A Maximum Entropy-based Word Sense Disambiguation system[*]

Armando Suárez **Manuel Palomar**
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante
Apartado de correos, 99
E-03080 Alicante, Spain
{armando, mpalomar}@dlsi.ua.es
http://www.dlsi.ua.es/~armando/publicaciones.html

Abstract

In this paper, a supervised learning system of word sense disambiguation is presented. It is based on *conditional maximum entropy models*. This system acquires the linguistic knowledge from an annotated corpus and this knowledge is represented in the form of features. Several types of features have been analyzed using the SENSEVAL-2 data for the Spanish lexical sample task. Such analysis shows that instead of training with the same kind of information for all words, each one is more effectively learned using a different set of features. This *best-feature-selection* is used to build some systems based on different maximum entropy classifiers, and a voting system helped by a knowledge-based method.

1 Introduction

Word sense disambiguation (WSD) is an open research field in natural language processing (NLP). The task of WSD consists in assigning the correct sense to words using an electronic dictionary as the source of word definitions. This is a hard problem that is receiving a great deal of attention from the research community.

Currently, there are two main methodological approaches in this research area: *knowledge-based* methods and *corpus-based* methods. The former approach relies on previously acquired linguistic knowledge, and the latter uses techniques from statistics and machine learning to induce models of language usage from large samples of text (Pedersen, 2001). Learning can be supervised or unsupervised. With supervised learning, the actual status (here, sense label) for each piece of data in the training example is known, whereas with unsupervised learning the classification of the data in the training example is not known (Manning and Schütze, 1999).

At SENSEVAL-2, researchers showed the latest contributions to WSD. Some supervised systems competed in the Spanish lexical sample task. The Johns Hopkins University system (Yarowsky et al., 2001) combined, by means of a voting-based classifier, several WSD subsystems based on different methods: decision lists (Yarowsky, 2000), cosine-based vector models, and Bayesian classifiers. The University of Maryland system (UMD-SST) (Cabezas et al., 2001) used support vector machines.

Pedersen (2002) proposes a baseline methodology for WSD based on decision tree learning and naive Bayesian classifiers, using simple lexical features. Several systems that combine different classifiers using distinct sets of features competed at SENSEVAL-2, both in the English and Spanish lexical sample tasks.

This paper presents a system that implements a corpus-based method of WSD. The method used to perform the learning over a set of sense-disambiguated examples is that of maximum entropy (ME) probability models. Linguistic information is represented in the form of feature vectors, which identify the occurrence of certain attributes that appear in contexts containing linguistic ambiguities. The context is the text surrounding an ambiguity that is relevant to the disambiguation process. The features used may be of a distinct nature: word collocations, part-of-speech (POS) labels, keywords, topic and domain information, grammatical relationships, and so on. Instead of training with the same kind of information for all words, which underestimates which information is more relevant to

[*] This paper has been partially supported by the Spanish Government (CICYT) under project number TIC2000-0664-C02-02.

each word, our research shows that each word is more effectively learned using a different set of features. Therefore, a more accurate feature selection can be done testing several combinations of features by means of a n-fold cross-validation over the training data.

At SENSEVAL-2, Stanford University presented a metalearner (Ilhan et al., 2001) combining simple classifiers (naive-Bayes, vector space, memory-based and others) that use voting and conditional ME models. García Varea et al. (2001) do machine translation tasks using ME to perform some kind of semantic classification, but they also rely on another statistical training procedure to define word classes.

In the following discussion, the ME framework will be described. Then, feature implementation and the complete set of feature definitions used in this work will be detailed. Next, evaluation results using several combinations of these features will be shown. Finally, some conclusions will be presented, along with a brief discussion of work in progress and future work planned.

2 The Maximum Entropy Framework

ME modeling provides a framework for integrating information for classification from many heterogeneous information sources (Manning and Schütze, 1999). ME probability models have been successfully applied to some NLP tasks, such as POS tagging or sentence boundary detection (Ratnaparkhi, 1998).

The WSD method used in this work is based on conditional ME models. It has been implemented using a supervised learning method that consists of building word-sense classifiers using a semantically tagged corpus. A classifier obtained by means of an ME technique consists of a set of parameters or coefficients which are estimated using an optimization procedure. Each coefficient is associated with one feature observed in the training data. The main purpose is to obtain the probability distribution that maximizes the entropy, that is, maximum ignorance is assumed and nothing apart from the training data is considered. Some advantages of using the ME framework are that even knowledge-poor features may be applied accurately; the ME framework thus allows a virtually unrestricted ability to represent problem-specific knowledge in the form of features (Ratnaparkhi, 1998).

Let us assume a set of contexts X and a set of classes C. The function $cl : X \rightarrow C$ chooses the class c with the highest conditional probability in the context x: $cl(x) = \arg\max_c p(c|x)$. Each feature is calculated by a function that is associated to a specific class c', and it takes the form of equation (1), where $cp(x)$ is some observable characteristic in the context[1]. The conditional probability $p(c|x)$ is defined by equation (2), where α_i is the parameter or weight of the feature i, K is the number of features defined, and $Z(x)$ is a value to ensure that the sum of all conditional probabilities for this context is equal to 1.

$$f(x,c) = \begin{cases} 1 & \text{if } c' = c \text{ and } cp(x) = true \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$p(c|x) = \frac{1}{Z(x)} \prod_{i=1}^{K} \alpha_i^{f_i(x,c)} \tag{2}$$

The implementation of this ME framework for WSD was done in *C++* and included the learning module, the classification module, the evaluation module, and the corpus translation module. The first two modules comprise the main components.

The learning module produces the classifiers for each word using a corpus that is syntactically and semantically annotated. This module has two subsystems. The first subsystem consists of two component actions: in a first step, the module processes the learning corpus in order to define the functions that will apprise the linguistic features of each context; in a second step, the module then fills in the feature vectors. The second subsystem of the learning module performs the estimation of the coefficients and stores the classification functions. For example, let us assume that we want to build a classifier for noun *interest* and that POS label of the previous word is the type of feature to use and the training corpus has these three samples:

[1] The ME approach is not limited to binary features, but the optimization procedure used for the estimation of the parameters, the *Generalized Iterative Scaling* procedure, uses this kind of functions.

... the widespread **interest#1** in the ...
... the best **interest#5** of both ...
... persons expressing **interest#1** in the ...

The learning module performs a sequential processing of this corpus looking for pairs <*POS-label, sense*>. Then, <*adjective,#1*>, <*adjective,#5*>, and <*noun,#1*> are used to define three functions (each context have a vector of three features). Next, each vector is filled in with the result of the evaluation of each function. Finally, the optimization procedure calculates the coefficients and the output is a classifier for the word *interest.*

The classification module carries out the disambiguation of new contexts using the previously stored classification functions. When ME does not have enough information about a specific context, several senses may achieve the same maximum probability and thus the classification cannot be done properly. In these cases, the most frequent sense in the corpus is assigned. However, this heuristic is only necessary for minimum number of contexts or when the set of linguistic attributes processed is very small.

3 Feature Implementation

An important issue in the implementation of this ME framework is the form of the functions that calculate each feature. These functions are defined in the training phase and depend upon the data in the corpus.

A usual definition of features would substitute $cp(x)$ in equation (1) with an expression like *info(x,i)* = *a*, where *info(x,i)* informs about a property that can be found at position i in a context x, and a is a predefined value. For example, if we consider that 0 is the position of the word to be learned and that i is related to 0, then *POS(x,-1)* = *'adjective'*. Therefore, equation (1) is used to generate a function for each possible value $(sense, a)$ at position i. Henceforth, we will refer to this type of features as "non-relaxed" features, against "relaxed" features described below. In the example of the previous section, three "non-relaxed" functions could be defined.

Other expressions, such as *info(x,i)* $\in W_{(c',i)}$, may be substituted for the term $cp(x)$ as a way to reduce the number of possible features. In the expression above, for example, $W_{(c',i)}$ is the set of attributes present in the learning examples at position i. Again, if we assume that $POS(x,-1)$, then for each sense of the ambiguous word, the system builds a set with the POS tags occurring in their previous positions. So this kind of function reduces the number of features to one per each sense at position i. In the example of the previous section, two "relaxed" functions could be defined from <{*adjective,noun*}*,#1*> and <*adjective,#5*>.

Due to the nature of the disambiguation task, the number of times that a feature generated by the first type of function ("non-relaxed") is activated is very low, and feature vectors have a large number of values equal to 0. The new function drastically reduces the number of features, with a minimum of degradation in evaluation results. At the same time, new features can be incorporated into the learning process with a minimum impact on efficiency.

4 Description of Features

The set of features defined for the training of the system is described in figure 1, and is based on the feature selection made by Ng and Lee (1996) and Escudero et al. (2000). Features are automatically defined as explained before and depend on the data in the training corpus. These features are based on words, collocations, and POS tags in the local context. Both "relaxed" and "non-relaxed" functions are used.

Figure 1: List of types of features

- **0**: ambiguous-word shape
- ***s***: words at positions ±1, ±2, ±3
- ***p***: POS-tags of words at positions ±1, ±2, ±3
- ***b***: lemmas of collocations at positions $(-2,-1)$, $(-1,+1)$, $(+1,+2)$
- ***c***: collocations at positions $(-2,-1)$, $(-1,+1)$, $(+1,+2)$
- ***k***m: lemmas of nouns at any position in context, occurring at least $m\%$ times with a sense
- ***r***: grammatical relation of the ambiguous word
- ***d***: the word that the ambiguous word depends on
- ***L***: lemmas of content-words at positions ±1, ±2, ±3 ("relaxed" definition)
- ***W***: content-words at positions ±1, ±2, ±3 ("relaxed" definition)
- ***S, B, C, P,* and *D***: "relaxed" versions

Actually, each item in figure 1 groups several sets of features. The majority of them depend on nearest words (for example, s comprises all possible features defined by the words occurring in each sample at positions w_{-3}, w_{-2}, w_{-1}, w_{+1}, w_{+2}, w_{+3} related to the ambiguous word). Types nominated with capital letters are based on the "relaxed" function form, that is, these features consists of a simply recognition of an attribute as belonging to the training data.

Keyword features (km) are vaguely inspired by Ng and Lee (1996). A nouns selection is done using frequency information for nouns co-occurring with a particular sense. For example, in a set of 100 examples of sense four of the noun *interest*, if the noun *bank* is found ten times or more (m = 10%), then a feature is defined for each possible sense of *interest*.

Moreover, new features have also been defined using other grammatical properties: relationship features (r) that refer to the grammatical relationship of the ambiguous word (*subject*, *object*, *complement*, ...) and dependency features (d and D) extract the word related to the ambiguous one through the dependency parse tree.

5 Evaluation

In this section we present the results of our evaluation over the training and test data of the SENSEVAL-2 Spanish lexical sample task. This corpus was parsed using *Conexor Functional Dependency Grammar parser for Spanish* (Tapanainen and Järvinen, 1997).

Table 1 shows the five best results using several sets of features. The classifiers were built from the training data and evaluated over the test data. These values mean the maximum accuracy that the system can achieve at this moment with a fixed set of features for all words. Nevertheless, there are clear differences between nouns, verbs and adjectives.

Table 1: Evaluation on SENSEVAL-2 Spanish data

ALL		Nouns	
0.671	0LWSBCk5	0.683	LWSBCk5
0.666	LWSBCk5	0.682	0LWSBCk5
0.663	sbcpdk5	0.666	0LWSBCPDk5
0.662	0LWSBCPDk5	0.666	0LWsBCPDk5
0.662	0LWsBCPDk5	0.666	0LWSBCPDk5
Verbs		Adjectives	
0.595	sk5	0.783	LWsBCp
0.584	sbcprdk3	0.778	0sprd
0.583	sbcpdk5	0.777	0sbcprdk5
0.580	sbcpk5	0.777	0sbcprdk10
0.580	0sbcprdk3	0.772	0spdk5

Table 2: 3-fold cross-validation results on SENSEVAL-2 Spanish training data

	Features	Functions	Accur	MFS
autoridad,N	sbcp	548	0.589	0.503
bomba,N	0LWSBCk5	176	0.762	0.707
canal,N	sbcprdk3	1258	0.579	0.307
circuito,N	0LWSBCk5	482	0.536	0.392
corazón,N	0Sbcpk5	210	0.781	0.607
corona,N	sbcp	420	0.722	0.489
gracia,N	0sk5	542	0.634	0.295
grano,N	0LWSBCr	102	0.681	0.483
hermano,N	0Sprd	152	0.731	0.602
masa,N	LWSBCk5	206	0.756	0.455
naturaleza,N	sbcprdk3	1213	0.527	0.424
operación,N	0LWSBCk5	399	0.543	0.377
órgano,N	0LWSBCPDk5	271	0.715	0.515
partido,N	0LWSBCk5	111	0.839	0.524
pasaje,N	sk5	389	0.685	0.451
programa,N	0LWSBCr	137	0.587	0.486
tabla,N	sk5	282	0.663	0.488
actuar,V	sk5	772	0.514	0.293
apoyar,V	0sbcprdk3	1257	0.730	0.635
apuntar,V	0LWsBCPDk5	729	0.661	0.478
clavar,V	sbcprdk3	1026	0.561	0.449
conducir,V	LWsBCPD	482	0.534	0.358
copiar,V	0sbcprdk3	1231	0.457	0.338
coronar,V	sk5	739	0.698	0.327
explotar,V	0LWSBCk5	643	0.593	0.318
saltar,V	LWsBC	518	0.403	0.132
tocar,V	0sbcprdk3	1888	0.583	0.313
tratar,V	sbcpk5	1421	0.527	0.208
usar,V	0Sprd	222	0.732	0.669
vencer,V	sbcprdk3	1063	0.696	0.618
brillante,A	sbcprdk3	1199	0.756	0.512
ciego,A	0spdk5	478	0.812	0.565
claro,A	0Sprd	177	0.919	0.854
local,A	0LWSBCr	64	0.798	0.750
natural,A	sbcprdk10	949	0.471	0.267
popular,A	sbcprdk10	2624	0.865	0.632
simple,A	LWsBCPD	522	0.776	0.621
verde,A	LWSBCk5	556	0.601	0.317
vital,A	Sbcp	591	0.774	0.441

Our main goal is to find a method to automatically obtain the best feature selection from the training data. Such method consists of a n-fold cross-validation testing several combinations of features over the training data and the analysis of the results obtained for each word.

Table 2 shows the best results obtained using a 3-fold cross-validation evaluation method on training data. Several feature combinations have been tested in order to find the best set for each selected word. The purpose was to achieve the most relevant information for each word from the corpus rather than applying the same combination of features to all of them. Therefore, column *Features* is the feature selection with the best result. Strings in each row represent the whole set of features used in the training of each classifier. For example, *autoridad* obtains its best result using nearest words, collocations of two lemmas, collocations of two words, and POS information; s, b, c and p features respectively (see figure 1). *Functions* is the number of functions generated from features, and *Accur* (for "accuracy") the number of correctly classified contexts divided by the total number of contexts. Column *MFS* is the accuracy obtained when the most frequent sense is selected.

In order to perform the three tests on each word, some preprocessing of the corpus was done. For each word, all senses were uniformly distributed in the three folds (each fold contains one third of examples of each sense). Those senses that had fewer than three examples in the original corpus file were rejected and not processed.

The data summarized in Table 2 reveal that utilization of "relaxed" features in the ME method is useful; both "relaxed" and "non-relaxed" functions are used, even for the same word. For example, adjective *vital* obtains the best result with "Sbcp" (the "relaxed" version of words in a window $(-3..+3)$, collocations of two lemmas and two words in a window $(-2..+2)$, and POS labels, in a window $(-3..+3)$ too); we can assume that single word information is less important than collocations in order to disambiguate *vital* correctly.

Ambiguous word shape (0 features) is useful for nouns, verbs and adjectives, but many of the words do not use it for its best feature selection. In general, these words have not a relevant relationship between shape and senses. On the other hand, POS information (p and P features) is selected less often. When comparing *lemma* features against *word* features (e.g., L versus W, and B versus C), they are complementary in the majority of cases. Grammatical relationships (r features) and word-word dependencies (d and D features) seem very useful too if combined with other types of attributes. Moreover, keywords (km features) are used very often, possibly due to the source and size of contexts of SENSEVAL-2 data.

Table 3: Best feature selections per POS

ALL		Nouns	
0.613	sbcprdk3	0.609	LWSBCk5
0.609	sbcprdk5	0.609	sbcprdk5
0.605	0sbcprdk3	0.609	sbcprdk3
0.604	sbcprdk10	0.608	sk5
0.602	sbcpdk5	0.602	0sbcprdk3
Verbs		Adjectives	
0.575	sbcprdk3	0.706	0spdk5
0.568	sbcpdk5	0.701	0sbcprdk10
0.567	sbcprdk5	0.699	sbcprdk10
0.567	sbcpk5	0.699	0sbcprdk5
0.560	sbcprdk10	0.696	LWsBCp

Table 3 shows the first five best feature selections for all words, and for nouns, verbs, and adjectives. Data in this table and in table 2 were used to build four different sets of classifiers in order to compare their accuracy: **MEfix** uses the overall best feature selection for all words; **MEbfs** the best selection of features for

Table 4: Evaluation of ME systems

ALL		Nouns	
0.677	MEbfs.pos	0.683	MEbfs.pos
0.676	vME	0.678	vME
0.667	MEbfs	0.661	MEbfs
0.658	MEfix	0.646	MEfix
Verbs		Adjectives	
0.583	vME	0.774	vME
0.583	MEbfs.pos	0.772	MEbfs.pos
0.583	MEfix	0.771	MEbfs
0.580	MEbfs	0.756	MEfix

MEfix: *sbcprdk3* for all words
MEbfs: each word with its best feature selection
MEbfs.pos: *LWSBCk5* for nouns, *sbcprdk3* for verbs, and *0spdk5* for adjectives
vME: majority voting between MEfix, MEbfs.pos, and MEbfs

each word; **MEbfs.pos** uses the best selection of each POS for all words of that POS; finally, **vME** is a majority voting system that has as input the answers of the three systems.

Table 4 shows the comparison of these four systems, the less efficient is MEfix that applies the same set of types of features to all words. However, the best feature selection per word (MEbfs) is not the best, probably because deeper analysis and more training examples are necessary. The best choice seems to select a fixed set of types of features for each POS (MEbfs.pos). This last system obtains an accuracy slightly better than the best evaluation result in table 1, that is, a *best-feature-selection* strategy from training data guarantees a successful disambiguation.

In general, verbs are difficult to learn and the accuracy of the method for them too low; in our opinion, more information (knowledge-based perhaps) is needed to build their classifiers, but the types of features used could be unsuitable too. The voting system (vME), based on the agreement between the other three systems, does not improve the accuracy.

Finally, the results of the ME method were compared with the systems that competed at SENSEVAL-2 in the Spanish lexical sample task (table 5)[2][3]. If the ME systems described previously are ranked within this scoring table, nouns and adjectives obtain a excellent results; verbs obtain worse results.

Table 5 also includes an enrichment of vME: vME+SM. This new voting system adds another classifier, *specification marks* (Montoyo and Palomar, 2001), a knowledge-based method that uses the semantic relationships between words stored in WordNet and EuroWordNet (Vossen, 1998). Because it works merely with nouns, vME+SM improves accuracy for them only, but obtains the same score than JHU(R). Overall score reaches the second place.

6 Conclusions

A WSD system based on maximum entropy conditional probability models has been presented. It is a supervised learning method that needs a corpus previously annotated with sense labels.

Using the training data of SENSEVAL-2 for the Spanish lexical sample task, several combinations of features were analyzed in order to identify which were the best. This information is the basis of various sets of classifiers, as well as two majority voting systems. The results obtained by these systems show that selecting best feature sets guarantees the success of the disambiguation method.

As we work to improve the ME method with a deeper analysis of the feature selection strategy, we are also working to develop a cooperative strategy between several other methods as well, both knowledge-based and corpus-based.

Future research will incorporate domain information as an additional information source for the system. WordNet Domains (Magnini and Strapparava, 2000) is an enrichment of WordNet with domain labels. These attributes will be incorporated into the learning of the system in the same way that features were incorporated, as described above, except that domain disambiguation will be evaluated as well; that is, WordNet senses (*synsets*) will be substituted for domains labels, thereby reducing the number of possible classes into which contexts can be classified.

[2] Systems: JHU and JHU(R) by Johns Hopkins University; CSS244 by Stanford University; UMD-SST by University of Maryland; Duluth systems by University of Manitoba; UA by University of Alicante.

[3] SENSECAL-2 data can be downloaded from http://www.sle.sharp.co.uk/senseval2/

References

Clara Cabezas, Philip Resnik, and Jessica Stevens. 2001. Supervised Sense Tagging using Support Vector Machines. In Preiss and Yarowsky (Preiss and Yarowsky, 2001), pages 59–62.

Gerard Escudero, Lluis Màrquez, and German Rigau. 2000. Boosting applied to word sense disambiguation. In *Proceedings of the 12th Conference on Machine Learning ECML2000*, Barcelona, Spain.

Ismael García-Varea, Franz J. Och, Hermann Ney, and Francisco Casacuberta. 2001. Refined lexicon models for statistical machine translation using a maximum entropy approach. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, pages 204–211.

H. Tolga Ilhan, Sepandar D. Kamvar, Dan Klein, Christopher D. Manning, and Kristina Toutanova. 2001. Combining Heterogeneus

Table 5: Comparing with SENSEVAL-2 systems

ALL		Nouns		Verbs		Adjectives	
0.713	jhu(R)	0.702	jhu(R)	0.643	jhu(R)	0.802	jhu(R)
0.684	**vME+SM**	0.702	**vME+SM**	0.609	jhu	0.774	**vME**
0.682	jhu	0.683	**MEbfs.pos**	0.595	css244	0.772	**MEbfs.pos**
0.677	**MEbfs.pos**	0.681	jhu	0.584	umd-sst	0.772	css244
0.676	**vME**	0.678	**vME**	0.583	**vME**	0.771	**MEbfs**
0.670	css244	0.661	**MEbfs**	0.583	**MEbfs.pos**	0.764	jhu
0.667	**MEbfs**	0.652	css244	0.583	**MEfix**	0.756	**MEfix**
0.658	**MEfix**	0.646	**MEfix**	0.580	**MEbfs**	0.725	duluth 8
0.627	umd-sst	0.621	duluth 8	0.515	duluth 10	0.712	duluth 10
0.617	duluth 8	0.612	duluth Z	0.513	duluth 8	0.706	duluth 7
0.610	duluth 10	0.611	duluth 10	0.511	ua	0.703	umd-sst
0.595	duluth Z	0.603	umd-sst	0.498	duluth 7	0.689	duluth 6
0.595	duluth 7	0.592	duluth 6	0.490	duluth Z	0.689	duluth Z
0.582	duluth 6	0.590	duluth 7	0.478	duluth X	0.687	ua
0.578	duluth X	0.586	duluth X	0.477	duluth 9	0.678	duluth X
0.560	duluth 9	0.557	duluth 9	0.474	duluth 6	0.655	duluth 9
0.548	ua	0.514	duluth Y	0.431	duluth Y	0.637	duluth Y
0.524	duluth Y	0.464	ua				

Classifiers for Word-Sense Disambiguation. In Preiss and Yarowsky (Preiss and Yarowsky, 2001), pages 87–90.

Bernardo Magnini and C. Strapparava. 2000. Experiments in Word Domain Disambiguation for Parallel Texts. In *Proceedings of the ACL Workshop on Word Senses and Multilinguality*, Hong Kong, China.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing.* The MIT Press, Cambridge, Massachusetts.

Andrés Montoyo and Manuel Palomar. 2001. Specification Marks for Word Sense Disambiguation: New Development. In Alexander F. Gelbukh, editor, *CICLing*, volume 2004 of *Lecture Notes in Computer Science*, pages 182–191. Springer.

Hwee Tou Ng and Hian Beng Lee. 1996. Integrating multiple knowledge sources to disambiguate word senses: An exemplar-based approach. In Arivind Joshi and Martha Palmer, editors, *Proceedings of the 34th Annual Meeting of the ACL*, San Francisco. Morgan Kaufmann Publishers.

Ted Pedersen. 2001. A decision tree of bigrams is an accurate predictor of word sense. In *Proceedings of the 2nd Annual Meeting of the North American Chapter of the ACL*, pages 79–86, Pittsburgh, July.

Ted Pedersen. 2002. A baseline methodology for word sense disambiguation. In Alexander F. Gelbukh, editor, *CICLing*, volume 2276 of *Lecture Notes in Computer Science*, pages 126–135. Springer.

Judita Preiss and David Yarowsky, editors. 2001. *Proceedings of SENSEVAL-2*, Toulouse, France, July. ACL-SIGLEX.

Adwait Ratnaparkhi. 1998. *Maximum Entropy Models for Natural Language Ambiguity Resolution.* Ph.D. thesis, University of Pennsylvania.

Pasi Tapanainen and Timo Järvinen. 1997. A non-projective dependency parser. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 64–71, April.

Piek Vossen. 1998. EuroWordNet: Building a Multilingual Database with WordNets for European Languages. *The ELRA Newsletter*, 3(1).

David Yarowsky, Silviu Cucerzan, Radu Florian, Charles Schafer, and Richard Wicentowski. 2001. The Johns Hopkins SENSEVAL-2 System Description. In Preiss and Yarowsky (Preiss and Yarowsky, 2001), pages 163–166.

David Yarowsky. 2000. Hierarchical decision lists for word sense disambiguation. *Computers and the Humanities*, 34(2):179–186.