randa in Computer and Cognitive Science MCCS-93-260, Computing Research Laboratory, New Mexico State University. Las Cruces, New Mexico.

[Hearst, 91] Hearst M., 1991.*Towards Noun Homonym Disambiguation Using Local Context in Large Text Corpora*. Proceedings of the Seventh Annual Conference of the UW Centre for the New OED and Text Research. Waterloo, Ontario.

[Lesk 86] Lesk M., 1986. *Automatic sense disambiguation: how to tell a pine cone from an ice cream cone*. Proceeding of the 1986 SIGDOC Conference, Association for Computing Machinery, New York.

[McRoy 92] McRoy S., 1992.*Using Multiple Knowledge Sources for Word Sense Discrimination*, Computational Linguistics, vol. 18, num. 1.

[Miller 90] Miller G., 1990. *Five papers on WordNet,* Special Issue of International Journal of Lexicography 3(4).

[Miller & Teibel 91] Miller G. and Teibel D., 1991. *A proposal for Lexical Disambiguation*. Proceedings of DARPA Speech and Natural Language Workshop, 395-399, Pacific Grove, California.

[Miller et al. 93] Miller G. Leacock C., Randee T. and Bunker R., 1993. *A Semantic Concordance*. Proceedings of the 3rd DARPA Workshop on Human Language Technology, 303-308, Plainsboro, New Jersey.

[Miller et al. 94] Miller G., Chodorow M., Landes S., Leacock C. and Thomas R., 1994. *Using a Semantic Concordance for Sense Identification*. Proceedings of ARPA Workshop on Human Language Technology, 232-235.

[Rada et al. 89] Rada R., Mili H., Bicknell E. and Blettner M., 1989. *Development an Applicationof a Metric on Semantic Nets.* IEEE Transactions on Systems, Man and Cybernetics, vol. 19, no. 1, 17-30.

[Resnik 93] Resnik P., 1993. *Semantic Classes and Syntactic Ambiguity*. Proceedings of ARPA Workshop on Human Language Technology, 303-308. Plainsboro, New Jersey.

[Resnik 95] Resnik P., 1995. *Disambiguating Noun Groupings with Respect to WordNet Senses.* Proceedings of the Third Workshop on Very Large Corpora, MIT. Cambridge, Massachusetts.

[Ribas 95] Ribas F., 1995. *On learning more Appropriate Selectional Restrictions*. Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics, 112-118, Belfield, Dublin, Ireland.

[Richarson et al. 94] Richarson R., Smeaton A.F. and Murphy J., 1994. *Using WordNet as a Kownledge Base for Measuring Semantic Similarity between Words*. Working Paper CA-1294, School of Computer Applications, Dublin City University. Dublin, Ireland.

[Rigau 94] Rigau G., 1994. *An experiment on Automantic Semantic Tagging of Dictionary Senses.* WorkShop "The Future of Dictionary", Aix-les-Bains, France.

[Rigau & Agirre 95] Rigau G., Agirre E., 1995. *Disambiguating bilingual nominal entries against WordNet*. Seventh European Summer School in Logic, Language and Information, ESSLLI'95, Barcelona.

[Schütze 92] Schütze H., 1992. *Context Space*. Workshop Notes of Fall Session of Statistically-Based Natural Language Processing Techniques, AAAI'92.

[Sussna 93] Sussna M., 1993.*Word Sense Disambiguation for Free-text Indexing Using a Massive Semantic Network.* Proceedings of the Second International Conference on Information and Knowledge Management. Arlington, Virginia USA.

[Voorhees 93] Voorhees E., 1993.*Using WordNet to Disambiguate Word Senses for Text Retrival*. Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Developement in Information Retrieval, pages 171-180, PA.

[Wilks et al. 93] Wilks Y., Fass D., Guo C., McDonald J., Plate T. and Slator B., 1993. *Providing Machine Tractable Dictionary Tools.* Semantics and the Lexicon (Pustejovsky J. ed.), 341-401.

[Yarowsky 92] Yarowsky, D., 1992.*Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora.* Proceedings of the 15th International Conference on Computational Linguistics (Coling'92). Nantes, France.

[Yarowsky 93] Yarowsky, D., 1993. *One sense per Collocation*. Proceedings of ARPA Workshop on Human Language Technology, 266-271, Plainsboro, New Jersey.

• Compute the upper bound of this method using WordNet.

How correct this methodology can be? That is, words belonging to the same narrow context in SemCor can represent distant correct concepts in WordNet (having other incorrect ones closer).

7 Conclusion

The automatic method for the disambiguation of nouns presented in this paper is ready to use in any general domain, free-running text, given part of speech tags. It does not need any training and uses word sense tags from WordNet, a widely used lexical data base. The algorithm is theoretically motivated, and offers a general measure of the semantic relatedness for any number of nouns.

Conceptual Density has been used for other tasks apart from the disambiguation of free-running test. Its application for automatic spelling correction is outlined in [Agirre et al. 94]. It was also used on Computational Lexicography, enriching dictionary senses with semantic tags extracted from WordNet [Rigau 94], or linking bilingual dictionaries to WordNet [Rigau and Agirre 95].

In the experiments, the algorithm disambiguated four texts (more than 9,000 words long) of SemCor, a subset of the Brown corpus. The results were obtained automatically by comparing the tags in SemCor with those computed by the algorithm. This allows the comparison with other disambiguation methods. Two other methods, [Sussna 93] and [Yarowsky 92], were also tried on the same texts, showing that our algorithm performs better.

The results are promising, considering the difficulty of the task (free running text, large number of senses per word in WordNet), and the lack of any discourse structure of the texts. Two kinds of results can be obtained: the specific sense or a coarser, file level, tag.

Acknowledgements

We wish to thank all the staff of the CRL and specially Jim Cowie, Joe Guthtrie, Louise Guthrie and David Farwell. We would also like to thank Ander Murua, who provided mathematical assistance, Xabier Arregi, Jose Mari Arriola, Xabier Artola, Arantxa Diaz de Ilarraza, Kepa Sarasola, and Aitor Soroa from the Computer Science Department of EHU and Francesc Ribas, Horacio Rodríguez and Alicia Ageno from the Computer Science Department of UPC.

BIBLIOGRAPHY

[Agirre et al. 94] Agirre E., Arregi X., Diaz de Ilarraza A. and Sarasola K., 1994.*Conceptual Distance and Automatic Spelling Correction.* Workshop on Speech recognition and handwriting. Leeds, England.

[Church et al. 91] Church K., Gale W., Hanks P. and Hindle D., 1991.*Using Statistics in Lexical Analysis*. Lexical Acquisition: Exploiting On-line Resources to Build a Lexicon. Zernik U. Ed. Lawrence Erlbaum Associates, publishers. Hillsdale, New Jersey.

[Cowie et al. 92] Cowie J., Guthrie J., Guthrie L., 1992. *Lexical Disambiguation using Simulated annealing.* Proceedings of DARPA WorkShop on Speech and Natural Language, 238-242, New York.

[Francis & Kucera 67] Francis S. and Kucera H., 1967.*Computational analisys of present-day American English*, Providence, RI: Brown University Press.

[Gale et al. 93] Gale W., Church K. and Yarowsky D., 1993. *A Method for Disambiguating Word Sense sin a Large Corpus*. Computers and the Humanities, n. 26.

[Guthrie et al. 93] Guthrie L., Guthrie J. and Cowie J., 1993. *Resolving Lexical Ambiguity,*. Memo-

%		Cover.	Prec.
C.Density	File	100.0	70.1
	Sense		60.1
Sussna	File	100.0	64.5
	Sense		52.3

Table 4: comparison with [Sussna 93]

6 Future Work

Initially, we would like to carry out a study on whether there is or is not a correlation between correct and erroneous sense assignations and the degree of Conceptual Density computed by formula 3. If this was the case, the error rate could be further decreased by setting a certain threshold for Conceptual Density values for winning senses.

There are other factors that could increase the performance of our algorithm:

- Work on coherent chunks of text.

Unfortunately any information about discourse structure is absent in SemCor, apart from sentence endings. If coherent pieces of discourse were taken as input, both performance and efficiency of the algorithm might improve. The performance would gain from the fact that sentences from unrelated topics would not be considered in the disambiguation window. We think that efficiency could also be improved if the algorithm worked on entire coherent chunks instead of one word at a time.

- Extend and improve the semantic data.

WordNet lacks cross-categorial semantic relations, which could be very useful for extending the notion of Conceptual Density of nouns to Conceptual Density of words. Apart from extending disambiguation to verbs, adjectives and adverbs, cross-categorial relations would allow the algorithm better capture the relations among senses and provide firmer grounds for disambiguating.

If Conceptual Density takes into account global relations among words, it may be advantageous to combine it with other sources of knowledge (both corpus-based or MRD-based) such as syntactic cues, word frequencies, collocations, selectional restrictions [Yarowsky 93], [Ribas 95], and so on. (c.f. [McRoy 92]). For instance, [Richardson et al. 94] defines conceptual similarity between two senses based on WordNet and informational measures taken from corpora, but does not give any evaluation of their method.

- Tune the sense distinctions to the level best suited for the application.

On the one hand, the sense distinctions made by WordNet 1.4 are not always satisfactory and, obviously, WordNet 1.4 is not a complete lexical Database. For instance, the three senses of abobe and the lack of connections among them, which are fixed up in WordNet 1.5. On the other hand, our algorithm is not designed to work on the file level, e.g. if the sense level is unable to distinguish among two senses, the file level also fails, even if both senses were from the same file. If the senses were collapsed at the file level, the coverage and precision of the algorithm at the file level might be better.

12.*Initial mutual constraint size is 10 and window size is 41. Meronymic links are also considered. All the links have the same weigth.*

The raw results presented here seem to be poor when compared to those shown in [Hearst 91], [Gale et al. 93] and [Yarowsky 92]. We think that several factors make the comparison difficult. Most of those works focus on a selected set of a few words, generally with a couple of senses of very different meaning (coarse-grained distinctions), and for which their algorithm could gather enough evidence. On the contrary, we tested our method with **all** the nouns in a subset of an unrestricted public domain corpus (more than 9.000 words), making fine-grained distinctions among all the senses in WordNet.

[Guthrie et al. 93] tested their method in similar conditions to ours, but without performing an extensive and automatic testing. The results reported there seem to be lower than those shown here. In an experiment with 50 sample sentences from LDOCE, 47% of the words were correctly disambiguated to the sense level, and 72% to the homograph level (our file level would stand between their homograph and sense levels).

An approach that uses hierarchical knowledge is that of [Resnik 95], which additionally uses the information content of each concept gathered from corpora. Unfortunately he applies his method on a different task, that of disambiguating sets of related nouns. The evaluation is done on a set of related nouns from Roget's Thesaurus tagged by hand. The fact that some senses were discarded because the human judged them not reliable makes comparison even more difficult.

In order to compare our approach we decided to implement [Yarowsky 92] and [Sussna 93], and test them on our texts. For [Yarowsky 92] we had to adapt it to work with WordNet. His method relies on cooccurrence data gathered on Roget's Thesaurus semantic categories. Instead, on our experiment we use saliency values[9] based on the lexicographic file tags in SemCor (see Figure 4). The results for a window size of 50 are those shown in table 3[10]. The precision attained by our algorithm is higher. To compare figures better consider the results in table 4, were the coverage of our algorithm was easily extended using the version presented below, increasing recall to 70.1%.

%	Cover.	Prec.	Recall
C.Density	86.2	71.2	61.4
Yarowsky	100.0	64.0	64.0

Table 3: comparison with [Yarowsky 92]

From the methods based on Conceptual Distance, [Sussna 93] is the most similar to ours. Sussna disambiguates several documents from a public corpus using WordNet. The test set was tagged by hand, allowing more than one correct senses for a single word. The method he uses has to overcome a combinatorial explosion[11] controlling the size of the window and “freezing” the senses for all the nouns preceding the noun to be disambiguated. In order to freeze the winning sense Sussna's algorithm is forced to make a unique choice. When Conceptual Distance is not able to choose a single sense, he has to choose one at random.

Conceptual Density overcomes the combinatorial explosion extending the notion of conceptual distance from a pair of words to n words, and therefore can yield more than one correct sense for a word. For comparison, we altered our algorithm to also make random choices when unable to choose a single sense. We applied the algorithm Sussna considers best, discarding the factors that do not affect performance significantly[12], and obtain the results in table 4.

9.*We tried both mutual information and association ratio, and the later performed better.*
10.*The results of our algorithm are those for window size 30, file matches and overall.*
11.*In our replication of his experiment the mutual constraint for the first 10 nouns (the optimal window size according to his experiments) of file br-r05 had to deal with more than 200.000 synset pairs.*

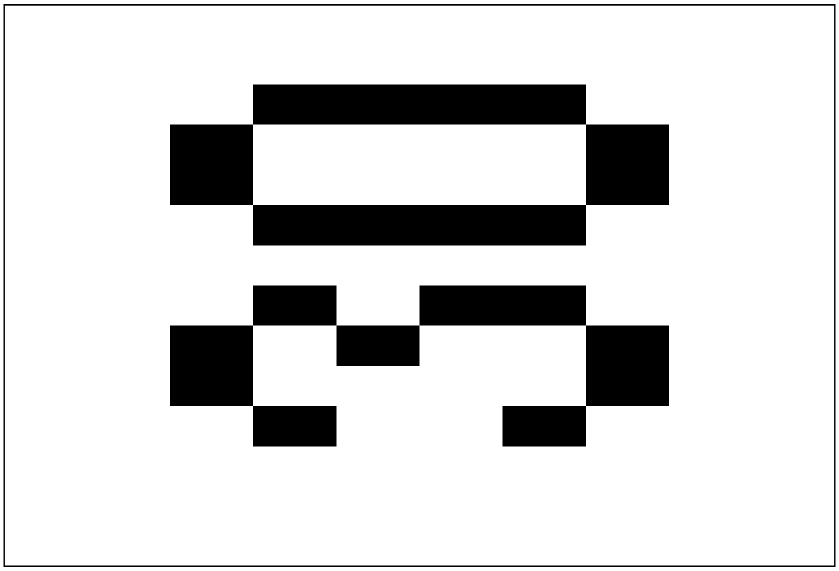

Figure 11: complete disambiguation and partial disambiguation

5.2.6 file vs. sense

WordNet synsets can be grouped by the lexicographic files they are coming from (e.g. `ACT`, `ANIMAL`, `FOOD`, etc.) Both file matches and synset matches are interesting to count. While the sense level gives a fine grained measure of the algorithm, the file level gives an indication of the performance if we were interested in a less precise level of disambiguation. The granularity of the sense distinctions made in [Hearst, 91], [Gale et al. 93] and [Yarowsky 92], also called homographs in [Guthrie et al. 93], can be compared to that of the file level in WordNet.

For instance, in [Yarowsky 92] two homographs of the noun `bass` are considered, one characterised as `MUSIC` and the other as `ANIMAL`, `INSECT`. In WordNet, the 6 senses of `bass` related to music appear in the following files: `ARTIFACT`, `ATTRIBUTE`, `COMMUNICATION` and `PERSON`. The 3 senses related to animals appear in the files `ANIMAL` and `FOOD`. This means that while the homograph level in [Yarowsky 92] distinguishes two sets of senses, the file level in WordNet distinguishes six sets of senses, still finer in granularity.

The following figure shows that, as expected, file-level matches attain better performance (71.2% overall and 53.9% for polysemic nouns) than sense-level matches.

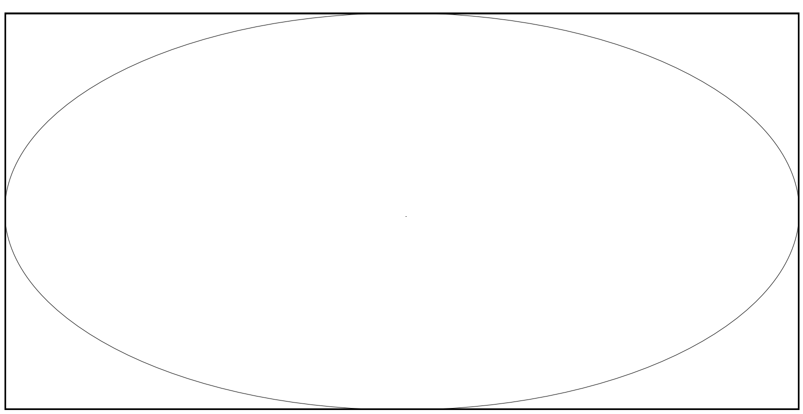

Figure 12: sense level v. file level

5.3 Comparison with other works

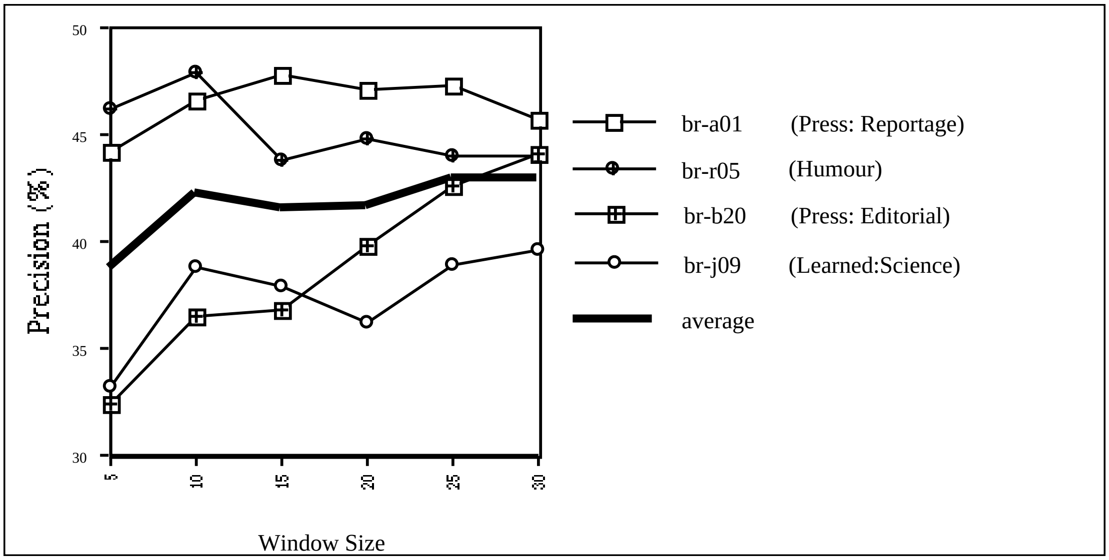

Figure 10: context size and different files

Each text is structured as a list of sentences, lacking any indication of headings, sections, paragraph endings, text changes, etc. This means that the program gathers the context without knowing whether the nouns actually occur in coherent pieces of text. This could account for the fact that in br-r05, composed mainly by short pieces of dialogues, the best results are for window size 10, the average size of pieces of this dialogue. Longer windows will include other pieces of unrelated dialogues that could cause the disambiguation process to go astray.

In addition, SemCor files can be composed of different pieces of unrelated texts without explicit indication. For instance, two of our test files (br-a01 and br-b20) are collections of short journalistic texts. This could explain why the performance of br-a01 decreases for windows of 30 nouns. For most nouns the context window would include nouns from other articles.

The polysemy level could also affect the performance, but in our texts less polysemy does not correlate with better performance. Nevertheless the actual nature of each text is certainly an important factor, difficult to measure, which could account for the different behaviour on its own. For instance, the poor performance on text br-j09 could be explained by its technical nature. Further analysis of the errors, contexts and relations found among the words would be needed to be more conclusive.

In order to give an overall view of the performance, we consider the average behaviour for formulating our conclusions leaving aside these considerations.

5.2.5 partial disambiguation

The disambiguation algorithm has an intermediate outcome between completely disambiguating a word or failing to do so. In some cases the algorithm just manages to discard some senses of the word, but can not choose a single sense. The automatic evaluation program does not take these cases into account, treating them as failures to disambiguate. While the number of words that are not disambiguated decreases for the benefit of completely disambiguated as the window size is bigger, the number of partially disambiguated words stays the same (see Figure 11).

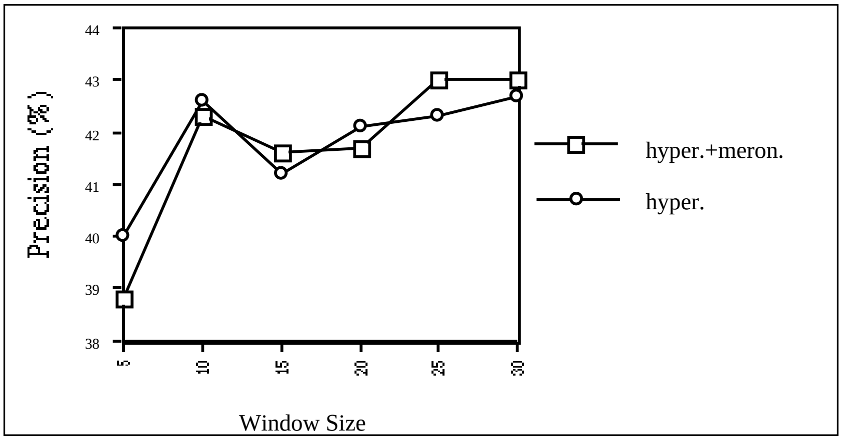

Figure 8: meronymy and hyperonymy

5.2.3 global nhyp is as good as local nhyp.

There was an aspect of the density formula which we could not decide analytically and which we wanted to check experimentally. It refers to the way *nhyp* is calculated (c.f. formula 2). If *nhyp* is computed using formula 2, we call it *local nhyp*, because it has to be computed for every concept of WordNet. Rather than using this *local nhyp*, it would be more desirable, specially for efficiency, if only one global *nhyp* were used for all the concepts. This *global nhyp* can be computed using the whole noun hierarchy. Depending on which *nhyp* is chosen will either be the real number of descendant senses for c (for local *nhyp*) or and estimation based on the global *nhyp*.

To decide whether using local *nhyp* or global *nhyp* affects the performance, we ran parallel experiments using both. The results (see Figure 9) show that there is only a slight difference between them. Therefore, *global nhyp* was used in the experiments.

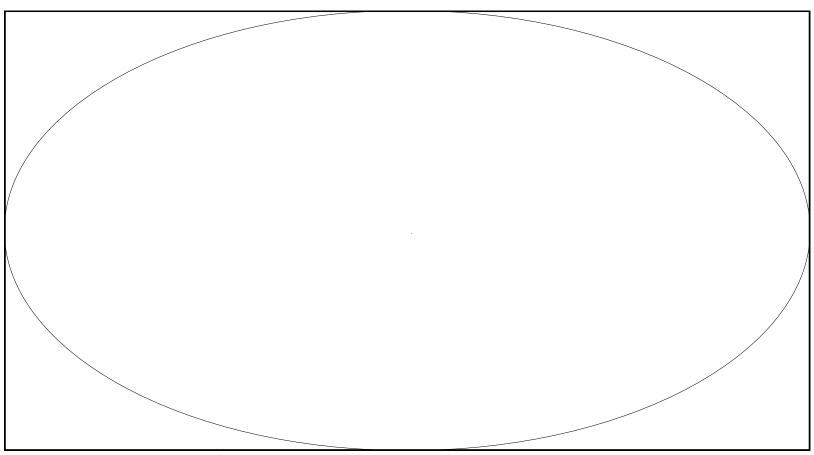

Figure 9: local *nhyp* vs. global *nhyp*

5.2.4 context size: different behaviour for each text

Deciding what context size was better for disambiguating using Conceptual Density is an important issue. One could assume that the more context there is, the better would be the disambiguation results. Our experiments show that each file from SemCor has a different behaviour (see Figure 10). While br-b20 shows clear improvement for bigger window sizes, br-r05 gets a local maximum at a size window of 10 nouns, etc.

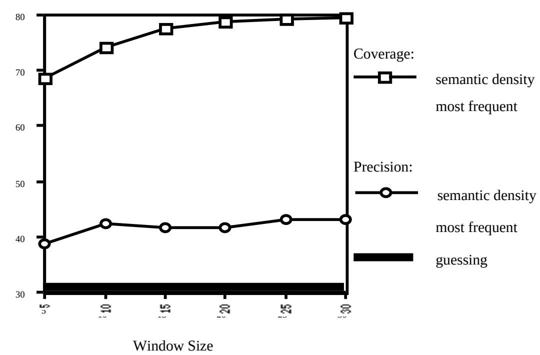

Figure 7: precision and coverage

The figure also shows the guessing baseline, given by selecting senses at random. First, it was calculated analytically using the polysemy counts for the files, which gave 30% of precision. This result was checked experimentally running an algorithm ten times over the files, which confirmed the previous result.

We also compare the performance of our algorithm with that of the most frequent heuristic. The frequency counts for each sense were collected using the rest of SemCor, and then apply the results to the four texts. While the precision is similar to that of our algorithm, the coverage is 8% worse.

All the data for the best window size can be seen in table 2.

%	w=30	Cover.	Prec.	Recall
overall	File	86.2	71.2	61.4
	Sense		64.5	55.5
polysemic	File	79.6	53.9	42.8
	Sense		43	34.2

Table 2: overall data for the best window size

The precision and coverage shown in all preceding plots were relative to the polysemous nouns only. If we also include monosemic nouns precision raises from 43% to 64.5%, and the coverage increases from 79.6% to 86.2%.

5.2.2 meronymy does not improve performance as expected.

One parameter controls whether meronymic relations, in addition to the hypo/hypernymy relation, are taken into account or not. In principle the more relations are taken in account, the better density would capture semantic relatedness and, therefore, the better the expected results. The experiments (see Figure 8) showed that there is not much difference; adding meronymic information does not improve precision, and raises coverage only 3% (approximately). Nevertheless, in the results reported, meronymy and hypernymy were used.

```
<wd>operation</wd><sn>[noun.state.0]</sn><tag>NN</tag>
```

```
<wd>Police_Department</wd><sn>[noun.group.0]</sn><tag>NN</tag>
```

```
<wd>prison_farms</wd><mwd>prison_farm</mwd><msn>[noun.arti-
fact.0]</msn>
    <tag>NN</tag>
```

```
</s>
```

Figure 5: Semcor format

After erasing the irrelevant information we get the following words[6]:

```
jury administration operation Police_Department
prison_farm
```

Figure 6: input words

The algorithm then produces a file with sense tags that can be compared automatically with the original file (see figure 5). An automatic program counts sense level matches and file level matches (see Section 5.2.6) for the three classes of results: complete disambiguation, partial disambiguation and failure to disambiguate. For the results shown in Section 5.2, partial disambiguation was considered as failure to disambiguate.

5.2 Results and evaluation

One of the goals of the experiments was to decide among different variants of the Conceptual Density formula. Results are given averaging the results of the four files. Partial disambiguation is treated as failure to disambiguate. Precision[7] is given in terms of polysemous nouns only. Plots are drawn against the size of the context[8] that was taken into account when disambiguating.

5.2.1 evaluation of the results

Figure 7 shows that, overall, coverage of polysemous nouns increases significantly with the window size, without losing precision. Coverage tends to stabilised near 80%, getting little improvement for window sizes bigger than 20.

6.*Note that in the input texts we already have the knowledge that police department and prison farm are compound nouns, and that the lemma of prison farms is prison farm.*
7.*Precision is defined as the ratio between correctly disambiguated senses and total number of answered senses. Coverage is given by the ratio between total number of answered senses and total number of senses. Recall is defined as the ratio between correctly disambiguated senses and the total number of senses.*
8.*context size is given in terms of nouns.*

procedure is repeated. At this point we start afresh with all senses of the words in the window.

Back in the example, the algorithm has disambiguated **`operation_3`**, **`police_department_0`**, **`jury_1`** and **`prison_farm_0`** (because this word is monosemous in WordNet), but the word *administration* is still ambiguous. The output of the algorithm , thus, will be that the sense for *operation* in this context, i.e. for this window, is **`operation_3`**. The disambiguation window will move rightwards, and the algorithm will try to disambiguate *Police_Department* taking as context *administration*, *operation*, *prison_farms* and whichever noun is first in the next sentence.

5 The Experiments

5.1 The texts

We selected four texts from SemCor at random: a press report (br-a01), an editorial (br-b20), a scientific text (br-j09) and a humorous article (br-r05). Table 1 shows some statistics for each text

text	words	nouns	nouns in WN	monosemous
br-a01	2079	564	464	149 (32%)
br-b20	2153	453	377	128 (34%)
br-j09	2495	620	586	205 (34%)
br-r05	2407	457	431	120 (27%)
total	9134	2094	1858	602 (32%)

Table 1

An average of 11% of all the nouns in these four texts were not found in WordNet. According to this data, the percentage of monosemous nouns in these texts is bigger (32% average) than the one calculated for the open-class words from the whole SemCor (27.2% according to [Miller et al. 94]). [Sussna 93] presents a similar degree of polysemy for nouns (34% of monosemous nouns), but in a different text collection.

These texts play both the role of input files (without semantic tags) and (tagged) test files. When they are treated as input files, we throw away all non-noun words, only leaving the lemmas of the nouns present in WordNet. The program does not deal with syntactic ambiguity, as the part of speech information is in the input files. Multiple word entries are also available in the input files, as long as they are present in WordNet. Proper nouns have a similar treatment: we only consider those that can be found in WordNet. Figure 5 shows the way the algorithm would input the example sentence in figure 3 after stripping non-noun words:

```
<s>

<wd>jury</wd><sn>[noun.group.0]</sn><tag>NN</tag>

<wd>administration</wd><sn>[noun.act.0]</sn><tag>NN</tag>
```

administration_1, governance, government, establishment, brass...

jury_2

=> <u>body</u>

=> people

=> group, grouping

Figure 4: partial lattice for the sample sentence

In this example only hypo/hypernym links are shown. The concepts in WordNet are represented as lists of synonyms. Word senses to be disambiguated are shown in bold. Underlined concepts are those selected with highest Conceptual Density. Monosemic nouns have sense number 0.

2) Once the lattice is completed, the program starts the disambiguation loop until there are no words which remains to be disambiguated. For each loop the program computes the Conceptual Density of every concept in the lattice. For instance <administrative_unit> has underneath 3 senses to be disambiguated and a subhierarchy size of 96 producing a Conceptual Density of 0.256. Meanwhile, <body>, with 2 senses and subhierarchy size of 86, has a Conceptual Density of 0.062.

3) The concept with the highest Conceptual Density (<administrative_unit> in our example) is selected.

4) In this step two actions are performed. Firstly the program follows hyponym chains down from the concepts selected in step 3 (<administrative_unit>) and the senses of the words found in the bottom are selected as the correct senses (**operation_3**, **police_department_0** and **jury_1** are the senses chosen for *<u>operation</u>*, *<u>Police Department</u>* and *<u>jury</u>*). All these nouns are considered to be disambiguated, even if more than one sense of a given word are below the concept selected in step 3. Lastly we build the lattice again as in step 1, but only considering the nouns not yet disambiguated. After that, the loop continues in step 2. In the example, the lattice is built for the senses of *<u>administration</u>* and *<u>prison farms</u>*, but their senses yield non-overlapping lattices (for instance the lattice for **administration_1** would be the same as in figure 4 without **jury_2**), and therefore the loop terminates and we continue in step 5.

5) The program has three possible outcomes for the noun in the middle of the window; one sense has been selected (disambiguated), more than one sense has been selected (partially disambiguated, several senses of the noun are under the same selected concept) or the selection of a sense has been impossible due to the lack of information in the context.

After disambiguating the word in the current window the window moves forward, and the

police_department_0

=> local department, department of local government

=> government department

=> department

jury_1, panel

=> committee, commission

operation_3, function

=> division

=> <u>administrative unit</u>

=> unit

=> organization

=> social group

=> people

=> group, grouping

considering the other words in the window as context.

For each window, the program performs the next disambiguation algorithm:

```
(Step 1)tree := compute_tree(words_in_window)
        loop
(Step 2)tree := compute_conceptual_distance(tree)
(Step 3)concept := select_concept_with_highest_weigth(tree)
        if  concept = null then exitloop
(Step 4)tree := mark_disambiguated_senses(tree,concept)
        endloop
(Step 5)output_disambiguation_result(tree)
```

To illustrate the process, consider the following text extracted from SemCor:

The jury(2) praised the administration(3) and operation(8) of the Atlanta Police_Department(1), the Fulton_Tax_Commissioner_'s_Office, the Bellwood and Alpharetta prison_farms(1), Grady_Hospital and the Fulton_Health_Department.

Figure 3: sample sentence from SemCor

The underlined words are nouns represented in WordNet with the number of senses between brackets (those with a 1 are unambiguous nouns). SemCor links multiword terms using underscores. The noun to be disambiguated in our example is *operation*., and a window size of five will be used.

1) Given the set of nouns constrained by the context window size, our algorithm collects for every noun all its possible senses and hypernyms. All these concepts and connections are placed in a lattice. For each concept in the lattice, the program also stores the set of words that are generalised by the concept.

The following figure shows partially the lattice for the example sentence. Since *Prison_farm* appears in a different hierarchy we do not show it in figure 4:

5.*In fact the algorithm can disambiguate all the nouns in the window in one go, but we consider that the context is most informative for the noun in the center of the window. This and related issues are discussed in Section 6.*

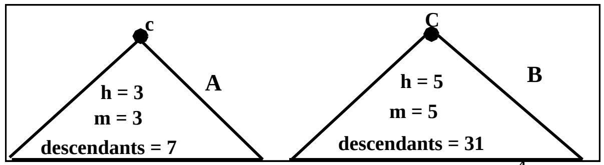

Figure 2: two hierarchies with CD = 1[4].

In order to tune the Conceptual Density formula, we have carried out several experiments adding two parameters, α and β. The α parameter modifies the strength of the exponential *i* because *h* ranges between 1 and 16 (the maximum number of levels in WordNet) while *m* ranges between 1 and the total number of senses in WordNet. Adding a constant β to *nhyp*, we tried to discover the role of the averaged number of hyponyms per concept. Formula 3 shows the resulting formula.

(3)

After a number of runs which were automatically evaluated, the results showed that β does not affect the behaviour of the formula, a strong indication that this formula is not sensitive to constant variations in the number of hyponyms. On the other hand, different values of α affected the performance consistently, yielding the best results in all the experiments where α was 0.20. The formula which was actually used in the experiments, thus, was the following:

(4)

where is the number of descendant senses of the concept *c*.

We have tested the formula in two different ways (see Section 5). The first one involves the manner in which *nhyp* and are calculated. The second arises from the manner in which the hierarchy is constructed: considering only hypo / hypernymy links, or including meronymic links as well.

4 The Disambiguation Algorithm Using Conceptual Density

The algorithm to disambiguate a given noun w in the middle of a window of nouns W roughly proceeds as follows. First, the algorithm represents in a lattice the nouns in the window, its senses and hypernyms (step 1). Then, the program computes the Conceptual Density of each concept in WordNet according to the senses it contains in its subhierarchy (step 2). It selects the concept c with the highest density (step 3) and select the senses below it as the correct senses for the respective words. If a word from W (step 4):

- has a single sense under c, it has already been disambiguated.
- has no a sense, it is still ambiguous
- has more than one sense with highest density, we can eliminate all the other senses of w, but have not yet completely disambiguated w.

It proceeds then to choose the next concept with highest density, and continues to disambiguate words in W. In the end the senses left for w are analysed and the result is output (step 5). This process will be further explained below.

Given a window size, the program moves the window one word at a time from the beginning of the document towards its end, disambiguating the word in the middle of the window[5] and

---

4.From formulas 1 and 2 we have:

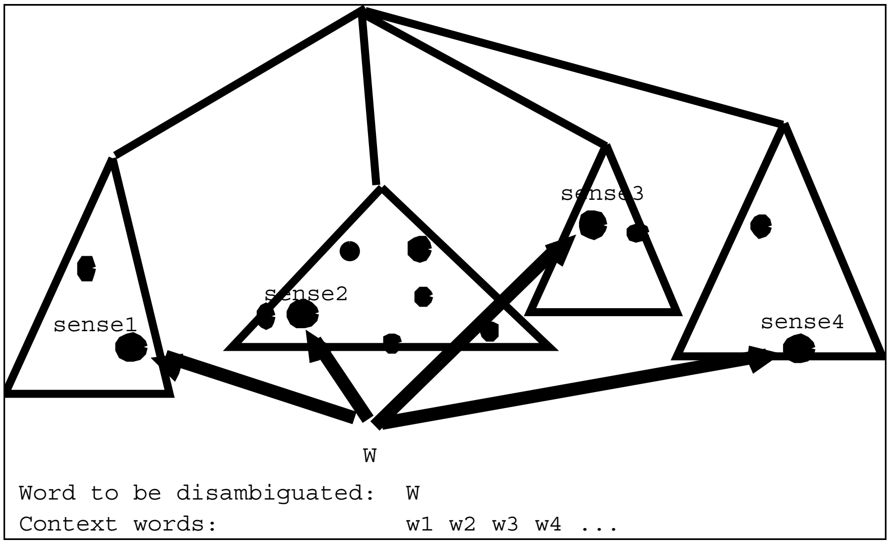

Figure 1: senses of a word in WordNet

The sense of W contained in the subhierarchy with highest Conceptual Density will be chosen as the sense disambiguating W in the given context. In figure 1, sense2 would be chosen.

Given a concept *c*, at the top of a subhierarchy, and given *nhyp* and *h* (mean number of hyponyms per node and height of the subhierarchy, respectively), the Conceptual Density for *c* when its subhierarchy contains a number *m* (marks) of senses of the words to disambiguate is given by the formula below:

(1)

The numerator expresses the expected area for a subhierarchy containing *m* marks (senses of the words to be disambiguated), while the divisor is the actual area, that is, the formula gives the ratio between weighted marks below *c* and the total area of the subhierarchy below *c*. The weight given to the marks tries to express that the height and the number of marks should be proportional.

nhyp is computed for each concept in WordNet in such a way as to satisfy equation 2, which expresses the relation among height, averaged number of hyponyms of each sense and total number of senses in a subhierarchy if it were homogeneous and regular:

(2)

Thus, if we had a concept *c* with a subhierarchy of height 5 and 31 descendants, equation 2 will hold that *nhyp* is 2 for *c*.

Conceptual Density weights the number of senses of the words to be disambiguated so as to make density equal to 1 when the number *m of* senses below *c* is equal to the height of the hierarchy *h*, to make density smaller than 1 if *m* is smaller than *h* and to make density larger than 1 whenever *m* is larger than *h*. The density can be kept constant for different *m*-s provided a certain proportion between the number of marks *m* and the height *h* of the subhierarchy is maintained. Both hierarchies **A** and **B** in figure 2, for instance, have Conceptual Density 1. For the sake of clarity we have assumed uniform hierarchies.

3 Conceptual Density and Word Sense Disambiguation

A measure of the relatedness among concepts can be a valuable predictive knowledge source for several decisions in Natural Language Processing. For example, the relatedness of a certain word-sense to the context allows us to select that sense over the others, and actually disambiguate the word. Relatedness can be measured by a fine-grained conceptual distance [Miller & Teibel 91] among concepts in a hierarchical semantic net such as WordNet. This measure would allow the discovery of the most lexically cohesive set of senses of a given set of words in English.

Several measures of relatedness among words based on cooccurrence in a text have been described; mutual information, t-test, etc. [Church et al. 91], the cosine function in Context Space [Schütze 92], conditional probability [Wilks et al. 93]. [Resnik 93] combines a knowledge based approach involving semantic classes taken from WordNet with cooccurrence data extracted from corpora. Less attention has been paid lately to measures of relatedness based on semantic structured hierarchical nets.

Conceptual distance tries to provide a basis for determining closeness in meaning among words, taking as reference a structured hierarchical net. The conceptual distance between two concepts is defined in [Rada et al. 89] as the length of the shortest path that connects the concepts in a hierarchical semantic net. Besides applying conceptual distance in a medical bibliographic retrieval system and merging several semantic nets, they demonstrate that their measure of conceptual distance is a metric. In a similar approach, [Sussna 93] employs the notion of conceptual distance between network nodes in order to improve precision during document indexing. Following these ideas, [Agirre et al. 94] describes a new conceptual distance formula for automatic spelling correction and [Rigau 94], using this conceptual distance formula, presents a methodology to enrich dictionary senses with semantic tags extracted from WordNet.

The measure of conceptual distance among concepts we are looking for should be sensitive to:

- the length of the shortest path that connects the concepts involved.
- the depth in the hierarchy: concepts in a deeper part of the hierarchy relatively closer than those in a more shallow part.
- the density of concepts in the hierarchy: concepts in a dense part of the hierarchy are relatively closer than those in a more sparse region.

But also:

- the measure should be independent of the number of concepts we are measuring.

We have experimented with several formulas that follow the four criteria presented above. Currently, we are working with a variant of conceptual distance which we call Conceptual Density that compares areas of subhierarchies.

As an example of how Conceptual Density can help to disambiguate a word, in figure 1 the word W has four senses and several context words. Each sense of the words belongs to a subhierachy of WordNet. The dots in the subhierarchies represent the senses of either the word to be disambiguated (W) or the words in the context. Conceptual Density will yield the highest density for the subhierarchy containing more senses of those, relative to the total amount of senses in the subhierarchy.

mantic Concordance or Semcor for short. We also use a public domain lexical knowledge resource, WordNet [Miller 90]. The advantage of this approach is clear; Semcor provides an appropriate environment for testing our procedures in a fully automatic way.

This paper presents a general automatic decision procedure for lexical ambiguity resolution based on a formula of conceptual distance among concepts: Conceptual Density. The procedure needs to know how words are clustered in semantic classes and how semantic classes are hierarchically organised. For this purpose, we have used a broad semantic taxonomy for English, WordNet. We have performed several experiments employing the notion of Conceptual Density among concepts in a structured hierarchical net. Given a piece of text from the Brown Corpus, our system tries to resolve the lexical ambiguity of nouns finding the combination of senses from a set of nouns in context that maximises the total Conceptual Density among senses.

In order to test our algorithms, we have selected at random four texts of SemCor. Our procedure only considers the words in SemCor with a noun part of speech tag. We discarded the nouns not present in WordNet (averaging around 10% of the nouns in all four texts)

Improvement in disambiguation compared with chance is clear and consistent, strongly suggesting that knowledge-based algorithms are competitive with statistically-based approaches, with the advantage of not needing training.

Even if this technique is presented as stand-alone, it is our belief, following the ideas of [McRoy 92] that full-fledged lexical ambiguity resolution should combine several information sources. Conceptual Density might be only one of a number of complementary sources of evidence for evaluating the plausibility of a certain word sense.

In section 2 we present the semantic knowledge sources used by the system. In section 3, we define Conceptual Density. In section 4, we discuss the disambiguation algorithm used in the experiment and in section 5, we explain and evaluate the experiments performed. In section 6, we discuss future directions and, finally, in the last section, we draw some conclusions.

2 WordNet and the Semantic Concordance

Sense is not a well defined concept and often has subtle distinctions in topic, register, dialect, collocation, part of speech, etc. For the purpose of this study, we take as the senses of a word those senses provided by WordNet 1.4 [Miller 90].

WordNet is an on-line lexicon based on psycholinguistic theories. It comprises nouns, verbs, adjectives and adverbs, organised around semantic relations, such as: synonymy and antonymy, hypernymy and hyponymy, meronymy and holonymy. Lexicalised concepts, represented as sets of synonyms called synsets, are the basic elements of WordNet. The senses of a word are represented by synsets, one for each word sense. The version used in this work, WordNet 1.4, contains 83,800 words, 63,300 synsets (word senses) and 87,600 links between concepts.

The nouns of WordNet can be viewed as a tangled hierarchy of hypo/hypernymy relations. Nominal relations include also three kinds of meronymic relations, which can be paraphrased as "member-of", "made-of" and "component-part-of".

SemCor [Miller et al. 93] is a corpus where part of speech and word sense tags (which correspond to WordNet synsets) have been included for all open-class words. SemCor is a subset taken from the Brown Corpus [Francis and Kucera, 67] which comprises approximately 250,000 words from a total of 1 million words. The coverage in WordNet of the senses for open-class words in SemCor reaches 96% according to Miller et al. The tagging was done manually, and the error rate reported is around 10% for polysemous words.

An Experiment on Word Sense Disambiguation of the Brown Corpus using WordNet[1]

Eneko Agirre.[2]*
Departamento de Lenguajes y Sistemas Informáticos. Universidad del Pais Vasco.
p.k. 649, 20080 Donostia. Spain. jibagbee@si.ehu.es

German Rigau.[3]**
Departament de Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya.
Pau Gargallo 5, 08028 Barcelona. Spain. g.rigau@lsi.upc.es

Abstract.

This paper presents a method for the resolution of lexical ambiguity and its automatic evaluation over the Brown Corpus. The method relies on the use of the wide-coverage noun taxonomy of WordNet and the notion of conceptual distance among concepts, captured by a Conceptual Density formula developed for this purpose. This fully automatic method requires no hand coding of lexical entries, hand tagging of text or any kind of training process. The results of the experiments have been automatically evaluated against SemCor, the sense-tagged version of the Brown Corpus.

Keywords: Word Sense Disambiguation, Conceptual Distance, WordNet, SemCor.

1 Introduction

Word sense disambiguation is a long-standing problem in computational linguistics. Much of recent work in lexical ambiguity resolution offers the prospect that a disambiguation system might be able to input unrestricted text and tag each word with the most likely sense with fairly reasonable accuracy and efficiency. The main idea is to attempt to use the context of the word to be disambiguated together with information about each of its word senses to solve this problem.

Several interesting experiments have been performed in recent years using pre-existing lexical knowledge resources. [Cowie et al. 92] and [Guthrie et al. 93] describe a method for lexical disambiguation of text using the definitions in the machine-readable version of the LDOCE dictionary as in the method described in [Lesk 86], but using simulated annealing for efficiency reasons. [Yarowsky 92] combines the use of the Grolier encyclopaedia as a training corpus with the categories of the Roget's International Thesaurus to create a statistical model for the word sense disambiguation problem with excellent results. [Gale et al. 93] explains a statistical approach using bilingual parallel corpora. [Wilks et al. 93] perform several interesting statistical disambiguation experiments using cooccurrence data collected from LDOCE. [Sussna 93], [Voorhees 93] and [Richarson et al. 94] define disambiguation programs based in WordNet with the goal of improving precision and coverage during document indexing.

Although each of these techniques looks somewhat promising for disambiguation, either they have been only applied to a small number of words, a few sentences or they are not in a public domain corpus. For this reason we have tried to disambiguate all the nouns from texts in the sense tagged version of the Brown corpora [Francis & Kucera 67], [Miller et al. 93], also called Se-

---

1.* *The research described in this paper was done during a stay in the Computer Research Laboratory, New Mexico State University, Las Cruces, New Mexico.*
2.** *Eneko Agirre was supported by a grant from the Basque Government.*
3.*** *German Rigau was supported by a grant from the Ministerio de Educación y Ciencia.*