Proceedings of the SIGLEX/SENSEVAL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions, Philadelphia, July 2002, pp. 67-73. Association for Computational Linguistics.

Unsupervised Italian Word Sense Disambiguation using WordNets and Unlabeled Corpora

Radu Florian and **Richard Wicentowski**
Center for Language and Speech Processing
Johns Hopkins University
{rflorian,richardw}@cs.jhu.edu

Abstract

This paper presents a novel method for unsupervised word sense disambiguation, which combines multiple information sources, including semantic relations, large unlabeled corpora, and cross-lingual distributional statistics. This method extends and builds on the JHU system that participated in the SENSEVAL2 exercise. Experiments performed on the SENSEVAL2 Italian lexical-sample data show significant improvements over previously published results on this data set.

1 Introduction

The goal of this paper is to present an unsupervised word sense disambiguation system which extends the JHU system for the Italian lexical sample task which participated in the SENSEVAL2 exercise (Yarowsky et al., 2001). Our system combines word semantic relations, large unlabeled corpora and cross-lingual distributional statistics. The combining system reduces the word sense error rate by 8.2% absolute (13.6% relative error reduction), when compared to the best system submitted in SENSEVAL2.

1.1 Previous Work

Several approaches that address the problem of unsupervised word sense disambiguation (WSD henceforth) have been presented in the past few years. In one of the most widely-cited unsupervised WSD systems, Yarowsky (1995) uses a very small seed set (2-3 examples) to bootstrap a WSD algorithm based on decision lists; the algorithm yields highly accurate results, competitive with similar supervised systems. Schütze (1998) creates word vectors by extracting ambiguous words and their contexts from an unlabeled corpus. After clustering the vectors[1], the disambiguation is performed by selecting the sense centroid closest to the test word vector. Pedersen and Bruce (1998) use an EM-based algorithm to group sentences containing the target word into unlabeled clusters which are then mapped to sense tags.

The existence of the freely available word sense relation database WordNet (Miller, 1995) has enabled the conception of several unsupervised WSD systems, including those by Resnik (1997) who uses syntactically parsed corpora, partially hand-labeled with senses from WordNet (Miller, 1995), to train a selectional preference system, and McCarthy et al. (2001), which uses a selectional preference model similar to Resnik (1997), but without the use of any labeled data, in the "one sense per discourse" paradigm (Gale et al., 1992), achieving good precision in the SENSEVAL2 English all-words task. Using a cross-lingual approach, Rigau et al. (1997) investigates an unsupervised system using a combination of monolingual dictionaries, bilingual dictionaries and the English WordNet. The bilingual dictionaries were used to map words from Spanish and French into English in order to leverage the semantic relationships in the English WordNet. In an approach similar in spirit to the one presented here, Mihalcea and Moldovan (1999) use the entire English WordNet to find sentences in online texts containing high-confidence examples of the target word. The WordNet glosses and acquired sentences are used as training data to automatically create a large sense-tagged corpus.

In work on the same dataset as used in this research, Magnini et al. (2001) manually anno-

[1] Similar to Yarowsky (1995), Schütze (1998) tested his algorithm only on words with two senses.

tates the relevant Italian synsets with a semantic class. The test samples are assigned to one of these classes using a supervised algorithm trained on an annotated English corpus and then these classes were mapped back to the WordNet synsets.

Unsupervised word sense disambiguation using WordNet relations also has also been used in real-world tasks. Some of the earlier examples include Voorhees (1993) (using the hyponym/hypernym relations from WordNet) and Sussna (1993) (using weighted relations derived from WordNet), employing word sense disambiguation to increase the performance of information retrieval systems.

2 Feature Representation

In the model presented in this research, documents[2] are represented as bags of words and/or lemmas; in addition, local n-grams around the ambiguous word are also part of the document's vector:

$$d = \left(d_1 \dots d_{|V|}\right),\ d_j = \frac{c_j}{N} W_j$$

where c_j is the number of times the feature j[3] appears in document d, N is the number of words in d and W_j is a weight associated with feature j[4]. Confusion between the same word participating in multiple feature roles is avoided by appending the feature values with their positional type (e.g. *uomo_L* is different from *uomo* in unmarked bag-of-words context).

All test documents were part-of-speech tagged using the Italian version of the decision tree-based POS tagger described in Schmid (1994). Extracting lemma information from Italian is an important process – Section 6 evaluates a scenario where lemmatization is not used, and shows that a substantial decrease in performance occurs. Lemmatization is performed using the supervised morphological analyzer from Yarowsky and Wicentowski (2000), trained on the filtered output of the POS tagger. We tested the accuracy of the morphological analyzer by randomly selecting 500 adjectives, 500 nouns and 500 verbs[5] in proportion to their token frequency in the unannotated corpus and had them hand-checked by a native speaker. Table 1 presents the POS and lemmatization accuracy for these 1500 words; the lemmatization accuracy is reported only on examples which were correctly labeled by the POS tagger.

	morph. analyzer		POS tagger	
	token	type	token	type
verbs	98.1%	99.0%	99.9%	99.4%
nouns	99.5%	98.7%	96.2%	94.8%
adjs	96.7%	99.6%	85.3%	98.0%

Table 1: Lemmatization and POS tagging accuracy

3 Information Sources

3.1 Italian WordNet

Since no training data was available for this task, we rely on alternative sources of information for inducing sense classification. One central resource in this process is the ItalWordNet, version 1.0, developed by the Italian National Project, SI-TAL (Roventini et al., 2000), which was provided with the data. This structure describes various semantic relationships between words (usually binary relations), including:

- *synonymy* – word u is a synonym of word v if word u has nearly the same definition as word v.
- *antonymy* – word u is an antonym of word v if words u and v have nearly opposite meanings.
- *hyperonymy* - word u is a hyperonym of word v if u is a generalization of word v;
- *hyponymy* – the reverse of the *hyperonymy* relation;
- *meronymy* – word u is a meronym of word v if the object represented by u has the object represented by v as a part (for instance, *car* is a meronym of *wheel*);
- *holonymy* – is the reverse relation of meronymy;

ItalWordNet is not a freely available resource; only the parts that were provided with the task have been used in this research[6].

[2] Throughout the paper, we will use the word *document* to denote the actual context where the ambiguous word appears.

[3] A feature can be a word, a lemma or a local ngram; the model uses either words or lemmas, but never both.

[4] The weight W_j depends on the type of the feature f_j: for the bag-of-word features, this weight is inversely proportional to the distance between the target word and the feature, while for extended ngram features it is a empirically estimated weight (same value used in a similar English sense classifier).

[5] As identified by the POS tagger.

[6] Of the 40248 synsets present in the ItalWordNet, only 616 were provided .

Relation	Number of relations
hypernym/hyponym	4126
meronym/holonym	106
cause	70
antonym	64
other	576
Total relations	4942

Table 2: ItalWordNet statistics for the provided subset

Of the 83 words that are part of the evaluation, 82 of them had entries in the ItalWordNet; one word, *bello*, had no direct entry, but there were entries related to this particular word in the other entries, and we used those as inductive bias in the classification. Table 2 shows the number of different relations present in the selected ItalWordNet.

Intuitively, some of the WordNet relations are more useful than others for sense disambiguation. For instance, the *synonymy* and *near-synonymy* relations are more relevant than the *role_instrument* relation. To address this problem, each relation is assigned a intuitively-motivated weight[7]; each relation influences the overall behavior of the algorithm proportionally to its weight.

3.2 Relations to the English WordNet

In addition to relations among Italian words, the ItalWordNet contains links to the English WordNet senses of the corresponding translations (if any exist). In some cases, direct translations are not present, but a relation to a English WordNet sense is present (such as *eq_has_hyponym* or *eq_generalization*). These resources provide access to an independent information source – the distributional frequency of these sense as found in the English WordNet (which is present in version 1.7) (Miller, 1995). This information is used to obtain a second word sense classifier, used in system combination (as presented in Section 5).

Since the English senses in ItalWordNet are the senses in WordNet 1.5, we used the sense mappings $wn1.5 \rightarrow wn1.6 \rightarrow wn1.7$, as described in Daudé et al. (1999)[8].

[7]Since no training data was available, the weights could not be adjusted to minimize error rate. An alternative would be to estimate the weights on known classifications, e.g. English, and assume that they are language independent.

[8]The mappings were obtained from *http://www.lsi.upc.es/~nlp/tools/mapping.html*

4 The Algorithm

At a high level, the proposed algorithm for disambiguating a word v consists of first identifying words w that are similar in sense with word v, and selecting contexts of 2-3 sentences containing words the words w (including contexts containing the word v itself), creating sense centroids using these contexts, and bootstrapping a K-means clustering algorithm with the initial seeds.

The algorithmic framework used in this research is based on the following assumption:

Assumption 1 *If a word* u *that has a sense* s_u *similar to the sense* s_v *of word* v *(as identified by a relation in ItalWordNet between* s_u *and* s_v*), then any context containing word* u *is indicative of sense* s_v*.*

Assumption 1 can yield poor results in cases where two senses of word w are associated with different senses of the same word u (such as *press* and *suit*); in this case, sentences corresponding to word u will contribute to both senses of word w associated with u. The hope in this case is that all the words participating in disambiguation will cooperate in increasing the likelihood of the correct sense, and the effect of ambiguous examples will constitute white noise in the mass of relevant distinctions. If the noise is actually biased, the algorithm may fail to identify the correct sense.

4.1 Identifying Relevant Contexts

From an engineering point of view, the ItalWordNet is considered to be a set of relations $\mathcal{W}$ defined on the set of word-sense pairs. Formally, to identify the degree to which two senses are related, we construct a weighted multigraph $G_W = (V, E)$, where the set of vertices V is defined as

$$V = \{(v, s) \,|s \text{ is a sense of } v\}$$

and the set of edges is defined as

$$E = \{((u, s_u), (v, s_v)) \,|\exists r \in \mathcal{W} \; s.t. \; ((u, s_u), (v, s_v)) \in r\}$$

The weight associated with an edge, $w_G(e)$ is the weight of the relation associated with the edge; we will interpret these weights as distances rather than similarities (smaller weights indicate more similar senses).

To identify the set of words that are related in meaning with an ambiguous word w, we start

from the senses corresponding to word w, $\mathcal{L}_0 = \{(w, s_1), \ldots, (w, s_n)\}$, and we then expand the set $\mathcal{L}$ in the graph G_W as follows

$$\mathcal{L}_{k+1} = \mathcal{L}_k \cup \{l | \exists l' \in \mathcal{L}_k, r \in \mathcal{W}\, s.t.\ (l, l') \in r\} \quad (1)$$

Intuitively, we expand the set of words that are related to the ambiguous word w by examining the relations r in which its senses are involved, after which we expand the newly obtained senses, and so on. The relationships are expanded by examining the *sense* of each node, but the final output will extract the *words* associated with those senses[9]. Once the final set $\mathcal{L}_K$ is computed, the relevance of each word in it is computed by

$$w(l) = \min_{p \text{ path from } \mathcal{L}_0 \text{ to } l} w_G(p) \quad (2)$$

and $w_G(l_0, \ldots, l_n) = \sum_i w_G(l_i, l_{i+1})$.

The next step in the algorithm is to extract contexts c_l associated with each word l in $\mathcal{L}_K$ – for this purpose, we used a corpus of clean Italian newspaper text (extracted from *Corriere Della Sera,* 1993). After expanding the set $\mathcal{L}_1$ (5543 words; initially, there are 83 ambiguous words), the selected contexts formed a corpus of approximately 700M words[10] .

4.2 Automatic Sense Clustering

Algorithm 1 presents the proposed K-means-style clustering procedure, consisting of two major parts: computing the initial sense centroids, and the the application of K-means clustering.

The initial centroids are computed by seeding given them by the contexts c_l; each such context has a similarity to a particular centroid, inherited from the word that induced the context.

In the following step, the test documents – including the contexts c_l containing the ambiguous word – are assigned to the sense centroids (equation (4)), by computing the similarity between their corresponding vectors and the sense centroid vectors. There are several choices for the similarity measure; the one displayed in equation (4) is computing the similarity as

$$sim(c_l, \overline{s}_i) = P(\overline{s}_i | c_l) = \frac{P(\overline{s}_i)\, P(c_l | \overline{s}_i)}{P(c_l)} \quad (7)$$

[9]Unfortunately, the version of Italian WordNet we had access to has only a small subset of the relations, so we were forced to stop at $k = 1$.

[10]Documents appear in several word lists and the numbers include punctuation.

Algorithm 1 K-means-style WSD

1. Input: the ambiguous word w.
2. Create the extended set of lemmas $\mathcal{L}_K$ as described in equation (1).
3. For each lemma $l \in \mathcal{L}_K$, select contexts c_l surrounding l from a large unlabeled corpus.
4. Assign the contexts c_l to centroids corresponding to the senses of word w : $s_1 \ldots s_N$:
$$\overline{s}_i = \sum_{l, r \text{ s.t. } (s_i, l) \in r} w(r) \cdot c_l \quad (3)$$
5. Compute the similarity of each context c_t (corresponding to the test samples) to the centroid $\overline{s}_i$. For example:
$$sim(c_t, \overline{s}_i) = \frac{P(\overline{s}_i) \prod_{w \in c_t} P(w | \overline{s}_i)}{\sum_{j=1}^{n} P(\overline{s}_j) \prod_{w \in c_t} P(w | \overline{s}_j)} \quad (4)$$
6. Assign all the test centroids c_t to senses $\overline{s}_i$, based on the similarity between the centroids c_t and $\overline{s}_i$:
$$\overline{s}_i = \sum_{c_t \text{ test context}} sim(c_t, \overline{s}_i) \cdot c_t \quad (5)$$
7. Repeat from step 6 until convergence or a desired number of iterations is reached.
8. Classify each test document t with the sense corresponding to the closest centroid
$$\hat{s}(t) = \arg \max_{i=1 \ldots n} sim(t, S_i) \quad (6)$$

and makes use of the naïve Bayes assumption that the words in document c_l are independent given the sense, yielding

$$P(\overline{s}_i | c_l) \cong \frac{P(\overline{s}_i) \prod_{w \in c_t} P(w | \overline{s}_i)}{\sum_{j=1}^{n} P(\overline{s}_j) \prod_{w \in c_t} P(w | \overline{s}_j)} \quad (8)$$

Other possible choices for the similarity $sim(c_l, \overline{s}_i)$ include Bayes ratio (Gale et al., 1992)

$$P(\overline{s}_i | c_l) = \frac{P(\overline{s}_i)}{P(\neg \overline{s}_i)} \prod_{w \in c_l} \frac{P(w | \overline{s}_i)}{P(w | \neg \overline{s}_i)} \quad (9)$$

and cosine similarity

$$P(\overline{s}_i | c_l) = \frac{\langle \overline{s}_i, c_l \rangle}{\|\overline{s}_i\|_2 \, \|c_l\|_2} \quad (10)$$

Once the similarities $P(\overline{s}_i | c_l)$ have been computed, the centroids $(\overline{s}_i)_i$ can be updated as in

equation (5) – the centroid assignment shown in (10) is a soft one – each document will contribute to every centroid, with a ratio corresponding to their similarity. An alternative is to use hard document assignment

$$\overline{s}_i = \sum_{c_t} \delta\left(\arg\max_j sim\left(c_l, \overline{s}_j\right), i\right) \cdot c_t \tag{11}$$

where each document will contribute only to the centroid closest to it; δ is the Kronecker symbol:

$$\delta(x, y) = \begin{cases} 0 & \text{if } x \neq y \\ 1 & \text{if } x = y \end{cases}$$

5 Combining Information Sources

It is quite useful for a classification task to have access to multiple information sources; it follows from a standard argument that the information one has about a process (as measured by the entropy of the process) can only decrease if one obtains additional information:

$$H(A|B) \leq H(A)$$

Several studies in the machine learning community have shown that combining the information obtained from several classifiers not only results in improved performance, but also improves robustness. Even in cases where one has access to several structurally different information sources that are not easily integrable (Stevenson and Wilks, 2001), it might be more beneficial to construct separate classifiers for each type of context and combine their outputs, than construct a more complex classifier that tries to handle the combination internally.

Aside from the output of the classifier described in Section 4, we have access to two other information sources:

- the English distributional usage of the translation of a particular Italian word sense;
- the output of another word sense disambiguation system (Magnini et al., 2001) (downloadable from the SENSEVAL2 web site).

In using the English distributional data, we make the following assumption:

Assumption 2 *If an Italian word sense s_I has an English translation in sense s_E, then the usage of sense s_E in English is characteristic of the usage of sense s_I in Italian.*

By using this assumption, we obtain another classifier, as depicted in Algorithm 2. Even though this classifier is relatively simple, it obtains reasonably good results, as we will see in the experimental section.

Algorithm 2 English Most-Likely Classifier

1. For each sense s_i of ambiguous word v
$$count_{It}(s_i) = \sum_{e_j \text{ translation of } s_i} count_{En}(e_j)$$
2. Compute the sense probability
$$P_v(s_i) = \frac{count_{It}(s_i)}{\sum_{s_j} count_{It}(s_j)}$$
3. For a test word t, return
$$\hat{s} = \arg\max_s P_t(s)$$

Given N classifiers (possibly having probabilistic output), an easy and effective way of combining their output is through voting (Brill and Wu, 1998; van Halteren et al., 1998; Yarowsky et al., 2001), by computing the output classification as

$$\hat{s} = \arg\max_s \sum_i \delta\left(s, \hat{s}_i(d)\right) P(s|d) \tag{12}$$

where $\hat{s}_i(d) = \arg\max_s P_i(s|d)$. In other words, each classifier votes for the sense which it considers most likely, weighted by the probability of that sense. In the end, the sense that has been voted the most wins[11].

6 Experimental Evaluation

The test data in the Italian lexical-sample task consists of 3889 contexts of 1 to 3 sentences, for 83 ambiguous words.

6.1 Influence of Morphological Analysis on Performance

To investigate the impact of using the morphological analyzer, we created a second set $\mathcal{L}_k^{'}$ derived from an unlemmatized corpus. This means that we do not include documents which are associated with inflections of words related by ItalWordNet, and that the clustering algorithm is run on this unlemmatized data set.

[11] Ties are broken in favor of the sense which appears first in the ItalWordNet; this strategy proved to perform the best on the other SENSEVAL2 tasks we experimented on (only then the order was lexicographical).

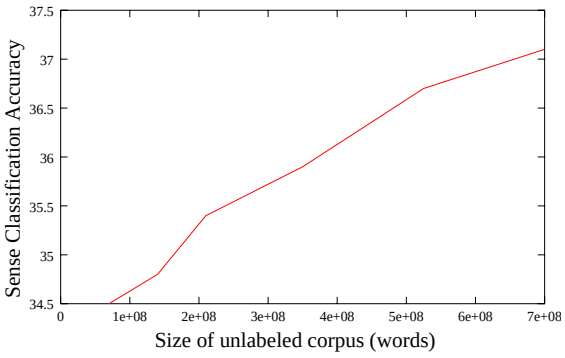

Figure 1: Sense classification accuracy versus initial unlabeled corpus size

	System	Accuracy	
		Fine	Coarse
1	Random Choice	25.3%	-
2	w/o Morphology	34.7%	41.8%
3	JHU01	35.3%	42.3%
4	k-Means System	37.1%	44.0%
5	Magnini01	39.0%	46.3%
6	Italian Most Likely	40.8%	45.3%
7	English Most Likely	38.9%	45.3%
8	I-ML(5) + E-ML(6)	46.4%	51.9%
9	Final system	**47.2%**	**53.7%**
10	Oracle Most Likely	65.4%	69.3%
11	Oracle Voting	73.4%	77.0%

Table 3: Sense classification accuracy for different variations of the system

As presented in Table 3, line 2, there is a significant decrease in performance when the morphological analyzer was omitted. Using the same initial corpus, approximately 75% of the originally extracted documents were selected[12]; however, our algorithm on the set $\mathcal{L}_k^{'}$ achieves only 34.6% accuracy, significantly lower than the results obtained by the full system on 75% of the data, 36.6% (as shown in Figure 1).

The unlemmatized system's performance is most comparable to the performance achieved when using only 10% of the corpus. In other words, it takes a corpus 10 times larger in order to compensate for the inability to perform morphological analysis.

6.2 Classification Results

Figure 1 presents the results obtained by the algorithm presented in Section 4, for varying sizes of the unlabeled corpus. The performance increases from 34.5% at 70M words to 37.1% at 700M words (the difference in performance is statistically significant at a confidence level of 10^{-3}).

The experiments we ran to test the performance of the classifier are presented in Table 3. There are many baselines against which one can measure performance. Line 1 of Table 3 shows the estimated performance of a system that chooses a sense at random. In line 2, the performance of the system is evaluated by running the system without lemmatization, as discussed in Section 6.1. Line 4 presents the performance of the K-means system presented in Section 4, and line 3 presents the performance of the original JHU system. The difference between the 2 consists mainly in the size of unlabeled corpus and quality of lemmatization.

Perhaps the most interesting result in Table 3 is the fact that the estimation of the most likely sense for a given word can be done more robustly than the estimation of individual senses. By making the system return the most likely sense (of the senses output by the clustering algorithm) for a given word, the performance increases by nearly 4% (line 6 in the table), yielding the best individual system results.

The classifier based on the sense distributions on the English WordNet, described in Algorithm 2, yields good performance (line 7), comparable with the best SENSEVAL2 system (line 5).

By combining the two most likely systems (the one obtained on the Italian data by using K-means clustering and the one obtained from the English WordNet), one can obtain an impressive performance of 46.4%, and adding the best competing system from the Italian SENSEVAL2 exercise, we obtain a performance of 47.2%. The difference in performance between the last two systems is not, however, statistically significant at a confidence level of 0.05.

As a final observation on the system performance, it is interesting to remark that there is still a long way to go before obtaining results that are competitive with the most-likely *oracle* performance, listed in the table on line 10. This oracle returns for each sample the "true" most-likely sense (computed on the test data); it obtains a performance of 65.3%, substantially better than the other results obtained on this data. This oracle is outperformed by a *voting* oracle, which returns the correct sense if at least

[12]The selection is done originally by using lemmas rather than words; when using words, fewer contexts will be selected.

one classifier predicted it, (line 11 of Table 3)[13].

7 Conclusion

In conclusion, we have presented a novel method of word sense classification by using large amounts of unlabeled data, word semantic relations both in the target and a second language. The procedure integrates these knowledge sources to provide a more robust estimation. The performance obtained, while still lower than the true most likely classification, substantially outperforms previously published results on this data set.

As future work, we plan to integrate more sophisticated syntactic knowledge/features into the model, to develop a better weighting scheme of individual semantic relations by training on labeled text (in another language, e.g. English) and also to improve the balance of the per sense training samples.

8 Acknowledgements

The authors would like to thank David Yarowsky for his useful comments and support, Gideon Mann for his helpful comments on an early draft of this paper, the Johns Hopkins University NLP lab and JHU SENSEVAL2 team for a creating a stimulating research environment, to Paola Virga for doing that annoying annotation, and to the anonymous reviewers for their helpful comments and suggestions, especially in identifying a mismatch between the English WordNet versions. This work was partially supported by NSF grant IIS-9985033 and ONR/MURI contract N00014-01-1-0685.

References

E. Brill and J. Wu. 1998. Classifier combination for improved lexical disambiguation. In *Proceedings of COLING-ACL'98*, pages 191–195.

J. Daudé, L. Padró, and G. Rigau. 1999. Experiments on applying relaxation labeling to map multilingual hierarchies. Technical Report LSI-99-5-R, Software Department. UPC.

W. Gale, K. Church, and D. Yarowsky. 1992. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26:415–439.

B. Magnini, C. Strapparava, G. Pezzulo, and A. Gliozzo. 2001. Using domain information for word sense disambiguation. In *Proceedings of SENSEVAL2*.

[13]The voting oracle performance effectively constitutes an upper bound on a voting system's performance.

D. McCarthy, J. Carroll, and J. Preiss. 2001. Disambiguating noun and verb senses using automatically acquired selectional preferences. In *Proceedings of SENSEVAL2*.

R. Mihalcea and D. Moldovan. 1999. An automatic method for generating sense tagged corpora. In *Proceedings of AAAI '99*, pages 461–466.

G. A. Miller. 1995. WordNet: A lexical database. *Communications of the ACM*, 38(11).

T. Pedersen and R. Bruce. 1998. Knowledge lean word sense disambiguation. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 800–805.

P. Resnik. 1997. Selectional preference and sense disambiguation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics*, pages 52–57.

G. Rigau, J. Atserias, and E. Agirre. 1997. Combining unsupervised lexical knowledge methods for word sense disambiguation. In *Proceedings of ACL '97*, pages 48–55.

A. Roventini, A. Alonge, F. Bertagna, B. Magnini, and N. Calzolari. 2000. ItalWordNet: a large semantic database for Italian. In *Proceedings of LREC-2000*, pages 783–790.

H. Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*.

H. Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124.

M. Stevenson and Y. Wilks. 2001. The interaction of knowledge sources in word sense disambiguation. *Computational Linguistics*, 27(3):321–349.

M. Sussna. 1993. Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of CIKM '93*.

H. van Halteren, J. Zavrel, and W. Daelemans. 1998. Improving data driven wordclass tagging by system combination. In *Proceedings of COLING-ACL'98*, pages 491–497.

E. Voorhees. 1993. Using WordNet to disambiguate word senses for text retrieval. In *Proceedings of SIGIR '93*, pages 171–180.

D. Yarowsky and R. Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of ACL-2000*, pages 207–216.

D. Yarowsky, S. Cucerzan, R. Florian, C. Schafer, and R. Wicentowski. 2001. The Johns Hopkins SENSEVAL2 system descriptions. In *Proceedings of SENSEVAL2*.

D. Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196.