,x) = distance(x,x) = 0.$

3c. The item was found as is, instead of after being stemmed ("acres" meaning "estate, demesne" instead of the plural of "acre")

3d. The item is really a proper noun ("time" as in *Time Magazine*)

3e. The item is used in a sense not found in WordNet ("time" as in "at that time")

We take these situations into account in deriving our measure of success or failure in disambiguation. Although the disambiguator in general works with every word that it is presented with, we focus only on those terms which have at least one good sense. In addition, we distinguish between "trivial" success and "nontrivial" success — words with at least one good sense but with no bad senses are trivial to disambiguate, since any choice is a success. Only when at least one sense is good and at least one is bad can we consider picking a correct sense a success worth rewarding. Thus we focus on nontrivial terms — those which are true tests of disambiguation prowess.

One obvious way of evaluating success is to find the percentage of terms correctly disambiguated (out of the nontrivial terms). We will use this "hit-or-miss" measure as a secondary indicator. Since it does not reflect the difficulty present for individual terms, we have chosen to focus on another measure that takes this difficulty into account. This is the "hit score" — the ratio of "actual hit points" to "maximum hit points." Hit points are awarded as follows.

Definition 3

For each term let s be the number of senses and let g be the number of good senses (in context). The *hit points* for a hit are $s/g - 1$. Misses get zero points. □

The actual hit points for individual terms are summed, and this sum is divided by the sum of the maximum number of hit points possible, derived by treating all nontrivial terms as having been disambiguated correctly and their hit points awarded accordingly. Formally, hit score over n terms equals

$$\frac{\sum_{i=1}^{n} hitpoints_i \quad where \quad term_i \quad is \quad a \quad hit}{\sum_{i=1}^{n} hitpoints_i}.$$

Hit scores range from 0 to 1.

After manual disambiguation, the first five *Time* documents served as a standard against which to measure the performance of the semantic distance software. During manual disambiguation, the several situations that can arise for a term which were outlined above were taken into account when classifying the terms. The large majority of the terms had at least one good sense. Some basic quantities for the five documents are:

```
1175 terms remaining after stopword removal
 544 of those are nouns and in WordNet

122 type 1 terms (multiple good senses)
364 type 2 terms (one good sense)
 58 type 3 terms (no good senses) as follows:
 18 type 3a (not really a noun)
  4 type 3b (wrong noun)
  6 type 3c (unstemmed, taken as is)
  7 type 3d (proper noun)
 23 type 3e (sense not in WordNet)

486 possible hits (at least 1 good sense)
167 poss. trivial hits (good but no bad senses)
319 poss. nontrivial hits (good and bad senses)

749.9 maximum hit points
```

As a baseline for comparison, senses were chosen randomly. This "chance" performance yielded expected values as follows:

```
nontrivial hits:           124.6
% correct of nontrivial:     .391  (124.6 / 319)
hit points:                194.4
hit score:                   .259  (194.4 / 749.9)
```

The standard deviation of the distribution of hit scores obtained from multiple runs of the "chance" software is approximately 0.04. In other words, taking a ± two standard deviation range, the "chance" software will give a hit score in the range 0.259 ± 0.08 with high probability. Thus if an alternative method scores well above 0.259 + 0.08 = 0.339, it is performing statistically significantly above the "chance" method.

These chance values were derived analytically and then verified empirically. For 20 empirical random sense selection runs the average hit score was between .25 and .26.

As "chance" provides a lower bound to compare our results against, human performance on the same tasks provides an upper bound. We had human subjects pick their estimate of the correct sense for each noun in WordNet for the first five *Time* documents. Two sets of printouts were distributed, each with the nouns in documents 1-5. Each noun's synset was given, along with its hypernym's synset and a gloss if available. The subjects were thus given roughly the same sparse information that the software was getting. Although the humans could bring to bear their world knowledge and linguistic knowledge, which should give them a large advantage, they were also handicapped by only receiving very local network data (node and parent only). In contrast, the software has the entire network at its disposal, albeit for its limited approach of looking at semantic distance. Also, the wording within synsets is quite terse and might not be highly suggestive of the actual sense intended. Thus, humans might find the information difficult to glean meaning from.

Averaging over the two tests, the average percent correct was .782 and the average hit score .706. Of course this sample is too small for statistical robustness. Nevertheless, it succeeds in giving us an idea of how people do under these same conditions.

For all of the experiments with the software, results are given for hit score unless otherwise stated. Generally hit score is more informative than simple percent correct.

4.2 Window variation

In the first series of experiments, window type and size were varied. First we tried frozen past windows of increasing size, from 1 to 100. These moving window results are given in Figures 1 and 2.

As one can see, success climbs to a point and then tapers off. This may be an effect of local discourse context size. As can be seen, the semantic distance approach produces results which are highly statistically significant. This is all the more significant, given the number of filters that the input text has gone through, and the amount of "noise"

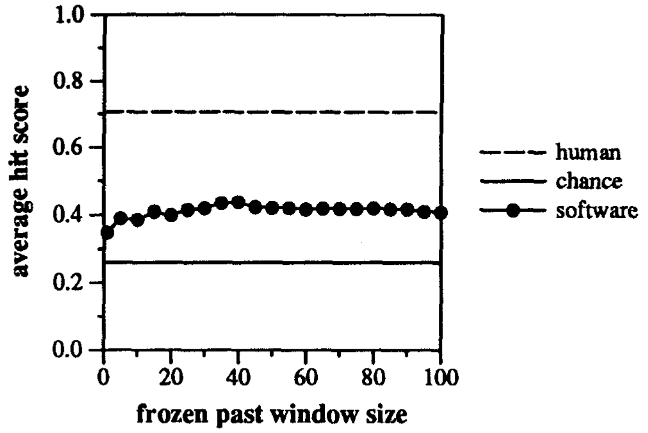

Figure 1: Comparison of hit scores for chance, semantic distance software, and human subjects for *Time* documents 1-5.

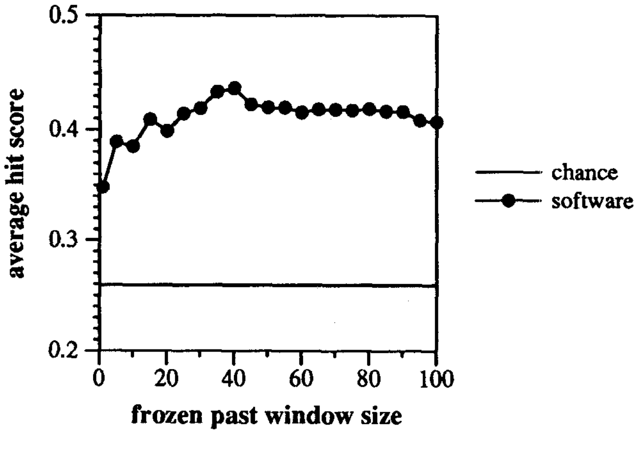

Figure 2: The same data as in Figure 1 but with the vertical scale restricted.

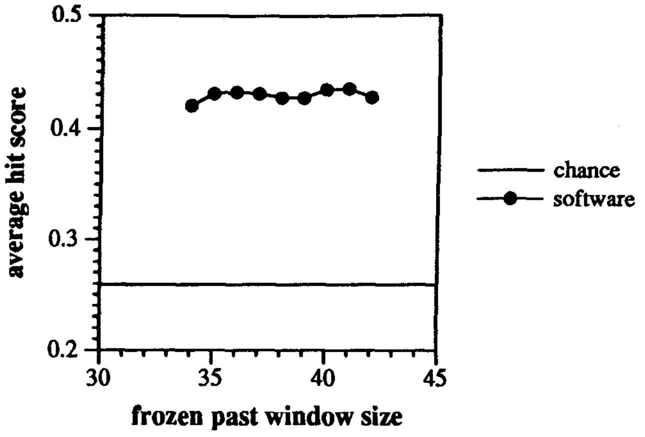

Figure 3: Pinning down the optimal moving frozen past window; no mutual constraint window.

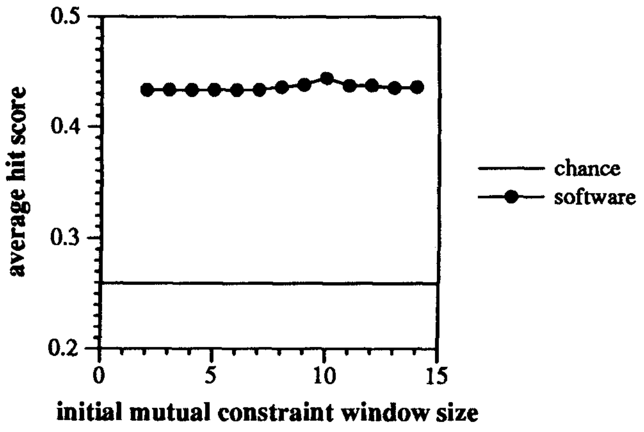

Figure 4: Initial mutual constraint window with frozen past window = 41.

in the remaining "signal." Also, since the semantic net resources used are relatively rudimentary compared to what they might be potentially, even greater success is possible.

The next experiments attempted to pin down the peak performance seen near window sizes of 35 and 40. The best result was with a frozen past window size of 41, .437525. See Figure 3.

Next, fixing the frozen past window size at 41, we tried augmenting this with an initial mutual constraint window. We were unable to proceed past an initial window size of 14 because the runs were taking exponentially longer. The best results were with an initial mutual constraint window of size 10, given the frozen past window of size 41 for all subsequent terms (henceforth "(10,41)"). The hit score was .446771. All terms within the initial mutual constraint window had their sense selections fixed simultaneously once the objective function had determined the winning combination of senses. See Figure 4.

We next tried a moving mutual constraint window. By the time we had made the window size 9, the runs were taking about three hours, so we stopped there. The results were tantalizing, as the hit scores were just getting above .4 at the point where we were forced to halt. See Figure 5.

Note that these runs take longer per window size than the ones where only the initial terms are processed using mutual constraint. The moving mutual constraint window

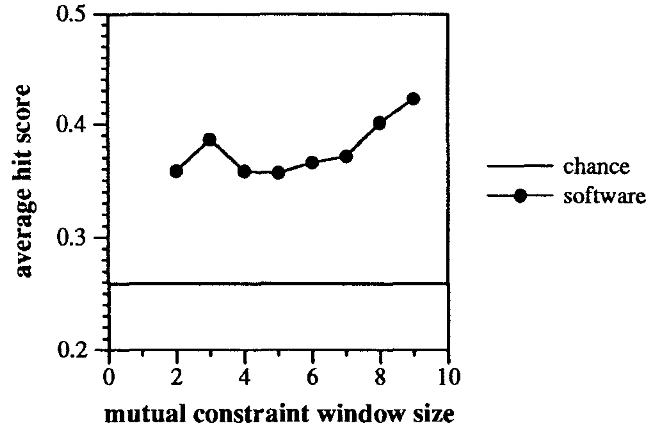

Figure 5: Moving mutual constraint window, no frozen past window.

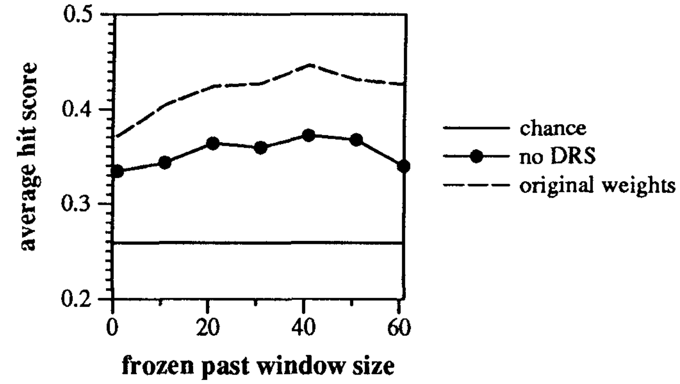

Figure 6: No depth-relative scaling (DRS), initial mutual constraint window = 10.

runs apply mutual constraint throughout an entire document, not just the first set of terms. Thus, instead of a one-time cost incurred for an exponential number of pairwise comparisons, the cost here is incurred roughly n times, where n is the number of terms in the document.

4.3 Weight variation

In the second series of experiments, we looked at the effects of varying the network weights in controlled ways. In each case we varied one parameter at a time.

We varied the network edge weights to see the effect on disambiguation performance. First, we turned off depth-relative scaling. Interestingly, as shown in Figure 6, the hit scores over a range of frozen past window sizes with an initial mutual constraint window fixed at size 10 were low. This indicates that depth-relative scaling makes an important difference.

Next, we tried making all weights equal. This gets rid of weight ranges and differences between relation types. The results indicate that this makes little difference in the outcome. Thus, the particular weights used may not make that much difference. Of course the original ranges were not that different from each other, nor that wide. The best results, still with an initial window of size 10 and a moving frozen past window size of 41, were lower than with the original weighting scheme. So, perhaps weight distinctions and weight ranges help fine tune the performance. See Figure 7.

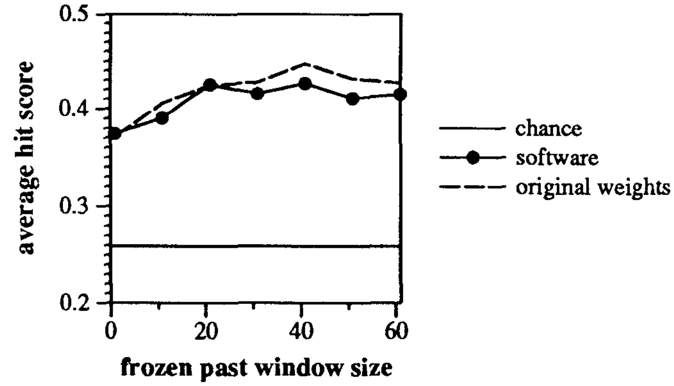

Figure 7: Uniform weights, initial mutual constraint window = 10.

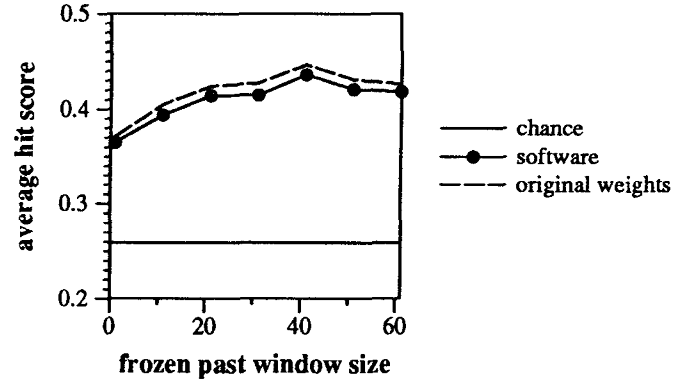

Figure 8: Privileged antonymy, initial mutual constraint = 10, frozen past.

Next we gave antonyms privileged status. Instead of having the highest weight of 2.5, we gave them the lowest weight, 0.5. This made negligible difference. See Figure 8.

In the next experiment, we again saw a noticeable change in behavior. Here we made the network essentially hierarchical, deemphasizing the part/whole and antonymy relationships and leaving the *is-a* relations dominant. The results are plotted in Figure 9.

We see a flat level of success across varying window sizes, and a mediocre one at that, the scores hovering in the range between .35 and .36. Thus, although using hierarchical relationships gives us some power, we need to exploit the richness of additional relationships between topics to increase our ability to disambiguate. This is evidence for the power of *mixed-link networks*, containing both hierarchical and nonhierarchical relations [Rada *et al.*, 1989; Kim and Kim, 1990].

Removing type-specific fanout made little difference, but again the scores were slightly lower without it. See Figure 10.

Finally, we tried the inverse of emphasizing the hierarchical relations. Here, we gave the hierarchical relations (hypernymy and hyponymy) large weights, ranging between 5 and 10 instead of between 1 and 2. This also made little difference. See Figure 11.

Other variations both in windowing and in weight variation are certainly conceivable. Nevertheless, we have performed a number of preliminary experiments which have revealed useful insights. Specifically, it seems that depth-

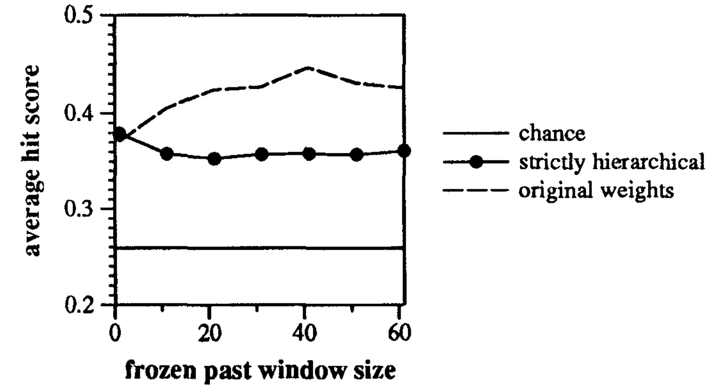

Figure 9: Strictly hierarchical weighting, where part/whole and antonymy links are deemphasized; initial mutual constraint window = 10, frozen past.

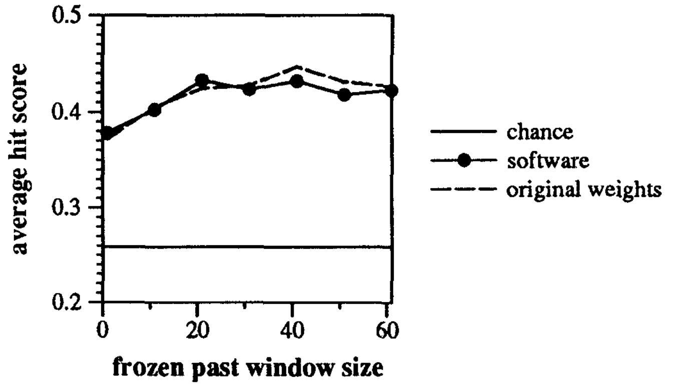

Figure 10: No type-specific fanout, initial mutual constraint = 10, frozen past.

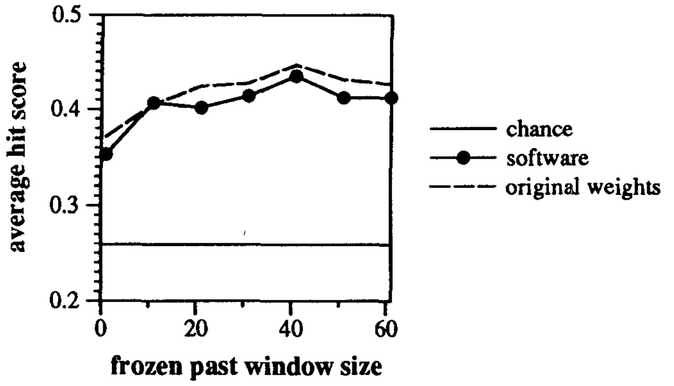

Figure 11: Highly-weighted hierarchical relations, initial mutual constraint = 10, frozen past.

relative scaling is important. In addition, it would appear that both hierarchical and nonhierarchical relations make contributions.

We also looked at another set of five documents to see if the patterns would hold up. Only a few variations were tried, rather than the more comprehensive testing that was performed for the first five documents. Although the scores were lower, the overall trends of increase and plateauing were still recognizable. Although the number of terms in the second batch of documents was only slightly lower (272 vs. 319), the expected hit score was much higher (c. .294). This was caused by a much higher proportion in the second batch of "high probability" terms, where there were very few senses for multi-sense terms. In the first batch there had been a larger number of difficult terms, with many senses. It is not clear whether this difference made the software less effective for the second batch. Nevertheless, even with the results for the second batch included, the software has performed well.

	Documents 1-5		Documents 1-10	
	% correct	hit score	% correct	hit score
chance	.398	.259	.393	.274
(10,41)	.558	.447	.531	.418
human	.782	.706	—	—

Table 1. Summary of disambiguation results for nontrivial nouns in *Time* documents 1-5 and 1-10. Results for human subjects are only available for documents 1-5.

It is important to note that these figures are for the nontrivial terms only. Although such terms form a significant portion of the documents, focusing on them might give the erroneous impression that more than half the terms in a document would not be disambiguated properly. The truth is, taking the other document terms into account, most nouns in a document will be disambiguated correctly or will not need to be disambiguated in the first place. Therefore, document content will actually be very well represented (at least at the individual term level).

To substantiate this point, for documents 1 to 5 there are 544 nouns in WordNet. Of these, only a small percentage are invalidated because there is no appropriate sense (type 3a-3e situations discussed earlier). 486 out of the 544 terms are valid. Out of these 486 remaining terms, 319 are nontrivial and 167 are trivial. Thus we get the 167 trivial terms correct for free. When we add that number to the 178 nontrivial terms that (10,41) disambiguated correctly, we get 345 hits out of 486. This is 71% correct. Of course we are only looking here at the nouns. If we could bring the other parts of speech to bear, possibly without even invoking sophisticated natural language processing techniques, we might strengthen our grasp of document content considerably.

5 Conclusion

We have seen that applying a semantic network to minimize semantic distance takes us a long way towards the goal of removing extraneous search terms from free-text being indexed for retrieval. Yet the sophistication of natural language processing required is kept minimal.

The methods that we have employed trade off space for time — we use large data structures and keep them in main memory so that the runtime processing effort is kept to a minimum. We do no syntactic analysis nor discourse synthesis, yet we try to exploit some semantics via the network's declarative structure. There is much room for increased sophistication in both the linguistic analysis performed and the richness of information made available in the network. Also, the network weights might be better optimized, and the distance determinations refined to a better approximation of the shortest distance between nodes.

We have seen in this preliminary investigation a number of suggestive indicators. It seems that using the moving frozen past window gives ascending performance to a point and then plateaus. The scores are consistently well above chance. Augmenting this with an initial mutual constraint window may help somewhat. And, the frozen past technique only takes linear time, which is an important consideration. While it is attractive theoretically, the moving mutual constraint window gives good results but becomes untenable with current technology due to exponential increase in processing time. If we were willing to settle for an approximate solution, then we might use a technique such as a genetic algorithm to locate good combinations.

We have seen that the above-chance performance is robust under a number of perturbations. For example, making the network weights uniform, making antonyms privileged, removing type-specific fanout, and devaluing the strictly hierarchical relations do not significantly impair performance. On the other hand, we have seen that depth-relative scaling and restriction to strictly hierarchical relations *do* noticeably impair performance, though it remains well above chance. It is possible that our approach will turn out to handle the full range of indexing; that is, from no keyword assignment at all up to selective keyword assignment from controlled vocabularies.

6 Acknowledgments

We would like to thank Charles Elkan and Paul Kube at UCSD; Ellen Voorhees, Stephen Hanson, and Stephen Judd at Siemens; and George Miller and Christiane Fellbaum at Princeton for discussions and support.

References

[Buckley, 1985] Chris Buckley. Implementation of the SMART information retrieval system. Technical Report 85-686, Computer Science Department, Cornell University, 1985.

[Kim and Kim, 1990] Young Whan Kim and Jin H. Kim. A model of knowledge based information retrieval with hierarchical concept graph. *Journal of Documentation*, 46(2):113–136, June 1990.

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

[Lesk, 1986] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of SIGDOC*, pages 24–26, 1986.

[Miller *et al.*, 1990] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4), 1990.

[Miller, 1990] George A. Miller. Nouns in WordNet: A lexical inheritance system. *International Journal of Lexicography*, 3(4), 1990.

[Rada *et al.*, 1989] Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30, Jan./Feb. 1989.

[Salton and McGill, 1983] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

[Tversky, 1977] A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.

[van Rijsbergen, 1983] C. J. van Rijsbergen. *Information Retrieval.* Butterworths, second edition, 1983.

[Voorhees *et al.*, 1992] 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*, Cambridge, MA, 1992. MIT Press. (in press).

[Wilks *et al.*, 1989] Y. Wilks, D. Fass, C-M. Guo, J. McDonald, T. Plate, and B. Slator. A tractable machine dictionary as a resource for computational semantics. In B. Boguraev and T. Briscoe, editors, *Computational Lexicography for Natural Language Processing.* Longman, 1989.