Ensemble Methods for Unsupervised WSD

Samuel Brody
School of Informatics
University of Edinburgh
s.brody@sms.ed.ac.uk

Roberto Navigli
Dipartimento di Informatica
Università di Roma "La Sapienza"
navigli@di.uniroma1.it

Mirella Lapata
School of Informatics
University of Edinburgh
mlap@inf.ed.ac.uk

Abstract

Combination methods are an effective way of improving system performance. This paper examines the benefits of system combination for unsupervised WSD. We investigate several voting- and arbiter-based combination strategies over a diverse pool of unsupervised WSD systems. Our combination methods rely on predominant senses which are derived automatically from raw text. Experiments using the SemCor and Senseval-3 data sets demonstrate that our ensembles yield significantly better results when compared with state-of-the-art.

1 Introduction

Word sense disambiguation (WSD), the task of identifying the intended meanings (senses) of words in context, holds promise for many NLP applications requiring broad-coverage language understanding. Examples include summarization, question answering, and text simplification. Recent studies have also shown that WSD can benefit machine translation (Vickrey et al., 2005) and information retrieval (Stokoe, 2005).

Given the potential of WSD for many NLP tasks, much work has focused on the computational treatment of sense ambiguity, primarily using data-driven methods. Most accurate WSD systems to date are supervised and rely on the availability of training data, i.e., corpus occurrences of ambiguous words marked up with labels indicating the appropriate sense given the context (see Mihalcea and Edmonds 2004 and the references therein). A classifier automatically learns disambiguation cues from these hand-labeled examples.

Although supervised methods typically achieve better performance than unsupervised alternatives, their applicability is limited to those words for which sense labeled data exists, and their accuracy is strongly correlated with the amount of labeled data available (Yarowsky and Florian, 2002). Furthermore, obtaining manually labeled corpora with word senses is costly and the task must be repeated for new domains, languages, or sense inventories. Ng (1997) estimates that a high accuracy domain independent system for WSD would probably need a corpus of about 3.2 million sense tagged words. At a throughput of one word per minute (Edmonds, 2000), this would require about 27 person-years of human annotation effort.

This paper focuses on unsupervised methods which we argue are useful for broad coverage sense disambiguation. Unsupervised WSD algorithms fall into two general classes: those that perform token-based WSD by exploiting the similarity or relatedness between an ambiguous word and its context (e.g., Lesk 1986); and those that perform type-based WSD, simply by assigning all instances of an ambiguous word its most frequent (i.e., predominant) sense (e.g., McCarthy et al. 2004; Galley and McKeown 2003). The predominant senses are automatically acquired from raw text without recourse to manually annotated data. The motivation for assigning all instances of a word to its most prevalent sense stems from the observation that current supervised approaches rarely outperform the simple heuristic of choosing the most common sense in the training data, despite taking local context into account (Hoste et al., 2002). Furthermore, the approach allows sense inventories to be tailored to specific domains.

The work presented here evaluates and compares the performance of well-established unsupervised WSD algorithms. We show that these algorithms yield sufficiently diverse outputs, thus motivating the use of combination methods for improving WSD performance. While combination approaches have been studied previously for supervised WSD (Florian et al., 2002), their use in an unsupervised setting is, to our knowledge, novel. We examine several existing and novel combination methods and demonstrate that our combined systems consistently outperform the

state-of-the-art (e.g., McCarthy et al. 2004). Importantly, our WSD algorithms and combination methods do not make use of training material in any way, nor do they use the first sense information available in WordNet.

In the following section, we briefly describe the unsupervised WSD algorithms considered in this paper. Then, we present a detailed comparison of their performance on SemCor (Miller et al., 1993). Next, we introduce our system combination methods and report on our evaluation experiments. We conclude the paper by discussing our results.

2 The Disambiguation Algorithms

In this section we briefly describe the unsupervised WSD algorithms used in our experiments. We selected methods that vary along the following dimensions: (a) the type of WSD performed (i.e., token-based vs. type-based), (b) the representation and size of the context surrounding an ambiguous word (i.e., graph-based vs. word-based, document vs. sentence), and (c) the number and type of semantic relations considered for disambiguation. We base most of our discussion below on the WordNet sense inventory; however, the approaches are not limited to this particular lexicon but could be adapted for other resources with traditional dictionary-like sense definitions and alternative structure.

Extended Gloss Overlap Gloss Overlap was originally introduced by Lesk (1986) for performing token-based WSD. The method assigns a sense to a target word by comparing the dictionary definitions of each of its senses with those of the words in the surrounding context. The sense whose definition has the highest overlap (i.e., words in common) with the context words is assumed to be the correct one. Banerjee and Pedersen (2003) augment the dictionary definition (gloss) of each sense with the glosses of related words and senses. The *extended glosses* increase the information available in estimating the overlap between ambiguous words and their surrounding context.

The range of relationships used to extend the glosses is a parameter, and can be chosen from any combination of WordNet relations. For every sense s_k of the target word we estimate:

$$SenseScore(s_k) = \sum_{Rel \in Relations} Overlap(context, Rel(s_k))$$

where *context* is a simple (space separated) concatenation of all words w_i for $-n \leq i \leq n, i \neq 0$ in a context window of length $\pm n$ around the target word w_0. The overlap scoring mechanism is also parametrized and can be adjusted to take the into account gloss length or to ignore function words.

Distributional and WordNet Similarity McCarthy et al. (2004) propose a method for automatically ranking the senses of ambiguous words from raw text. Key in their approach is the observation that distributionally similar neighbors often provide cues about a word's senses. Assuming that a set of neighbors is available, sense ranking is equivalent to quantifying the degree of similarity among the neighbors and the sense descriptions of the polysemous word.

Let $N(w) = \{n_1, n_2, \ldots, n_k\}$ be the k most (distributionally) similar words to an ambiguous target word w and $senses(w) = \{s_1, s_2, \ldots s_n\}$ the set of senses for w. For each sense s_i and for each neighbor n_j, the algorithm selects the neighbor's sense which has the highest WordNet similarity score (*wnss*) with regard to s_i. The ranking score of sense s_i is then increased as a function of the WordNet similarity score and the distributional similarity score (*dss*) between the target word and the neighbor:

$$RankScore(s_i) = \sum_{n_j \in N_w} dss(w, n_j) \frac{wnss(s_i, n_j)}{\sum_{s'_i \in senses(w)} wnss(s'_i, n_j)}$$

where $wnss(s_i, n_j) = \max_{ns_x \in senses(n_j)} wnss(s_i, ns_x)$.

The predominant sense is simply the sense with the highest ranking score (*RankScore*) and can be consequently used to perform type-based disambiguation. The method presented above has four parameters: (a) the semantic space model representing the distributional properties of the target words (it is acquired from a large corpus representative of the domain at hand and can be augmented with syntactic relations such as subject or object), (b) the measure of distributional similarity for discovering neighbors (c) the number of neighbors that the ranking score takes into account, and (d) the measure of sense similarity.

Lexical Chains Lexical cohesion is often represented via lexical chains, i.e., sequences of related words spanning a topical text unit (Morris and Hirst, 1991). Algorithms for computing lexical chains often perform WSD before inferring which words are semantically related. Here we describe one such disambiguation algorithm, proposed by Galley and McKeown (2003), while omitting the details of creating the lexical chains themselves.

Galley and McKeown's (2003) method consists of two stages. First, a graph is built representing all possible interpretations of the target words

in question. The text is processed sequentially, comparing each word against all words previously read. If a relation exists between the senses of the current word and any possible sense of a previous word, a connection is formed between the appropriate words and senses. The strength of the connection is a function of the type of relationship and of the distance between the words in the text (in terms of words, sentences and paragraphs). Words are represented as nodes in the graph and semantic relations as weighted edges. Again, the set of relations being considered is a parameter that can be tuned experimentally.

In the disambiguation stage, all occurrences of a given word are collected together. For each sense of a target word, the strength of all connections involving that sense are summed, giving that sense a unified score. The sense with the highest unified score is chosen as the correct sense for the target word. In subsequent stages the actual connections comprising the winning unified score are used as a basis for computing the lexical chains.

The algorithm is based on the "one sense per discourse" hypothesis and uses information from every occurrence of the ambiguous target word in order to decide its appropriate sense. It is therefore a type-based algorithm, since it tries to determine the sense of the word in the entire document/discourse at once, and not separately for each instance.

Structural Semantic Interconnections Inspired by lexical chains, Navigli and Velardi (2005) developed Structural Semantic Interconnections (SSI), a WSD algorithm which makes use of an extensive lexical knowledge base. The latter is primarily based on WordNet and its standard relation set (i.e., hypernymy, meronymy, antonymy, similarity, nominalization, pertainymy) but is also enriched with collocation information representing semantic relatedness between sense pairs. Collocations are gathered from existing resources (such as the Oxford Collocations, the Longman Language Activator, and collocation web sites). Each collocation is mapped to the WordNet sense inventory in a semi-automatic manner (Navigli, 2005) and transformed into a *relatedness* edge.

Given a local word context $C = \{w_1, ..., w_n\}$, SSI builds a graph $G = (V, E)$ such that $V = \bigcup_{i=1}^{n} senses(w_i)$ and $(s, s') \in E$ if there is at least one interconnection j between s (a sense of the word) and s' (a sense of its context) in the lexical knowledge base. The set of valid interconnections is determined by a manually-created context-free grammar consisting of a small number of rules. Valid interconnections are computed in advance on the lexical database, not at runtime.

Method	WSD	Context	Relations
LexChains	types	document	first-order
Overlap	tokens	sentence	first-order
Similarity	types	corpus	higher-order
SSI	tokens	sentence	higher-order

Table 1: Properties of the WSD algorithms

Disambiguation is performed in an iterative fashion. At each step, for each sense s of a word in $\mathcal{C}$ (the set of senses of words yet to be disambiguated), SSI determines the degree of connectivity between s and the other senses in $\mathcal{C}$:

$$SSIScore(s) = \frac{\sum_{s' \in \mathcal{C} \setminus \{s\}} \sum_{j \in Interconn(s,s')} \frac{1}{length(j)}}{\sum_{s' \in \mathcal{C} \setminus \{s\}} |Interconn(s,s')|}$$

where $Interconn(s, s')$ is the set of interconnections between senses s and s'. The contribution of a single interconnection is given by the reciprocal of its length, calculated as the number of edges connecting its ends. The overall degree of connectivity is then normalized by the number of contributing interconnections. The highest ranking sense s of word w_i is chosen and the senses of w_i are removed from the context $\mathcal{C}$. The procedure terminates when either $\mathcal{C}$ is the empty set or there is no sense such that its *SSIScore* exceeds a fixed threshold.

Summary The properties of the different WSD algorithms just described are summarized in Table 1. The methods vary in the amount of data they employ for disambiguation. SSI and Extended Gloss Overlap (Overlap) rely on sentence-level information for disambiguation whereas McCarthy et al. (2004) (Similarity) and Galley and McKeown (2003) (LexChains) utilize the entire document or corpus. This enables the accumulation of large amounts of data regarding the ambiguous word, but does not allow separate consideration of each individual occurrence of that word. LexChains and Overlap take into account a restricted set of semantic relations (paths of length one) between any two words in the whole document, whereas SSI and Similarity use a wider set of relations.

3 Experiment 1: Comparison of Unsupervised Algorithms for WSD

3.1 Method

We evaluated the disambiguation algorithms outlined above on two tasks: predominant sense acquisition and token-based WSD. As previously explained, Overlap and SSI were not designed for acquiring predominant senses (see Table 1), but a token-based WSD algorithm can be trivially modified to acquire predominant senses by disambiguating every occurrence of the target word in context and selecting the sense which was chosen most frequently. Type-based WSD algorithms simply tag all occurrences of a target word with its predominant sense, disregarding the surrounding context.

Our first set of experiments was conducted on the SemCor corpus, on the same 2,595 polysemous nouns (53,674 tokens) used as a test set by McCarthy et al. (2004). These nouns were attested in SemCor with a frequency > 2 and occurred in the British National Corpus (BNC) more than 10 times. We used the WordNet 1.7.1 sense inventory.

The following notation describes our evaluation measures: W is the set of all noun types in the SemCor corpus ($|W| = 2,595$), and W_f is the set of noun types with a dominant sense. $senses(w)$ is the set of senses for noun type w, while $f_s(w)$ and $f_m(w)$ refer to w's first sense according to the SemCor gold standard and our algorithms, respectively. Finally, $T(w)$ is the set of tokens of w and $sense_s(t)$ denotes the sense assigned to token t according to SemCor.

We first measure how well our algorithms can identify the predominant sense, if one exists:

$$Acc_{ps} = \frac{|\{w \in W_f \,|\, f_s(w) = f_m(w)\}|}{|W_f|}$$

A baseline for this task can be easily defined for each word type by selecting a sense at random from its sense inventory and assuming that this is the predominant sense:

$$Baseline_{sr} = \frac{1}{|W_f|} \sum_{w \in W_f} \frac{1}{|senses(w)|}$$

We evaluate the algorithms' disambiguation performance by measuring the ratio of tokens for which our models choose the right sense:

$$Acc_{wsd} = \frac{\sum_{w \in W} |\{t \in T(w) | f_m(w) = sense_s(t)\}|}{\sum_{w \in W} |T(w)|}$$

In the predominant sense detection task, in case of ties in SemCor, any one of the predominant senses was considered correct. Also, all algorithms were designed to randomly choose from among the top scoring options in case of a tie in the calculated scores. This introduces a small amount of randomness (less than 0.5%) in the accuracy calculation, and was done to avoid the pitfall of defaulting to the first sense listed in WordNet, which is usually the actual predominant sense (the order of senses in WordNet is based primarily on the SemCor sense distribution).

3.2 Parameter Settings

We did not specifically tune the parameters of our WSD algorithms on the SemCor corpus, as our goal was to use hand labeled data solely for testing purposes. We selected parameters that have been considered "optimal" in the literature, although admittedly some performance gains could be expected had parameter optimization taken place.

For Overlap, we used the semantic relations proposed by Banerjee and Pedersen (2003), namely hypernyms, hyponyms, meronyms, holonyms, and troponym synsets. We also adopted their overlap scoring mechanism which treats each gloss as a bag of words and assigns an n word overlap the score of n^2. Function words were not considered in the overlap computation. For LexChains, we used the relations reported in Galley and McKeown (2003). These are all first-order WordNet relations, with the addition of the *siblings* – two words are considered siblings if they are both hyponyms of the same hypernym. The relations have different weights, depending on their type and the distance between the words in the text. These weights were imported from Galley and McKeown into our implementation without modification.

Because the SemCor corpus is relatively small (less than 700,00 words), it is not ideal for constructing a neighbor thesaurus appropriate for McCarthy et al.'s (2004) method. The latter requires each word to participate in a large number of co-occurring contexts in order to obtain reliable distributional information. To overcome this problem, we followed McCarthy et al. and extracted the neighbor thesaurus from the entire BNC. We also recreated their semantic space, using a RASP-parsed (Briscoe and Carroll, 2002) version of the BNC and their set of dependencies (i.e., Verb-Object, Verb-Subject, Noun-Noun and Adjective-Noun relations). Similarly to McCarthy et al., we used Lin's (1998) measure of distributional similarity, and considered only the 50 highest ranked

Method	Acc_{ps}	$Acc_{wsd/dir}$	$Acc_{wsd/ps}$
Baseline	34.5	–	23.0
LexChains	$48.3^{*\dagger\$}$	–	$40.7^{*\#\dagger\$}$
Overlap	$49.4^{*\dagger\$}$	$36.5^{\$}$	$42.5^{*\dagger\$}$
Similarity	54.9^{*}	–	$46.5^{*\$}$
SSI	53.7^{*}	42.7	47.9^{*}
UpperBnd	100	–	68.4

Table 2: Results of individual disambiguation algorithms on SemCor nouns[2] (*: sig. diff. from Baseline, †: sig. diff. from Similarity, $^{\$}$: sig diff. from SSI, $^{\#}$: sig. diff. from Overlap, $p < 0.01$)

neighbors for a given target word. Sense similarity was computed using the Lesk's (Banerjee and Pedersen, 2003) similarity measure[1].

3.3 Results

The performance of the individual algorithms is shown in Table 2. We also include the baseline discussed in Section 3 and the upper bound of defaulting to the first (i.e., most frequent) sense provided by the manually annotated SemCor. We report predominant sense accuracy (Acc_{ps}), and WSD accuracy when using the automatically acquired predominant sense ($Acc_{wsd/ps}$). For token-based algorithms, we also report their WSD performance in context, i.e., without use of the predominant sense ($Acc_{wsd/dir}$).

As expected, the accuracy scores in the WSD task are lower than the respective scores in the predominant sense task, since detecting the predominant sense correctly only insures the correct tagging of the instances of the word with that first sense. All methods perform significantly better than the baseline in the predominant sense detection task (using a χ^2-test, as indicated in Table 2). LexChains and Overlap perform significantly worse than Similarity and SSI, whereas LexChains is not significantly different from Overlap. Likewise, the difference in performance between SSI and Similarity is not significant. With respect to WSD, all the differences in performance are statistically significant.

[1]This measure is identical to the Extended gloss Overlap from Section 2, but instead of searching for overlap between an extended gloss and a word's context, the comparison is done between two extended glosses of two synsets.

[2]The LexChains results presented here are not directly comparable to those reported by Galley and McKeown (2003), since they tested on a subset of SemCor, and included monosemous nouns. They also used the first sense in SemCor in case of ties. The results for the Similarity method are slightly better than those reported by McCarthy et al. (2004) due to minor improvements in implementation.

	Overlap	LexChains	Similarity
LexChains	28.05		
Similarity	35.87	33.10	
SSI	30.48	31.67	37.14

Table 3: Algorithms' pairwise agreement in detecting the predominant sense (as % of all words)

Interestingly, using the predominant sense detected by the Gloss Overlap and the SSI algorithm to tag all instances is preferable to tagging each instance individually (compare $Acc_{wsd/dir}$ and $Acc_{wsd/ps}$ for Overlap and SSI in Table 2). This means that a large part of the instances which were not tagged individually with the predominant sense were actually that sense.

A close examination of the performance of the individual methods in the predominant-sense detection task shows that while the accuracy of all the methods is within a range of 7%, the actual words for which each algorithm gives the correct predominant sense are very different. Table 3 shows the degree of overlap in assigning the appropriate predominant sense among the four methods. As can be seen, the largest amount of overlap is between Similarity and SSI, and this corresponds approximately to $\frac{2}{3}$ of the words they correctly label. This means that each of these two methods gets more than 350 words right which the other labels incorrectly.

If we had an "oracle" which would tell us which method to choose for each word, we would achieve approximately 82.4% in the predominant sense task, giving us 58% in the WSD task. We see that there is a large amount of complementation between the algorithms, where the successes of one make up for the failures of the others. This suggests that the errors of the individual methods are sufficiently uncorrelated, and that some advantage can be gained by combining their predictions.

4 Combination Methods

An important finding in machine learning is that a set of classifiers whose individual decisions are combined in some way (an *ensemble*) can be more accurate than any of its component classifiers, provided that the individual components are relatively accurate and diverse (Dietterich, 1997). This simple idea has been applied to a variety of classification problems ranging from optical character recognition to medical diagnosis, part-of-speech tagging (see Dietterich 1997 and van Halteren et al. 2001 for overviews), and notably supervised

WSD (Florian et al., 2002).

Since our effort is focused exclusively on unsupervised methods, we cannot use most machine learning approaches for creating an ensemble (e.g., stacking, confidence-based combination), as they require a labeled training set. We therefore examined several basic ensemble combination approaches that do not require parameter estimation from training data.

We define $Score(M_i, s_j)$ as the (normalized) score which a method M_i gives to word sense s_j. The predominant sense calculated by method M_i for word w is then determined by:

$$PS(M_i, w) = \underset{s_j \in senses(w)}{\text{argmax}} \; Score(M_i, s_j)$$

All ensemble methods receive a set $\{M_i\}_{i=1}^k$ of individual methods to combine, so we denote each ensemble method by $MethodName(\{M_i\}_{i=1}^k)$.

Direct Voting Each ensemble component has one vote for the predominant sense, and the sense with the most votes is chosen. The scoring function for the voting ensemble is defined as:

$$Score(Voting(\{M_i\}_{i=1}^k), s)) = \sum_{i=1}^{k} eq[s, PS(M_i, w)]$$

$$\text{where } eq[s, PS(M_i, w)] = \begin{cases} 1 & \text{if } s = PS(M_i, w) \\ 0 & \text{otherwise} \end{cases}$$

Probability Mixture Each method provides a probability distribution over the senses. These probabilities (normalized scores) are summed, and the sense with the highest score is chosen:

$$Score(ProbMix(\{M_i\}_{i=1}^k), s)) = \sum_{i=1}^{k} Score(M_i, s)$$

Rank-Based Combination Each method provides a ranking of the senses for a given target word. For each sense, its placements according to each of the methods are summed and the sense with the lowest total placement (closest to first place) wins.

$$Score(Ranking(\{M_i\}_{i=1}^k), s)) = \sum_{i=1}^{k} (-1) \cdot Place_i(s)$$

where $Place_i(s)$ is the number of distinct scores that are larger or equal to $Score(M_i, s)$.

Arbiter-based Combination One WSD method can act as an arbiter for adjudicating disagreements among component systems. It makes sense for the adjudicator to have reasonable performance on its own. We therefore selected SSI as the arbiter since it had the best accuracy on the WSD task (see Table 2). For each disagreed word w, and for each sense s of w assigned by any of the systems in the ensemble $\{M_i\}_{i=1}^k$, we calculate the following score:

$$Score(Arbiter(\{M_i\}_{i=1}^k), s) = SSIScore^*(s)$$

where $SSIScore^*(s)$ is a modified version of the score introduced in Section 2 which exploits as a context for s the set of agreed senses and the remaining words of each sentence. We exclude from the context used by SSI the senses of w which were not chosen by any of the systems in the ensemble . This effectively reduces the number of senses considered by the arbiter and can positively influence the algorithm's performance, since it eliminates noise coming from senses which are likely to be wrong.

Method	Acc_{ps}	$Acc_{wsd/ps}$
Similarity	54.9	46.5
SSI	53.5	47.9
Voting	$57.3^{\dagger\$}$	$49.8^{\dagger\$}$
PrMixture	$57.2^{\dagger\$}$	$50.4^{\dagger\$\ddagger}$
Rank-based	$58.1^{\dagger\$}$	$50.3^{\dagger\$\ddagger}$
Arbiter-based	$56.3^{\dagger\$}$	$48.7^{\dagger\$\ddagger}$
UpperBnd	100	68.4

Table 4: Ensemble Combination Results (†: sig. diff. from Similarity, \$: sig. diff. from SSI, ‡: sig. diff. from Voting, $p < 0.01$)

5 Experiment 2: Ensembles for Unsupervised WSD

5.1 Method and Parameter Settings

We assess the performance of the different ensemble systems on the same set of SemCor nouns on which the individual methods were tested. For the best ensemble, we also report results on disambiguating all nouns in the Senseval-3 data set. We focus exclusively on nouns to allow comparisons with the results obtained from SemCor. We used the same parameters as in Experiment 1 for constructing the ensembles. As discussed earlier, token-based methods can disambiguate target words either in context or using the predominant sense. SSI was employed in the predominant sense setting in our arbiter experiment.

5.2 Results

Our results are summarized in Table 4. As can be seen, all ensemble methods perform significantly

Ensemble	Acc_{ps}	$Acc_{wsd/ps}$
Rank-based	58.1	50.3
Overlap	57.6 (−0.5)	49.7 (−0.6)
LexChains	57.2 (−0.7)	50.2 (−0.1)
Similarity	56.3 (−1.8)	49.4 (−0.9)
SSI	56.3 (−1.8)	48.2 (−2.1)

Table 5: Decrease in accuracy as a result of removal of each method from the rank-based ensemble.

better than the best individual methods, i.e., Similarity and SSI. On the WSD task, the voting, probability mixture, and rank-based ensembles significantly outperform the arbiter-based one. The performances of the probability mixture, and rank-based combinations do not differ significantly but both ensembles are significantly better than voting. One of the factors contributing to the arbiter's worse performance (compared to the other ensembles) is the fact that in many cases (almost 30%), none of the senses suggested by the disagreeing methods is correct. In these cases, there is no way for the arbiter to select the correct sense. We also examined the relative contribution of each component to overall performance. Table 5 displays the drop in performance by eliminating any particular component from the rank-based ensemble (indicated by −). The system that contributes the most to the ensemble is SSI. Interestingly, Overlap and Similarity yield similar improvements in WSD accuracy (0.6 and 0.9, respectively) when added to the ensemble.

Figure 1 shows the WSD accuracy of the best single methods and the ensembles as a function of the noun frequency in SemCor. We can see that there is at least one ensemble outperforming any single method in every frequency band and that the rank-based ensemble consistently outperforms Similarity and SSI in all bands. Although Similarity has an advantage over SSI for low and medium frequency words, it delivers worse performance for high frequency words. This is possibly due to the quality of neighbors obtained for very frequent words, which are not semantically distinct enough to reliably discriminate between different senses.

Table 6 lists the performance of the rank-based ensemble on the Senseval-3 (noun) corpus. We also report results for the best individual method, namely SSI, and compare our results with the best unsupervised system that participated in Senseval-3. The latter was developed by Strapparava et al. (2004) and performs domain driven disambiguation (IRST-DDD). Specifically, the approach compares the domain of the context surrounding the target word with the domains of its senses and uses a version of WordNet augmented with domain labels (e.g., economy, geography). Our baseline selects the first sense randomly and uses it to disambiguate all instances of a target word. Our upper bound defaults to the first sense from SemCor. We report precision, recall and Fscore. In cases where precision and recall figures coincide, the algorithm has 100% coverage.

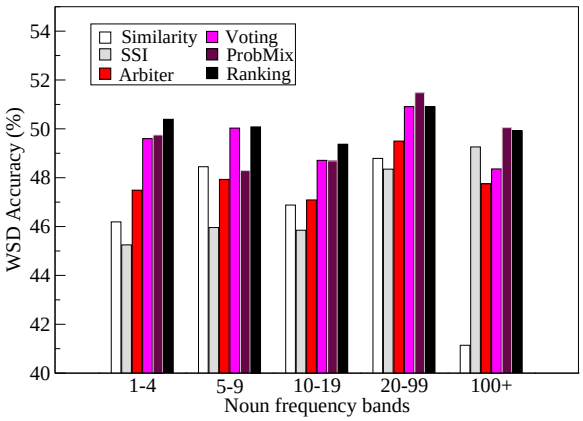

Figure 1: WSD accuracy as a function of noun frequency in SemCor

Method	Precision	Recall	Fscore
Baseline	36.8	36.8	36.8
SSI	62.5	62.5	62.5
IRST-DDD	63.3	61.2	62.2
Rank-based	63.9	63.9	63.9
UpperBnd	68.7	68.7	68.7

Table 6: Results of individual disambiguation algorithms and rank-based ensemble on Senseval-3 nouns

As can be seen the rank-based, ensemble outperforms both SSI and the IRST-DDD system. This is an encouraging result, suggesting that there may be advantages in developing diverse classes of unsupervised WSD algorithms for system combination. The results in Table 6 are higher than those reported for SemCor (see Table 4). This is expected since the Senseval-3 data set contains monosemous nouns as well. Taking solely polysemous nouns into account, SSI's Fscore is 53.39% and the ranked-based ensemble's 55.0%. We further note that not all of the components in our ensemble are optimal. Predominant senses for Lesk and LexChains were estimated from the Senseval-3 data, however a larger corpus would probably yield more reliable estimates.

6 Conclusions and Discussion

In this paper we have presented an evaluation study of four well-known approaches to unsupervised WSD. Our comparison involved type- and token-based disambiguation algorithms relying on different kinds of WordNet relations and different amounts of corpus data. Our experiments revealed two important findings. First, type-based disambiguation yields results superior to a token-based approach. Using predominant senses is preferable to disambiguating instances individually, even for token-based algorithms. Second, the outputs of the different approaches examined here are sufficiently diverse to motivate combination methods for unsupervised WSD. We defined several ensembles on the predominant sense outputs of individual methods and showed that combination systems outperformed their best components both on the SemCor and Senseval-3 data sets.

The work described here could be usefully employed in two tasks: (a) to create preliminary annotations, thus supporting the "annotate automatically, correct manually" methodology used to provide high volume annotation in the Penn Treebank project; and (b) in combination with supervised WSD methods that take context into account; for instance, such methods could default to an unsupervised system for unseen words or words with uninformative contexts.

In the future we plan to integrate more components into our ensembles. These include not only domain driven disambiguation algorithms (Strapparava et al., 2004) but also graph theoretic ones (Mihalcea, 2005) as well as algorithms that quantify the degree of association between senses and their co-occurring contexts (Mohammad and Hirst, 2006). Increasing the number of components would allow us to employ more sophisticated combination methods such as unsupervised rank aggregation algorithms (Tan and Jin, 2004).

Acknowledgements

We are grateful to Diana McCarthy for her help with this work and to Michel Galley for making his code available to us. Thanks to John Carroll and Rob Koeling for insightful comments and suggestions. The authors acknowledge the support of EPSRC (Brody and Lapata; grant EP/C538447/1) and the European Union (Navigli; Interop NoE (508011)).

References

Banerjee, Satanjeev and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th IJCAI*. Acapulco, pages 805–810.

Briscoe, Ted and John Carroll. 2002. Robust accurate statistical annotation of general text. In *Proceedings of the 3rd LREC*. Las Palmas, Gran Canaria, pages 1499–1504.

Dietterich, T. G. 1997. Machine learning research: Four current directions. *AI Magazine* 18(4):97–136.

Edmonds, Philip. 2000. Designing a task for SENSEVAL-2. Technical note.

Florian, Radu, Silviu Cucerzan, Charles Schafer, and David Yarowsky. 2002. Combining classifiers for word sense disambiguation. *Natural Language Engineering* 1(1):1–14.

Galley, Michel and Kathleen McKeown. 2003. Improving word sense disambiguation in lexical chaining. In *Proceedings of the 18th IJCAI*. Acapulco, pages 1486–1488.

Hoste, Véronique, Iris Hendrickx, Walter Daelemans, and Antal van den Bosch. 2002. Parameter optimization for machine-learning of word sense disambiguation. *Language Engineering* 8(4):311–325.

Lesk, Michael. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th SIGDOC*. New York, NY, pages 24–26.

Lin, Dekang. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th ICML*. Madison, WI, pages 296–304.

McCarthy, Diana, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant senses in untagged text. In *Proceedings of the 42th ACL*. Barcelona, Spain, pages 280–287.

Mihalcea, Rada. 2005. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of the HLT/EMNLP*. Vancouver, BC, pages 411–418.

Mihalcea, Rada and Phil Edmonds, editors. 2004. *Proceedings of the SENSEVAL-3*. Barcelona, Spain.

Miller, George A., Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Proceedings of the ARPA HLT Workshop*. Morgan Kaufman, pages 303–308.

Mohammad, Saif and Graeme Hirst. 2006. Determining word sense dominance using a thesaurus. In *Proceedings of the EACL*. Trento, Italy, pages 121–128.

Morris, Jane and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics* 1(17):21–43.

Navigli, Roberto. 2005. Semi-automatic extension of large-scale linguistic knowledge bases. In *Proceedings of the 18th FLAIRS*. Florida.

Navigli, Roberto and Paola Velardi. 2005. Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. *PAMI* 27(7):1075–1088.

Ng, Tou Hwee. 1997. Getting serious about word sense disambiguation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*. Washington, DC, pages 1–7.

Stokoe, Christopher. 2005. Differentiating homonymy and polysemy in information retrieval. In *Proceedings of the HLT/EMNLP*. Vancouver, BC, pages 403–410.

Strapparava, Carlo, Alfio Gliozzo, and Claudio Giuliano. 2004. Pattern abstraction and term similarity for Word Sense Disambiguation: IRST at Senseval-3. In *Proceedings of SENSEVAL-3*. Barcelona, Spain, pages 229–234.

Tan, Pang-Ning and Rong Jin. 2004. Ordering patterns by combining opinions from multiple sources. In *Proceedings of the 10th KDD*. Seattle, WA, pages 22–25.

van Halteren, Hans, Jakub Zavrel, and Walter Daelemans. 2001. Improving accuracy in word class tagging through combination of machine learning systems. *Computational Linguistics* 27(2):199–230.

Vickrey, David, Luke Biewald, Marc Teyssier, and Daphne Koller. 2005. Word-sense disambiguation for machine translation. In *Proceedings of the HLT/EMNLP*. Vancouver, BC, pages 771–778.

Yarowsky, David and Radu Florian. 2002. Evaluating sense disambiguation across diverse parameter spaces. *Natural Language Engineering* 9(4):293–310.