

# Survey of Word Sense Disambiguation Approaches

Xiaohua Zhou and Hyoil Han

College of Information Science & Technology, Drexel University  
3401 Chestnut Street, Philadelphia, PA 19104  
xiaohua.zhou@drexel.edu, hyoil.han@cis.drexel.edu

## Abstract

Word Sense Disambiguation (WSD) is an important but challenging technique in the area of natural language processing (NLP). Hundreds of WSD algorithms and systems are available, but less work has been done in regard to choosing the optimal WSD algorithms. This paper summarizes the various knowledge sources used for WSD and classifies existing WSD algorithms according to their techniques. The rationale, tasks, performance, knowledge sources used, computational complexity, assumptions, and suitable applications for each class of WSD algorithms are also discussed. This paper will provide users with general knowledge for choosing WSD algorithms for their specific applications or for further adaptation.

## 1. Introduction

Word Sense Disambiguation (WSD) refers to a task that automatically assigns a sense, selected from a set of pre-defined word senses to an instance of a polysemous word in a particular context. WSD is an important but challenging technique in the area of natural language processing (NLP). It is necessary for many real world applications such as machine translation (MT), semantic mapping (SM), semantic annotation (SA), and ontology learning (OL). It is also believed to be helpful in improving the performance of many applications such as information retrieval (IR), information extraction (IE), and speech recognition (SR).

The reasons that WSD is difficult lie in two aspects. First, dictionary-based word sense definitions are ambiguous. Even if trained linguists manually tag the word sense, the inter-agreement is not as high as would be expected (Ng 1999; Fellbaum and Palmer 2001). That is, different annotators may assign different senses to the same instance. Second, WSD involves much world knowledge or common sense, which is difficult to verbalize in dictionaries (Veronis 2000).

Sense knowledge can be represented by a vector, called a sense knowledge vector (*sense ID, features*), where features

can be either symbolic or empirical. Dictionaries provide the definition and partial lexical knowledge for each sense. However, dictionaries include little well-defined world knowledge (or common sense). An alternative is for a program to automatically learn world knowledge from manually sense-tagged examples, called a training corpus.

The word to be sense tagged always appears in a context. Context can be represented by a vector, called a context vector (*word, features*). Thus, we can disambiguate word sense by matching a sense knowledge vector and a context vector. The conceptual model for WSD is shown in figure 1.

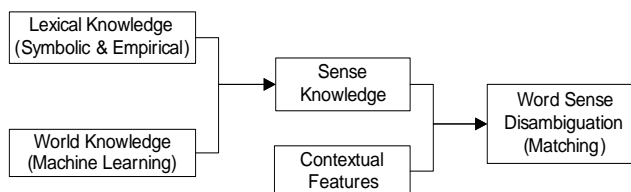


Figure 1. Conceptual Model for Word Sense Disambiguation

Apart from knowledge sources, we need to consider other issues such as performance, computing complexity, and tasks when choosing WSD algorithms. Precision and recall are two important measures of performance for WSD. Precision is defined as the proportion of correctly classified instances of those classified, while recall is the proportion of correctly classified instances of total instances. Thus, the value of recall is always less than that of precision unless all instances are sense tagged.

The remainder of the paper is organized as follows: in section 2, we summarize lexical knowledge and various contextual features used for WSD, while in section 3 we present the core component, which is the classification and evaluation of existing WSD algorithms. A short conclusion finishes the article.

## 2. Knowledge Sources

Knowledge sources used for WSD are either lexical knowledge released to the public, or world knowledge learned from a training corpus.

## 2.1 Lexical Knowledge

In this section, the components of lexical knowledge are discussed. Lexical knowledge is usually released with a dictionary. It is the foundation of unsupervised WSD approaches.

**Sense Frequency** is the usage frequency of each sense of a word. Interestingly, the performance of the naïve WSD algorithm, which simply assigns the most frequently used sense to the target, is not very bad. Thus, it often serves as the benchmark for the evaluation of other WSD algorithms.

**Sense glosses** provides a brief explanation of a word sense, usually including definitions and examples. By counting common words between the gloss and the context of the target word, we can naïvely tag the word sense.

**Concept Trees** represent the related concepts of the target in the form of semantic networks as is done by WordNet (Fellbaum 1998). The commonly used relationships include hypernym, hyponym, holonym, meronym, and synonym. Many WSD algorithms can be derived on the basis of concept similarity measured from the hierarchical concept tree.

**Selectional Restrictions** are the semantic restrictions placed on the word sense. LDOCE (Longman Dictionary of Contemporary English) senses provide this kind of information. For example, the first sense of *run* is usually constrained with human subject and an abstract thing as an object. Stevenson & Wilks (2001) illustrates how to use selectional restriction to deduct the suitable word sense.

**Subject Code** refers to the category to which one sense of the target word belongs. In LDOCE, primary pragmatic codes indicate the general topic of a text in which a sense is likely to be used. For example, LN means “Linguistic and Grammar” and this code is assigned to some senses of words such as “ellipsis”, “ablative”, “bilingual”, and “intransitive” (Stevenson and Wilks 2001). It could do WSD in conjunction with topical words. Further details could be found in (Yarowsky 1992; Stevenson and Wilks 2001).

**Part of Speech (POS)** is associated with a subset of the word senses in both WordNet and LDOCE. That is, given the POS of the target, we may fully or partially disambiguate its sense (Stevenson & Wilks, 2001).

## 2.2 Learned World Knowledge

World knowledge is too complex or trivial to be verbalized completely. So it is a smart strategy to automatically acquire world knowledge from the context of training corpora on demand by machine learning techniques. The

frequently used types of contextual features for learning are listed below.

**Indicative Words** surround the target and can serve as the indicator of target senses. In general, the closer to the target word, the more indicative to the sense. There are several ways, like fixed-size window, to extract candidate words.

**Syntactic Features** here refer to sentence structure and sentence constituents. There are roughly two classes of syntactic features. One is the Boolean feature; for example, whether there is a syntactic object. The other is whether a specific word appears in the position of subject, direct object, indirect object, prepositional complement, etc. (Hasting 1998; Fellbaum 2001).

**Domain-specific Knowledge**, like selectional restrictions, is about the semantic restrictions on the use of each sense of the target word. However, domain-specific knowledge can only be acquired from training corpora, and can only be attached to WSD by empirical methods, rather than by symbolic reasoning. Hasting (1998) illustrates the application of this approach in the domain of terrorism.

**Parallel Corpora** are also called bilingual corpora, one serving as primary language, and the other working as a secondary language. Using some third-party software packages, we can align the major words (verb and noun) between two languages. Because the translation process implies that aligned pair words share the same sense or concept, we can use this information to sense the major words in the primary language (Bhattacharya et al. 2004).

Usually, unsupervised approaches use lexical knowledge only, while supervised approaches employ learned world knowledge for WSD. Examining the literature, however, we found the trend of combination of lexical knowledge and learned world knowledge in recently developed WSD models.

## 3. Algorithms

According to whether additional training corpora are used, WSD algorithms can be roughly classified into supervised and unsupervised categories.

### 3.1 Unsupervised Approach

The unsupervised approach does not require a training corpus and needs less computing time and power. It is suitable for online machine translation and information retrieval. However, it theoretically has worse performance than the supervised approach because it relies on less knowledge.

**Simple Approach (SA)** refers to algorithms that reference only one type of lexical knowledge. The types of lexical knowledge used include sense frequency, sense glosses (Lesk 1986), concept trees (Agiree and Rigau 1996; Agiree 1998; Galley and McKeown 2003), selectional restrictions, and subject code. It is easy to implement the simple approach, though both precision and recall are not good enough. Usually it is used for prototype systems or preliminary researches.

**Combination of Simple Approaches (CSA)** is an ensemble of the heuristics created by simply summing up the normalized weights of separate simple approaches (SA). Because multiple knowledge sources offer more confidence on a sense being used than a single source does, the ensemble usually outperforms any single approach (Agirre 2000). However, this method doesn't address the relative importance of each lexical knowledge source in the question. One alternative is to learn the weights of various lexical knowledge sources from training corpora by machine learning techniques such as Memory Based Learning (See section 3.2).

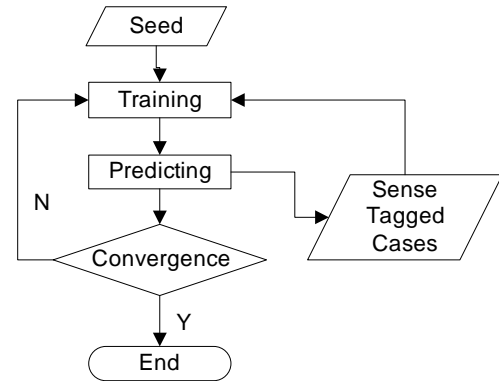
**Iterative approach (IA)** only tags some words, with high confidence in each step maintained by synthesizing the information of sense-tagged words in the previous steps and other lexical knowledge (Mihalcea and Moldovan, 2000). It is based on a fine assumption that words in a discourse are highly cohesive in terms of meaning expression, and consequently achieves high precision and acceptable recall. Mihalcea and Moldovan (2000) use this approach, disambiguating 55% of the nouns and verbs with 92.2% precision. This approach is a good choice for applications that need to sense tag all major words in text.

**Recursive Filtering (RF)** shares the same assumption as the iterative approach. That is, the correct sense of a target word should have stronger semantic relations with other words in the discourse than does the remaining sense of the target word. Therefore, the idea of the recursive filtering algorithm is to gradually purge the irrelevant senses and leave only the relevant ones, within a finite number of processing cycles (Kwong 2000). The major difference from an iterative approach is that it does not disambiguate the senses of all words until the final step.

This approach leaves open the measure of semantic relation between two concepts. Thus, it offers the flexibility of the semantic relation measure ranging from the very narrow to the very broad subject to the availability of lexical knowledge sources at the point of implementation. Kwong (2001) reports a system with maximum performance, 68.79% precision and 68.80% recall.

**Bootstrapping (BS)** looks like supervised approaches, but it needs only a few seeds instead of a large number of

training examples. The charm of this approach lies in its continuous optimization of the trained model until it reaches convergence.



**Figure 2.** Flow of Recursive Optimization Algorithm

As shown in Figure 2, it recursively uses the trained model to predict the sense of new cases and in return optimizes the model by new predicted cases. The key to the success of this method is the convergence of the supervised model. Yarowsky (1995) applies decision lists as the supervised model and achieves 96.5% precision for 12 words on average. Any supervised model can be adapted to this approach as long as it can reach convergence. RO truly achieves very high precision, rivaling supervised methods while costing much less, but it is limited to sense disambiguation of a few major words in text.

### 3.2 Supervised Approach

A supervised approach uses sense-tagged corpora to train the sense model, which makes it possible to link contextual features (world knowledge) to word sense. Theoretically, it should outperform unsupervised approaches because more information is fed into the system. Because more and more training corpora are available nowadays, most recently developed WSD algorithms are supervised. However, it does not mean unsupervised approach is already out of mode.

Supervised models fall roughly into two classes, hidden models and explicit models based on whether or not the features are directly associated with the word sense in training corpora. The explicit models can be further categorized according to the assumption of interdependence of features. Log linear models (Yarowsky 1992; Chodorow et al. 2000) simply assume each feature is conditionally independent of others. Maximum Entropy (Fellbaum 2001; Berger 1996) and Memory-based Learning do not make any assumptions regarding the independence of features. Decomposable models (Bruce 1999; O'hara et al. 2000) select the interdependence settings against the training corpus.

**Log Linear Model (LLM)** simply assumes that each feature is conditionally independent of others. For each sense  $s_i$ , the probability is computed with Bayes' rule, where  $c_j$  is  $j$ -th feature:

$$p(s_i | c_1, \dots, c_k) = \frac{p(c_1, \dots, c_k | s_i) p(s_i)}{p(c_1, \dots, c_k)}$$

Because the denominator is the same for all senses of the target word, we simply ignore it. According to the independence assumption, the term can be expressed as:

$$p(c_1, \dots, c_k | s_i) = \prod_{j=1}^k p(c_j | s_i).$$

Thus the sense for the test case should be:

$$S = \underset{S_i}{\text{ARGMAX}} \log p(s_i) + \sum_{j=1}^k \log p(c_j | s_i)$$

By counting the frequency of each feature, we can estimate the term  $\log p(c_j | s_i)$  from training data. But the seeming neatness of the algorithm can not hide its two defects: (1) the independence assumption is clearly not reasonable; (2) it needs some techniques such as Good-Turing (Good, 1953) to smooth the term of some features,  $p(c_j | s_i)$ , due to data parse problem (Chodorow et al. 2000).

**Decomposable Probabilistic Models (DPM)** fix the false assumption of log linear models by selecting the settings of interdependence of features based on the training data. In a typical decomposable model, some features are independent of each other while some are not, which can be represented by a dependency graph (Bruce and Wiebe 1999). The Grling-Sdm system in (O'Hara et al. 2000), based on a decomposable model, performs at an average level in the SENSEVAL competition. It could achieve better performance if the size of training data is large enough to compute the interdependence settings of features.

**Memory-based Learning (MBL)** classifies new cases by extrapolating a class from the most similar cases that are stored in the memory (Daelemans 1999). The basic similarity metric (Daelemans 1999) can be expressed as:

$$\Delta(X, Y) = \sum_{i=1}^n w_i \delta(x_i, y_i)$$

Where:

$$\delta(x_i, y_i) = \begin{cases} \frac{x_i - y_i}{\max_i - \min_i} & \text{if numeric, else} \\ 0 & \text{if } x_i = y_i \\ 1 & \text{if } x_i \neq y_i \end{cases}$$

In the absence of information about feature relevance, the feature weight ( $w_i$ ) can be simply set to 1. Otherwise, we can add domain knowledge bias to weight or select different

features. Information Gain and Information Ratio (Quinlan, 1993) are two frequently used metrics that address the relative importance of each feature. The overlap metric ( $\delta$ ), shown above, is the basic measure of distance between two values of a certain feature. It uses exact matching for symbolic features. To smooth this metric, Modified Value Difference Metric (MVDM) was defined by Stanfill and Waltz (1986) and further refined by Cost and Salzberg (1993). It determines the similarity of values of a feature by observing the co-occurrence of values with target classes.

$$\delta(x_i, y_i) = \sum_{j=1}^n |P(c_j | x_i) - P(c_j | y_i)|$$

Because Memory-based Learning (MBL) supports both numeric features and symbolic features, it can integrate various features into one model. Stevenson and Wilks (2001) built a WSD system using an MBL model and the recall (the precision is the same) for all major words in text surprisingly reaches 90.37% to the fine sense level.

**Maximum Entropy (ME)** is a typical constrained optimized problem. In the setting of WSD, it maximizes the entropy of  $P_\lambda(y|x)$ , the conditional probability of sense  $y$  under facts  $x$ , given a collection of facts computed from training data. Each fact is linked with a binary feature expressed as an indicator function:

$$f(x, y) = \begin{cases} 1 & \text{if sense } y \text{ is under condition } x \\ 0 & \text{otherwise} \end{cases}$$

We find:

$$P_\lambda(y | x) = \frac{1}{Z_\lambda(x)} \exp\left(\sum_i \lambda_i f_i(x, y)\right)$$

Where  $Z_\lambda(x)$  is the normalizing constant determined by the requirement that  $\sum_y P_\lambda(y | x) = 1$  for all  $x$ .

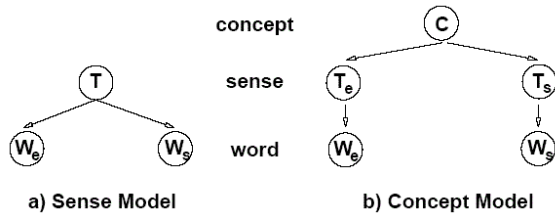
Thus the word sense of the test case should be:

$$y = \arg \max_y \frac{1}{Z_\lambda(x)} \exp\left(\sum_i \lambda_i f_i(x, y)\right)$$

From the training data, the parameter  $\lambda$  can be computed by a numeric algorithm called Improve Iterative Scaling (Berger, 1996). Berger also presents two numeric algorithms to address the problem of feature selection as there are a large number of candidate features (facts) in the setting of WSD.

Dang and Palmer (2002) apply ME to WSD. Although their model includes only contextual features without the use of lexical knowledge, the result is still highly competitive. But ME model always contains a large number of features because ME supports only binary features. Thus it is highly computing intensive.

**Expectation Maximum (EM)** generally solves the maximization problem containing hidden (incomplete) information by an iterative approach (Dempster et al. 1977). In the setting of WSD, incomplete data means the contextual features that are not directly associated with word senses. For example, given the English text and its Spanish translation, we use a sense model or a concept model to link aligned word pairs to English word sense, as shown in figure 3 (Bhattacharya et al. 2004).



**Figure 3.** Translation Model (Bhattacharya et al, 2004)

Suppose the same sense assumption is made as in the example. English word sense is the hidden variable and the complete data is  $(W_e, W_s, T)$ , denoted by  $X$ . The WSD is equivalent to choosing a sense that maximizes the conditional probability  $P(X|Y, \Theta)$ .

$$X(W_e, W_s, T) = \arg \max_T P(X | Y, \Theta)$$

Where:

$$\Theta = \{p(T), p(W_e | T), p(W_s | T)\}$$

EM then uses an iterative approach, which consists of two steps, estimation and maximization, to estimate the parameters  $\Theta$  from training data. EM is a kind of climbing algorithm. Whether it can reach global maximum depends on the initial value of the parameters. Thus, we should be careful to initialize the parameters. It is often a good choice to use lexicon statistics for initialization.

EM can learn the conditional probability between hidden sense and aligned word pairs from bilingual corpora so that it does not require the corpus to be sense-tagged. Its

performance is still highly competitive. The precision and recall of the concept model in (Bhattacharya et al, 2004) reach 67.2% and 65.1% respectively. Moreover, it is allowed to develop a big model for all major words for WSD.

So far, we have examined the characteristics of all classes of WSD algorithms. Table 1 briefly summarizes the tasks, needed knowledge sources, the level of computing complexity, resulting performance, and other features for each class of algorithms. According to the information above, we can choose the appropriate algorithms for specific applications. For example, online information retrieval requires quick response and provides a little contextual information, thus simple approach (SA) or combination of simple approach (CSA) might be a good choice.

In general, knowledge sources available to the application dramatically reduce the range of choices; computing complexity is an important consideration for time-sensitive applications; and the task type of the application further limits the applicable algorithms. After that, we may take the performance and other special characteristics into account of WSD algorithm choice.

Examining the literature of WSD, we also identify three trends with respect to the future improvement of algorithms. First, it is believed to be efficient and effective for improvement of performance to incorporate both lexical knowledge and world knowledge into one WSD model (Agirre et al. 2000; O’Hara et al. 2000; Stevenson & Wilks, 2001; Veronis, 2000). Second, it is better to address the relative importance of various features in the sense model by using some elegant techniques such as Memory-based Learning and Maximum Entropy. Last, there should be enough training data to learn the world knowledge or underlying assumptions about data distribution (O’Hara et al. 2000).

Group	Tasks	Knowledge Sources	Computing Complexity	Performance	Other Characteristics
SA	all-word	single lexical source	low	low	
CSA	all-word	multiple lexical sources	low	better than SA	
IA	all-word	multiple lexical sources	low	high precision average recall	
RF	all-word	single lexical source	average	average	flexible semantic relation
BS	some-word	sense-tagged seeds	average	high precision	sense model converges
LLM	some-word	contextual sources	average	above average	independence assumption
DPM	some-word	contextual sources	very high	above average	need sufficient training data
MBL	all-word	lexical and contextual sources	high	high	
ME	some-word	lexical and contextual sources	very high	above average	feature selection
EM	all-word	bilingual texts	very high	above average	Local maximization problem

**Table 1.** Brief summaries for each class of WSD algorithms. “all-word” means the approach is appropriate to disambiguate the sense of all major words (verb, noun, adjective and adverb) in text; “some-word” represents the suitable approach for sense disambiguation of some major words (usually verb or noun). The performance in the fifth column refers to precision and recall by default.

## 4. Conclusions

This paper summarized the various knowledge sources used for WSD and classified existing WSD algorithms according to their techniques. We further discussed the rationale, tasks, performance, knowledge sources used, computational complexity, assumptions, and suitable applications for each class of algorithms. We also identified three trends with respect to the future improvement of algorithms. They are the use of more knowledge sources, addressing the relative importance of features in the model by some elegant techniques, and the increase of the size of training data.

## References

- Agirre, E. et al. 2000. Combining supervised and unsupervised lexical knowledge methods for word sense disambiguation. *Computer and the Humanities* 34: P103-108.
- Berger, A. et al. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics* 22: No 1.
- Bhattacharya, I., Getoor, L., and Bengio, Y. 2004. Unsupervised sense disambiguation using bilingual probabilistic models. *Proceedings of the Annual Meeting of ACL 2004*.
- Bruce, R. & Wiebe, J. 1999. Decomposable modeling in natural language processing. *Computational Linguistics* 25(2).
- Chodorow, M., Leacock, C., and Miller G. 2000. A Topical/Local Classifier for Word Sense Identification. *Computers and the Humanities* 34:115-120.
- Cost, S. & Salzberg, S. 1993. A weighted nearest neighbor algorithm for learning with symbolic features, *Machine Learning, Machine Learning* 10: 57-78.
- Daelemans, W. et al. 1999. TiMBL: Tilburg Memory Based Learner V2.0 Reference Guide, Technical Report, ILK 99-01. Tilburg University.
- Dang, H.T. & Palmer, M. 2002. Combining Contextual Features for Word Sense Disambiguation. *Proceedings of the SIGLEX SENSEVAL Workshop on WSD*, 88-94. Philadelphia, USA.
- Dempster A. et al. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J Royal Statist Soc Series B* 39: 1-38.
- Fellbaum, C.1998. *WordNet: An electronic Lexical Database*, Cambridge: MIT Press.
- Fellbaum, C. & Palmer, M. 2001. Manual and Automatic Semantic Annotation with WordNet. *Proceedings of NAACL 2001 Workshop*.
- Galley, M., & McKeown, K. 2003. Improving Word Sense Disambiguation in Lexical Chaining, *International Joint Conferences on Artificial Intelligence*.
- Good, I.F. 1953. The population frequencies of species and the estimation of population parameters. *Biometrika* 40: 154-160.
- Hastings, P. et al. 1998. Inferring the meaning of verbs from context *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society (CogSci-98)*, Wisconsin, USA.
- Kwong, O.Y. 1998. Aligning WordNet with Additional Lexical Resources. *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, Montreal, Canada.
- Kwong, O.Y. 2000. Word Sense Selection in Texts: An Integrated Model, Doctoral Dissertation, University of Cambridge.
- Kwong, O.Y. 2001. Word Sense Disambiguation with an Integrated Lexical Resources. *Proceedings of the NAACL WordNet and Other Lexical Resources Workshop*.
- Lesk, M. 1986. Automatic Sense Disambiguation: How to Tell a Pine Cone from and Ice Cream Cone. *Proceedings of the SIGDOC'86 Conference, ACM*.
- Mihalcea, R. & Moldovan, D. 2000. An Iterative Approach to Word Sense Disambiguation. *Proceedings of Flairs 2000*, 219-223. Orlando, USA.
- Ng, H.T., Lim, C. and Foo, S. 1999. A Case Study on Inter-Annotator Agreement for Word Sense Disambiguation, in *Proceedings of the ACL SIGLEX Workshop: Standardizing Lexical Resources*.
- O'Hara, T, Wiebe, J., & Bruce, R. 2000. Selecting Decomposable Models for Word Sense disambiguation: The Grling-Sdm System. *Computers and the Humanities* 34: 159-164.
- Quinlan, J.R. 1993. C4.5: *Programming for Machine Learning*. Morgan Kaufmann, San Mateo, CA.
- Stevenson, M. & Wilks, Y. 2001. The Interaction of Knowledge Sources in Word Sense Disambiguation. *Computational Linguistics* 27(3): 321 - 349.
- Stanfill, C. & Waltz, D. 1986. Towards memory-based reasoning, *Communications of the ACM* 29(12): 1213-1228.
- Veronis, J. 2000. Sense Tagging: Don't Look for the Meaning But for the Use, *Workshop on Computational Lexicography and Multimedia Dictionaries*, 1-9. Patras, Greece.
- Yarowsky, D. 1992. Word Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. *Proceedings of COLING-92*, 454-460. Nantes, France.
- Yarowsky, D. 1994. Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French. *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, NM.
- Yarowsky, D. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. *Meeting of the Association for Computational Linguistics*, 189-196.