Draft version of the paper published in NLDB'10

Automatic Word Sense Disambiguation using Cooccurrence and Hierarchical Information

David Fernandez-Amoros*[1], Ruben Heradio †[1], Jose Antonio Cerrada‡[1], and Carlos Cerrada§[1]

[1]ETS de Ingenieria Informatica, Universidad Nacional de Educacion a Distancia, Madrid, Spain

Abstract

We review in detail here a polished version of the systems with which we participated in the SENSEVAL-2 competition English tasks (all words and lexical sample). It is based on a combination of selectional preference measured over a large corpus and hierarchical information taken from WordNet, as well as some additional heuristics. We use that information to expand sense glosses of the senses in WordNet and compare the similarity between the contexts vectors and the word sense vectors in a way similar to that used by Yarowsky and Schuetze. A supervised extension of the system is also discussed. We provide new and previously unpublished evaluation over the SemCor collection, which is two orders of magnitude larger than SENSEVAL-2 collections as well as comparison with baselines. Our systems scored first among unsupervised systems in both tasks. We note that the method is very sensitive to the quality of the characterizations of word senses; glosses being much better than training examples.

1 Introduction

We advocate unsupervised techniques for Word Sense Disambiguation (WSD). Supervised techniques often offer better results but they need reliable training examples which are expensive in terms of human taggers. Furthermore, the problem is considerably more complex than others that have been successfully tackled with machine learning techniques (such as part-of-speech tagging) so it is unclear what amount of training examples will be enough to solve the problem to a reasonable extent, provided that it is a matter of quantity. In the next section we describe some related work. In section 3, the process of constructing the relevance matrix is resumed. In section 4, we present the particular heuristics used for the competing systems. We show the results in section 5. Finally, in section 6 we discuss the results and draw some conclusions.

2 Related work

We are interested in performing in-depth measures of the disambiguation potential of different information sources. We have previously investigated the informational value of semantic distance measures in

*david@lsi.uned.es

†rheradio@issi.uned.es

‡jcerrada@issi.uned.es

§ccerrada@issi.uned.es

[4]. For SENSEVAL-2, we have combined word cooccurrence and hierarchical information as sources of disambiguation evidence [5].

Cooccurrence counting has played an important role in the area of WSD, starting with [7] whose algorithm, consisted in counting co-occurrences of words in the sense definitions of the words in context. To disambiguate a word, the sense with a higher number of co-occurrences with the sense definitions of the other words in the sentence was chosen. He did not make a rigorous evaluation of his algorithm but it has been widely used as a starting point in developing more complex techniques, such as those presented in [2]. They realized that the amount of noise could be reduced if the process of counting co-occurrences was limited to one sense of a word at a time. To solve the combinatorial explosion problem they employed the non-linear optimization technique known as *simulated annealing*. [9] makes the following claim about co-occurrences of words and the senses associated with them:

1. The probability of a relationship between two word-senses occurring in the same sentence is high enough to make it possible to extract useful information from statistics of co-occurrence.

2. The extent to which this probability is above the probability of chance cooccurrence provides an indicator of the strength of of the relationship.

3. If there are more and stronger relationships among the word-senses in one assignment of word-senses to words in a sentence than in another, then the first assignment is more likely to be correct.

Wilks et al. counted the co-occurrences of words in the LDOCE (Longman's Dictionary of Contemporary English) and used a combination of relatedness functions between words (using the co-occurrence information) and a set of similarity measures between sense-vectors and co-occurrence vectors. They used all this information to perform quite a number of experiments disambiguating the word *bank*.

[8] measured the co-occurrence in a Wall Street Journal corpus and expanded the disambiguation contexts by adding the words most related via co-occurrence with the words in context. They used this information to perform WSD applied to Information Retrieval (IR).

[10] computed the co-occurrences of words in the Grolier Enclyclopedia. He calculated $\frac{Pr(w|RCat)}{Pr(w)}$ that is, the probability of a word w appearing in the context of a Roget Category divided by its overall probability in the corpus.

3 The Relevance matrix

Before building our systems we have developed a resource we have called the *relevance matrix*. The raw data used to build the matrix comes from the Project Gutenberg (PG) [1].

At the time of the creation of the matrix, the PG consisted of more than 3000 books of diverse genres. We have adapted these books for our purpose : First, discarding books not written in English; we applied a simple heuristic that uses the percentage of English stop words in the text. This method is considered acceptable in the case of large texts. We stripped off the disclaimers, then proceeded to tokenize, lemmatize, strip punctuation and stop words and detect numbers and proper nouns. The result is a collection of around 1.3GB of text.

3.1 Cooccurrence matrix

We have built a vocabulary of the 20000 most frequent words (or labels, as we have changed all the proper nouns detected to the label PROPER_NOUN and all numbers detected to NUMBER) in the text and

[1] http://promo.net/pg

Table 1: Most relevant words for a sample of words

word	Relevant words
art	decorative pictorial thou proficient imitative hub archaeology whistler healing angling culinary sculpture corruptible photography handicraft adept
authority	vested abrogate municipal judiciary legislative usurped subordination marital parental guaranty federal papal centralized concurrent unquestioned ecclesiastical
bar	capstan shuttered barkeeper bartender transverse visor barmaid bolt iron cage padlock stanchion socket gridiron whitish croup
blind	deaf lame maimed buff unreasoning sightless paralytic leper dumb slat eyesight crustacean groping venetian necked wilfully
carry	stretcher loads portage suitcase satchel swag knapsack valise sedan litter petrol momentum connotation sling basket baggage
chair	bottomed sedan wicker swivel upholster cushioned washstand horsehair rocker seating rickety tilted mahogany plush vacate carven
church	congregational anglican methodist militant presbyterian romish lutheran episcopal steeple liturgy wesleyan catholic methodists spire baptists chancel
circuit	node relay integrated transmitter mac testing circuit billed stadium carbon installation tandem microphone platinum id generator

a symmetric cooccurrence matrix between these words within a context of 61 words (we thought a broad context of radius 30 would be appropriate since we are trying to capture vague semantic relations).

3.2 Relevance matrix

In a second step, we have built another symmetric matrix, which we have called *relevance matrix*, using a mutual information measure between the words (or labels), so that for two words a and b, the entry for them would be $\frac{P(a \cap b)}{P(b)P(a)}$, where $P(a)$ is the probability of finding the word a in a random context of a given size. $P(a \cap b)$ is the probability of finding both a and b in a random context of the fixed size. We approximated those probabilities with the frequencies in the corpus.

The mutual information measure is known to overestimate the relation between low frequency words or those with very dissimilar frequencies [3]. To avoid that problem we adopted a similar approach as [1] and ignored the entries where the frequency of the intersection was less than 50.

We have also introduced a threshold of 2 below which we set the entry to zero for practical purposes (we are only interested in strongly related pairs of words). We think that this is a valuable resource that could be of interest for many other applications other than WSD. Also, it will grow in quality as soon as we feed it with a larger amount of raw data. An example of the most relevant words for some of the words in the lexical sample task of the SENSEVAL-2 competition can be seen in table 1.

4 Cascade of heuristics

We have developed a very simple language in order to systematize the experiments. This language allows the construction of WSD systems comprised of different heuristics that are applied in cascade so that each word to be disambiguated is presented to the first heuristic, and if it fails to disambiguate, then the word is passed on to the second heuristic and so on. We now present the heuristics considered to build the systems.

4.1 Monosemous expressions

Monosemous expressions are simply unambiguous words in the case of the all words English task since we did not take advantage of the satellite features. In the case of the lexical sample English task, however, the annotations include multiword expressions. We have implemented a multiword term detector that considers the multiword terms from WordNet *index.sense* file and detects them in the test file using a multilevel backtracking algorithm that takes account of the inflected and base forms of the components of a particular multiword in order to maximize multiword detection. We tested this algorithm against the PG and found millions of these multiword terms.

We restricted ourselves to the multiwords already present in the training file since there are, apparently, multiword expressions that where overlooked during manual tagging (for instance the WordNet expression *the_good_old_days* is not hand-tagged as such in the test files) In the SemCor collection the multiwords are already detected and we just used them as they were.

4.2 Statistical filter

WordNet comes with a file, *cntlist*, literally *a file listing number of times each tagged sense occurs in a semantic concordance* so we use this to compute the relative probability of a sense given a word (approximate in the case of collections other than SemCor). Using this information, we eliminated the senses that had a probability under 10% and if only one sense remains we choose it. Otherwise we go on to the next heuristic. In other words, we didn't apply complex techniques with words which are highly skewed in meaning.

Some people may argue that this is a supervised approach. In our opinion, the *cntlist* information does not make a system supervised *per se*, because it is standard information provided as part of the dictionary. In fact, we think that every dictionary should make that information available. Word senses are not just made up. An important criterion for incorporating a new sense is SFIP, that is Sufficiently Frequent and Insufficiently Predictable. If one wants to build a dictionary for a language it is customary to create meanings for the words in a large corpus and assign them. But in order to do so it is imperative that the frequency of distribution of the senses be kept. Besides, we don't use the examples to feed or train any procedure.

4.3 Relevance filter

This heuristic makes use of the relevance matrix. In order to assign a score to a sense, we count the cooccurrences of words in the context of the word to be disambiguated with the words in the definition of the senses (the WordNet gloss tokenized, lemmatized and stripped out of stop words and punctuation signs) weighting each cooccurrence by the entry in the relevance matrix for the word to be disambiguated and the word whose cooccurrences are being counted. Also, not to favor senses with long glosses we divide by the number of terms in the gloss.

We use idf (inverse document frequency) a concept typically used in information retrieval to weight the terms in a way that favors specific terms over general ones. Finally, for each word in context we weight its contribution with a $distance$ function. The distance function is a gaussian that favors terms coming from the immediate surroundings of the target word. So, if s is a sense of the word α whose definition is S and C is the context in which α is to be disambiguated, the score for s would be calculated by eq. 1.

$$\sum_{w \in C} R_{w\alpha} \text{freq}(w, C) \text{distance\_weight}(w, \alpha) \text{freq}(w, S) \text{idf}(w, \alpha) \tag{1}$$

Where $\text{idf}(w, \alpha) = \log \frac{N}{d_w}$, with N being the number of senses for word α and d_w the number of sense glosses in which w appears. $\text{freq}(w, C)$ is the frequency of word w in the context C, $\text{freq}(w, S)$ is the frequency of w in the sense gloss S and $\text{distance\_weight}(w, \alpha) = 0.1 + e^{-\text{distance}(w,\alpha)^2/2\sigma^2}$. σ has been assigned the value 2.17 in our experiments with SENSEVAL-1 data.

The idea is to prime the occurrences of words that are relevant to the word being disambiguated and give low credit (possibly none) to the words that are incidentally used in the context.

Also, in the all words task (where POS tags from the TreeBank are provided) we have considered only the context words that have a POS tag *compatible* with that of the word being disambiguated. For instance, adverbs are used to disambiguate verbs, but not to disambiguate nouns.

We also filtered out senses with low values in the *cntlist* file, and in any case we only considered at most the first six senses of a word. We finally did not use this heuristic. Instead, we used it as an starting point for the next ones.

4.4 Enriched senses and mixed filter

The problem with the Relevance filter is that there is little overlapping between the definitions of the senses and the contexts in terms of cooccurrence (after removing stop words and computing idf) which means that the previous heuristic didn't disambiguate many words. To overcome this problem, we enrich the senses characteristic vectors taking for each word in the vector the words related to it via the relevance matrix weights. This corresponds to the algebraic notion of multiplying the matrix and the characteristic vector. In other words, if R is the relevance matrix and v our characteristic vector we would finally use Rv.

Since the matrix is so big, this product is very expensive computationally. Instead, given the fact that we are using it just to compute the score for a sense with equation 2, where, s is a sense of the word α to be disambiguated, whose definition is S and C is the context in which α is to be disambiguated, we only need to sum up a number of terms which is the product of the number of terms in C multiplied by the number of terms in S.

$$\sum_{i \in C} \sum_{j \in S} R_{ij} \text{freq}(i, C)) \text{distance\_weight}(i, \alpha) \text{freq}(j, S) \tag{2}$$

One interesting effect of the relevance matrix being symmetric is that it can be easily proved that the effect of enriching the sense characteristic vectors is the same as enriching the contexts.

The *mixed filter* is a particular case of this one, when we also discard senses with low relative frequency in SemCor.

For those cases that could not be covered by other heuristics we employed the first sense heuristic. The difference is almost negligible since it is rarely used.

Table 2: UNED Unsupervised heuristics

	All Words task			Lexical Sample			SemCor		
Heuristic	Coverage	Prec	Recall	Coverage	Prec	Recall	Coverage	Prec	Recall
Monosemous	18%	89%	16%	4%	58%	2%	20%	100%	20%
Statistical Filter	23%	68%	16%	25%	43%	11%	28%	83%	23%
Mixed Filter	34%	38%	13%	44%	34%	15%	33%	42%	13%
Enriched Senses	21%	50%	10%	23%	47%	11%	15%	46%	7%
First Sense	1%	59%	0%	0%	100%	0%	1%	61%	1%
Total	99%	57%	57%	99%	41%	40%	100%	67%	67%

Table 3: UNED Unsupervised vs baselines

	All Words task			Lexical Sample			SemCor		
System	Coverage	Prec	Recall	Coverage	Prec	Recall	Coverage	Prec	Recall
FIRST	99%	60%	59%	99%	43%	42%	100%	75%	75%
UNED	99%	57%	57%	99%	41%	40%	100%	67%	67%
Mixed Filter	76%	59%	46%	75%	38%	29%	82%	71%	58%
Enriched Senses	97%	46%	45%	98%	35%	34%	97%	50%	49%
RANDOM	99%	36%	35%	99%	18%	18%	100%	40%	40%
Statistical Filter	41%	77%	32%	30%	46%	13%	49%	90%	44%
Monosemous	18%	89%	16%	4%	58%	2%	20%	100%	20%

5 Systems and Results

The heuristics we used and the results evaluated for SENSEVAL-2 and SemCor for each of them are shown in Table 2. If the individual heuristics are used as standalone WSD systems we would obtain the results in Table 3.

We have also built a supervised variant of the previous systems. We have added the training examples to the definitions of the senses giving the same weight to the definition and to all the examples as a whole (i.e. definitions are given more credit than examples). The evaluation is only interesting for the lexical sample, the results are given in Table 4 and discussed in the next section.

It is worth mentioning the difference in the size of the collections: The all words task consisted of 2473 test cases, the lexical sample task had 4328 test cases and the SemCor collection, 192639. The SemCor evaluation, which is nearly two orders of magnitude larger than the SENSEVAL tasks, is perhaps the main contribution of this paper insofar as results are much more significant.

6 Discussion and conclusions

The results obtained support Wilks' claim (as quoted in the related work section) in that co-occurrence information is an interesting source of evidence for WSD. If we look at table 3, we see that the Enriched Senses heuristic performs 27% better that random for the all words and 25% better for the SemCor collection. This relative improvement jumps to 94% in the case of the lexical sample. This is not surprising

Table 4: UNED Trained heuristics & UNED Trained vs baselines

	Lexical Sample		
Heuristic	Coverage	Prec	Recall
Monosemous	4%	58%	2%
Statistical Filter	25%	43%	11%
Mixed Filter	44%	22%	10%
Enriched Senses	23%	24%	5%
First Sense	0%	0%	0%
Total	99%	30%	29%

	Lexical Sample		
System	Coverage	Prec	Recall
First Sense	99%	43%	42%
UNED	99%	30%	29%
Mixed Filter	75%	32%	24%
Enriched Senses	99%	15%	15%
Random	99%	18%	18%
Statistical Filter	30%	46%	13%
Monosemous	4%	58%	2%

since the lexical sample contains no monosemous words at all, but actually rather ambiguous ones, which severely affects the random heuristic performance.

The results in table 4 are surprising, in the sense that using training examples to enrich the senses characteristic vectors actually harms the performance. So, while using just the sense glosses to enrich via the relevance matrix yielded good results, as we have just seen in the previous paragraph, adding the training examples makes the precision fall from 47% to 24% as a heuristic in the cascade, and from 35% to 15% used as a baseline on its own. It is pertinent to remind here that all the training examples for a word were given the same weight as the sense gloss, so using just training examples to enrich the sense would probably yield catastrophic results.

We think it would be worth experimenting with smoothing techniques for the matrix such as the one described by [6] since we have experienced the same kind of problems mixing the frequencies of dissimilarly occurring words.

We were very confident that the relevance filter would yield good results as we have already evaluated it against the SENSEVAL-1 and SemCor data. We felt however that we could improve the coverage of the heuristic enriching the definitions multiplying by the matrix. The quality of the information used to characterize word senses seems to be very important, since multiplying by the matrix gives good results with the glosses, however the precision degrades terribly if we multiply the matrix with the training examples plus the glosses.

As for the overall scores, the unsupervised lexical sample obtained the highest recall of the unsupervised systems in SENSEVAL-2, which proves that carefully implementing simple techniques still pays off. In the all words task we also obtained the highest recall among the unsupervised systems.

References

[1] Kenneth W. Church and P. Hanks. Word association norms, mutual information and lexicography. In *27th Annual Conference of the Association of Computational Linguistics*, pages 76–82, 1989.

[2] Jim Cowie, J. Guthrie, and L. Guthrie. Lexical disambiguation using simulated annealing. In *International conference in computational linguistics (COLING), Nantes*, pages 359–365, 1992.

[3] Ted E. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

[4] D. Fernández-Amorós, J. Gonzalo, and F. Verdejo. The role of conceptual relations in word sense disambiguation. In *Applications of Natural Language to Information Systems (NLDB), Madrid*, pages 87–98, 2001.

[5] D. Fernández-Amorós, J. Gonzalo, and F. Verdejo. The uned systems at senseval-2. In *Proceedings of the* 2^{nd} *International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL), Toulouse*, 2001.

[6] William A. Gale, Kenneth W. Church, and David Yarowsky. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26(5):415–439, 1993.

[7] Michael E. Lesk. Automatic sense disambiguation using machine readable dictionaries : How to tell a pine cone from an ice cream cone. In *Proceedings of SIGDOC (Special Interest Group for Documentation) Conference, Toronto, Canada*, 1986.

[8] H. Schuetze and J. Pedersen. Information retrieval based on word senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval, pages 161–175, Las Vegas, NV, 1995.*, 1995.

[9] Yorick Wilks, D. Fass, C. Guo, J. McDonald, T. Plate, and B. Slator. Providing machine tractable dictionary tools. In *Machine Translation 5(2), 99-151.*, 1990.

[10] David Yarowsky. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of COLING-92*, pages 454–460, Nantes, France, 1992.