Learning Rules for Large Vocabulary Word Sense Disambiguation

Georgios Paliouras, Vangelis Karkaletsis, Constantine D. Spyropoulos
Institute of Informatics & Telecommunications,
NCSR "Demokritos",
Aghia Paraskevi Attikis
Athens, 15310
Greece

Abstract

Word Sense Disambiguation (WSD) is the process of distinguishing between different senses of a word. In general, the disambiguation rules differ for different words. For this reason, the automatic construction of disambiguation rules is highly desirable. One way to achieve this aim is by applying machine learning techniques to training data containing the various senses of the ambiguous words. In the work presented here, the decision tree learning algorithm C4.5 is applied on a corpus of financial news articles. Instead of concentrating on a small set of ambiguous words, as done in most of the related previous work, all content words of the examined corpus are disambiguated. Furthermore, the effectiveness of word sense disambiguation for different parts of speech (nouns and verbs) is examined empirically.

1 Introduction

The meaning of a word may vary significantly according to the context in which it is used. For instance the word "bank" will have a completely different meaning in financial text than in geological text. This is a case of a clearly identifiable sense distinction, but there are cases where different senses of a word may be harder to distinguish, e.g. "bank" as a financial institution and as a building. Both senses are likely to appear in the same context and one needs to take into account the details of their use, in order to distinguish between them. The process of distinguishing between different senses of a word is called word-sense disambiguation (WSD). Word-sense disambiguation is necessary for a number of tasks in natural language processing (NLP), such as machine translation, query-based information retrieval and information extraction.

In general, the rules for distinguishing between the senses of different words differ. For instance, a valid disambiguation rule for the senses of the word "bank" would examine the occurrence of the words "river", "financial", etc. in the context of the ambiguous word. This evidence would be completely irrelevant for most other words. Thus the disambiguation rules are in general word-specific. Furthermore, it is difficult to construct such rules manually, especially when the difference between the senses is not great, e.g. "bank" the institution and the building. For this reason, the automatic construction of disambiguation rules is highly desirable. One way to achieve this aim is by applying machine learning techniques to training data containing the various senses of the ambiguous words.

The machine learning method used here belongs in the class of symbolic supervised machine learning, requiring that the training texts are hand-tagged with the correct senses for ambiguous words. An important aspect of the work presented here, as compared to similar previous work, is that all content words (rather than a handful of them) in the training texts are subject to disambiguation. This step towards large-vocabulary disambiguation is necessary if WSD systems are to be used in practice. However, the automatic construction of large-vocabulary disambiguators is hard, due to the sparseness of the training data for each individual word. One of the issues examined in this context is the construction of simple general rules that apply to all words, capturing regularities in less frequent words in the data.

Another important issue that we examine is the effectiveness of word sense disambiguation for different parts of speech (nouns and verbs) and the ability to learn disambiguators for each of those two word-types. The learning algorithm is applied separately to verbs and nouns and the results are compared.

Section 2 presents related work in WSD. The WSD task, as this is realised in our approach, is presented in Section 3. Our experiments (i.e., experimental setup and results) are presented in Section 4. Finally, in section 5, we summarise the work and present our future plans.

2 Related Work

Early efforts in automating the sense disambiguation task made use of Machine-Readable Dictionaries (MRDs) and thesauri, which associate different senses of a word with short definitions, examples, synonyms, hypernyms, hyponyms, etc. A simple approach of this type is to compare the dictionary definitions of words appearing in the surrounding text of an ambiguous word with the text in the definition of each sense of the ambiguous word in the dictionary. Clearly, the higher the overlap between the dictionary definitions of the surrounding words and the definition of a particular sense of the ambiguous word, the more likely it is that this is the correct sense for the word. Some of the methods that are based on MRDs and thesauri are presented in [Lesk, 1986; Wilks, *et al.*, 1990; Cowie, *et al.*, 1992]. The resources that are commonly used in these studies are: the WordNet, Longman's Dictionary of Contemporary English (LDOCE), Roget's thesaurus and Collins English Dictionary (CDE). A more thorough account of this work can be found in [Ide and Veronís, 1998].

Despite the useful information that they contain, MRDs and thesauri are often inadequate for WSD, e.g. MRD sense definitions are often non-representative of the context in which the sense is met. As a result, the focus of WSD research has recently turned to *corpus-based* methods. According to this approach, a corpus of text is used as training data for the construction of disambiguation rules for different words. The construction of these disambiguation rules is achieved by a variety of machine learning methods.

An important distinguishing feature for machine learning methods is the extent of supervision provided for training. Supervision is provided in the form of hand-labelling the examples that are used for learning. In the case of WSD, a fully supervised method requires that all occurrences of an ambiguous word in the training text be labelled with the correct sense. The sense labels are typically taken from a dictionary. Given this information, a supervised learning algorithm constructs rules that achieve high discrimination between occurrences of different word-senses. Examples of supervised learning methods for WSD appear in [Black, 1988; Gale *et al.*, 1993; Leacock *et al.*, 1993; Yarowsky, 1994; Towell and Voorhees, 1998]. The learning methods used in those studies are general-purpose, including: decision-tree induction, decision-list induction, feed-forward neural networks with backpropagation and naïve Bayesian learning. Their results are very encouraging, exceeding 90% correct sense labelling in some cases.

However, this high disambiguation rate is achieved at the expense of disambiguating only a small number of words. In all of the above-mentioned studies only a handful of words are included in the evaluation experiments and for each of these words a sufficient number of examples are provided, covering all senses of the word. This is an unrealistic scenario, when aiming to construct a system to be used in practice. The results presented here are on a much larger scale, considering all content words of a corpus. A similar approach has been adopted by the system that won the Senseval competition[1] and is presented in forthcoming work [Hawkins and Nettleton, 1999]. Despite the fact that the Senseval competition did not involve large-scale disambiguation, the system presented in [Hawkins and Nettleton, 1999] is designed to deal with a large number of words, each represented by a small number of examples. For this purpose it has been evaluated on the SEMCOR corpus, which contains about 200,000 content words, achieving 63.72% accuracy on low-level WordNet senses. The low accuracy figure, in conjunction with the fact that the same system won the Senseval competition, illustrates the difficulty of large-vocabulary disambiguation.

In addition to the supervised approaches to learning WSD systems, unsupervised learning has been used for the same purpose, which does not require hand-tagging of the training data, e.g. [Yarowsky, 1992; Leacock *et al.*, 1998; Schütze, 1998]. As expected, the performance of the unsupervised learning approaches is lower than that of their supervised counterparts. However, performance evaluation of unsupervised learning methods is not straightforward, as there are no correct tags against which to compare the results of the disambiguation.

A compromise solution between supervised and unsupervised learning is the use of a small number of tagged examples, together with a large set of untagged data. Such partially supervised learning methods are presented in [Yarowsky, 1995; Towell and Vorhees, 1998], using rule-learning and neural networks respectively.

An important issue for any WSD learning algorithm is what features will be used to construct the disambiguation rules, i.e., what evidence is relevant for WSD. Since syntactic information is not considered useful for hard WSD tasks, the evidence commonly used consists of words that can be found in the neighbourhood of the ambiguous word. The question that arises then is how large this neighbourhood ought to be, i.e., how broad a context is needed for disambiguation. According to this criterion, the WSD methods in the literature can be divided into two large groups: *local* and *topical* WSD. In local WSD only the close neighbourhood of the word (<10 words on each side) is used. Topical methods on the other hand use a larger context window (> 50 words on each side). None of the fairly recent approaches presented above uses purely local information. Yarowsky [1992] and Schütze [1998] present purely topical methods, but in both papers the value of local information is noted. Most of the recent approaches, e.g. [Yarowsky, 1994; Towell and Voorhees, 1998], combine local and topical information, in order to improve their performance. Another interesting claim is that different sizes of context window are effective for different parts of speech. Noun senses seem to be dependent on topical information, while verbs and adjectives are better disambiguated using local information [Yarowsky, 1993].

[1] Senseval was the first competition for WSD systems. For more information see [Kilgariff, 1998].

A critical component of any application of machine learning is the representation of the training examples and the generated model, i.e., the disambiguator here. The most popular representation for training examples in machine learning is the feature vector, i.e., a fixed set of features, taking values from a fixed set. Examples of such features in WSD may be collocated words, within a window surrounding the ambiguous word. These types of feature dominate the literature on learning methods for WSD. Despite the encouraging results obtained in most studies, features of this type cause a combinatorial explosion in the space of possible solutions. This is due to the unbounded value set of the features, e.g. any word can be a collocate of any other word. Little work has been done so far on alternative forms of representation. Yarowsky [1992] looks at classes of words and Schütze [1998], groups words that occur in similar contexts.

3 The Word Sense Disambiguation Task

The data used in this study are extracted from the SEMCOR corpus, mentioned also in section 2. SEMCOR is a selection of various texts from the Brown corpus. The important feature of this corpus is that the content words, i.e., nouns, verbs, adjectives and adverbs, have been hand-tagged with syntactic and semantic information, as part of the WordNet project. A subset of SEMCOR is used here containing only financial news articles. We have chosen this subset of SEMCOR, because a previous study [Paliouras *et al.,* 1998] has shown that better disambiguation performance can be achieved by focusing on a specific thematic domain.

The SEMCOR corpus is tagged with WordNet sense-numbers. However, the information extraction system for which we want to use the WSD methods,[2] makes use of the Longman Dictionary of Contemporary English (LDOCE). For this reason we translated the WordNet tags into their equivalent in LDOCE. This translation was supported by a resource that was constructed in the WordNet project: a mapping between the senses in the two dictionaries [Bruce and Guthrie, 1992]. The mapping between WordNet and LDOCE senses suffers in several respects. Two important problems are:

- there is a large number of senses in both dictionaries that have not been mapped onto senses in the other dictionary;
- the mapping between senses is hardly ever one-to-one, e.g. seven different Wordnet senses for the verb 'absorb' are mapped onto the same LDOCE sense, while the word has four LDOCE senses.

Due to these problems, there is a loss of information in the translation of the data from WordNet to LDOCE tags. In average, only a quarter of the words in the corpus can be assigned LDOCE senses.

The SEMCOR text is translated into training data using the feature-vector representation. For each word, each of its LDOCE senses with the correct part of speech is represented as a separate example case for learning. The correct sense is labeled as a positive example and all other senses as negative. Each example case contains the following characteristic information about the word and the context in which it appears: the lemma of the word, the rank of the sense in LDOCE corresponding to how frequently the sense appears in general text, the part-of-speech tag for the word and ten collocates (first noun/verb/preposition to the left/right and first/second word to the left). For instance, the word "bank" in the following example:

If you destroy confidence in banks you do something to the economy.

is being used with LDOCE sense rank 1. The feature vector representation of the positive example extracted from this sentence is:

{"bank", 1, noun, "confidence", "destroy", "in", "economy", "do", "to", "in", "confidence", "you", "do"}

where "bank" is the lemma, 1 is the sense rank, noun the part of speech, "confidence", "destroy", "in", the noun, verb, preposition on the left, "economy", "do", "to", similarly on the right and "in", "confidence", "you", "do", the four words surrounding the word "bank". In addition to this positive example a number of negative vectors are generated, one for each alternative sense for the noun "bank".

The evidence used in the disambiguation of different words differs significantly. This is because the disambiguation evidence consists of specific words that commonly occur in the context of the ambiguous word. Due to this fact, almost all work in WSD has considered words individually, i.e., a different disambiguator is built for each word. In the work presented here we test this hypothesis by attempting to construct a common disambiguation system for all content words in SEMCOR, using the lemma of the ambiguous word as a feature in the training examples. Depending on the use of the lemma by the disambiguation system, we can judge whether the system performs word-specific disambiguation, or whether the learning procedure has identified word-independent patterns in the data. The identification of such general patterns is particularly useful when disambiguating rare words, for which there is insufficient training information in the data. This is unavoidable when performing large-vocabulary WSD.

Another issue that is examined in the following experiments is the ability to learn disambiguators for words of different part of speech. In particular we extract the training examples corresponding to verbs and nouns, ignoring adjectives and adverbs that are few and usually unambiguous. Past work [Yarowsky, 1993] has suggested that verbs are disambiguated better than nouns, when using local context, which is the case here.

[2] The work presented here was part of the project ECRAN (Extraction of Content: Research at Near-market), LE-2110, Telematics Application Programme.

4 Experimental results

4.1 Experimental set-up

The machine learning algorithm used here is called C4.5 [Quinlan, 1993] and performs symbolic supervised learning, using hand-tagged training data. C4.5 constructs decision trees, which can also be translated into lists of rules. The algorithm is general-purpose, i.e., it has not been designed specifically for this task and has actually been used with success in various other real-world problems. Thus, our aim is not to evaluate the algorithm as such, but to examine whether it is appropriate for the particular WSD task.

The measures that were chosen for the evaluation of the algorithm are those typically used in the language engineering and machine learning literature: *recall*, *precision* and *accuracy*. The recall measure is based on the number of positive examples that were identified as such by the WSD system. These are called True Positives (TP). The actual recall measure is the ratio of True Positives to the total number of positives (P) in the test data. On the other hand, precision is the ratio of True Positives to all examples classified as positive by the system. In addition to these two measures the percentage correct classification (accuracy), which is a standard measure for machine learning methods is used. In summary the three ratios:

recall = TP/P,
precision = TP/(TP+FP),
accuracy = (TP+TN)/(P+N),

where TP, FP, P as described above, TN the true negative and N the total number of negative examples.

In all of the figures presented in the following sections, the performance of the system is measured on unseen data. Furthermore, in order to arrive at a robust estimate of performance, we use 10-fold cross-validation at each individual experiment. Thus, each recall, precision and accuracy figure presented in the following sections is an average over ten runs, rather than a single train-and-test result, which can often be accidentally high or low. Finally, it should be noted that, unlike MRD-based approaches, the constructed disambiguators make no use of external resources.

4.2 Results on all content words

The financial news articles of SEMCOR consist of 3,613 word occurrences, of which 1,987 have been tagged with WordNet senses, resulting in 753 word occurrences with LDOCE senses and 355 distinct words. The LDOCE polysemy of the dataset is 3,516/753=4.67, and the ratio of word occurrences to distinct words, i.e., the average word repetition is 753/355=2.12. Word repetition is one indication of the richness of the vocabulary in the text. The closer the ratio is to 1, the richer the vocabulary.

In order to set the results in context, we examine the following simple base case: we consider as appropriate the first sense of each word in LDOCE, i.e., the most frequently used sense. Table 1 presents the results, using this naïve rule. Clearly, any results close or below these values are not acceptable as a solution to the problem.

Recall	Precision	Accuracy
48.0%	65.0%	77.3%

Table 1: The base case for all content words.

C4.5 facilitates pre- and post-pruning of the decision tree using significance statistics. The performance of the decision trees generated by C4.5 for different levels of pruning was examined. Figure 1 plots the results for different tree sizes. The curves for precision and accuracy are nearly flat for trees with at least 1,000 nodes. However, recall starts with low values and increases significantly for tree sizes between 2,500 and 3,500 nodes. Thus, performance is better for large trees. The results of C4.5 are safely above the base case for tree sizes greater than 1,000 nodes.

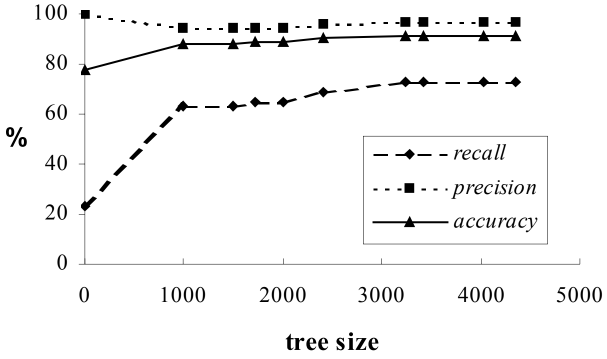

Figure 1. Performance of C4.5 on all content words.

It should be noted here that a tree of 1,000 nodes in this application is not large, because of the large branching factor. Each feature in the feature vector takes a large number of values (corresponding to words appearing in the text) and when a feature is examined, one node for each value of the feature is constructed.

The tree-disambiguators are doing particularly well in terms of precision, but not so in terms of recall. This is an indication that the trees are conservative in labelling example cases as positive. This bias towards negative examples is due to the disproportionately large number of negative examples: 2,489 out of 3,516 examples.

One of the questions set initially was whether word-independent rules can be constructed, i.e., whether general patterns can be identified that are independent of specific words and still can be used for disambiguation. In general, the constructed decision trees represent collections of word-specific disambiguators, i.e., they are composed of subtrees that focus on specific words. The only exception to this phenomenon is the use of the sense rank to group words that appear less frequently in the training set. For most of those words, the learning algorithms decide to select the most frequent sense (highest rank), rather than building complex disambiguation rules, using the collocates. This combination of general and word-specific disambiguation is desirable for large-vocabulary WSD.

4.3 Results on nouns and verbs separately

Another issue examined here is the different behaviour of disambiguators for words of different part of speech (verbs and nouns). Out of the 3,516 examples in the complete dataset, 557 are verb-cases, and 2,846 are noun-cases. The remaining 113 examples correspond to adjectives and adverbs.

The 557 verb-cases represent 134 occurrences of 77 different verbs. Thus, LDOCE polysemy in this subset of the data is 557/134=4.16 and average word repetition 134/77=1.74. The base case performance of choosing the most frequent sense is shown in Table 2.

Recall	Precision	Accuracy
66.2%	71.6%	84.4%

Table 2: The base case for verbs only.

The base case results in this case are better than those in the complete dataset, suggesting an easier disambiguation problem. This is in accordance to the lower polysemy value. However, average word repetition is considerably lower than before, making learning more difficult.

Figure 2 shows the performance of C4.5 on this reduced problem. In comparison to the results in Figure 1, recall has improved slightly, while precision has decreased considerably. Overall, there is little improvement over the base case for all three measures.

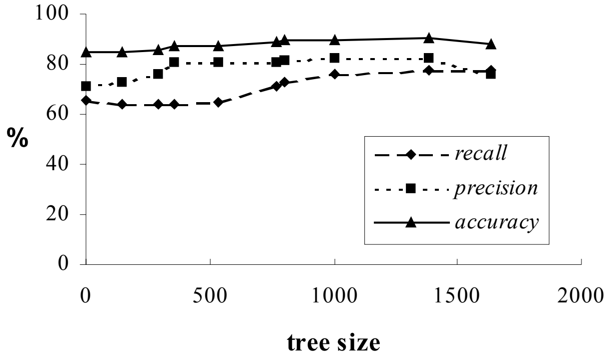

Figure 2. Performance of C4.5 on verbs only.

The 2,489 noun-cases represent 534 occurrences of 244 different nouns in the text. Thus the polysemy in the dataset is 2,489/534=4.66 and the average word repetition is 534/244=2.19. Both values are close to those in the complete dataset, since the noun-cases correspond to a large proportion of the dataset. The polysemy is larger than for verbs, suggesting a difficult disambiguation task. However, word repetition is also higher than for verbs, suggesting that learning can do better in this problem. The base case for the naïve most-frequent-sense rule is shown in Table 3.

Recall	Precision	Accuracy
39.7%	58.5%	75.5%

Table 3: The base case for nouns only.

According to all measures, this problem seems harder than the disambiguation of verbs. The results for the base case are in accordance with the higher polysemy.

Figure 3 presents the performance of C4.5 for noun disambiguation. As expected, the results in this experiment are similar to these for the whole dataset. The main difference is the level of recall, which is considerably lower. This can be explained by the removal of adjectives and adverbs from the dataset, for which almost 100% recall is achieved. Compared to the results for verb disambiguation, recall is lower, but precision is higher. Thus, it is difficult to draw a conclusion about whether verbs or nouns are disambiguated better. However, in terms of learning the results are much better for nouns than for verbs, since there is an improvement over the base-case results.

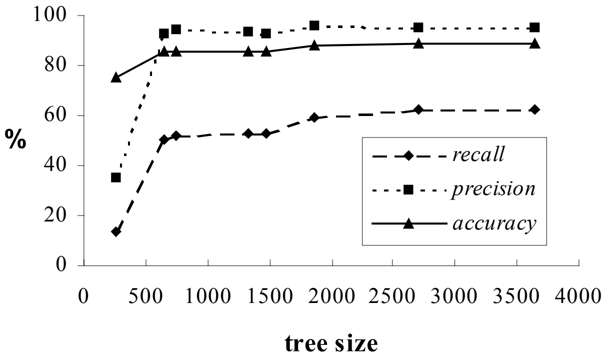

Figure 3. Performance of C4.5 on nouns only.

5 Concluding Remarks and Further Work

Machine learning algorithms are a promising approach to the automatic construction of word sense disambiguators. We examined a symbolic supervised learning technique, C4.5, which requires that the training texts are hand-tagged with the correct senses for ambiguous words. The learning algorithm was evaluated on financial news articles from the SEMCOR corpus. The textual data were translated into feature-vector examples, as needed by the learning algorithm. 10-fold cross-validation was used to gain an unbiased estimate of the performance of the algorithm. Two experiments were carried out: one using all content words and one examining verbs and nouns separately.

An important difference of the work presented here from previous work on this subject is the size of the vocabulary being disambiguated. Rather than restricting the attention of the system to a handful of words, all content words in the data were considered for disambiguation. This is a more realistic scenario, introducing the problem of sparseness of the training data. The reaction of the learning algorithm to this was to combine a simple general disambiguation filter for the words that appear less frequently in text, with word-specific disambiguation rules for the remaining words. This combination of word-specific and general disambiguation rules is an interesting outcome of our experiments that deserves further study. The overall disambiguation results were comparable to those presented in [Hawkins and Nettleton, 1999], where large-vocabulary disambiguation is also examined. However, the results of the two studies are not directly comparable, due to the use of a different set of senses, i.e., LDOCE instead of WordNet.

Another interesting issue was generated by the second experiment that looked at the disambiguation of different parts of speech. The behaviour of the learning algorithm was different for nouns than for verbs, but no conclusion could be reached as to whether local information favours verbs or nouns. However, the interesting observation is the difference between the difficulty of the disambiguation problem and the learning task. The verb disambiguation problem examined here seems easier than the noun disambiguation one. However, the task of learning a good disambiguator for verbs was harder than that of learning to disambiguate nouns.

Another issue that we want to examine in the future is the appropriate representation of training examples. The representation that was used here separates word instances into different senses, which are then treated as individual examples. Alternative representations that would allow the grouping of all senses related to a single word, should also be examined.

Finally, an important issue in WSD is the extent of the context used for disambiguation. Only local context was taken into account here. Topical evidence has also been shown to help in WSD and should be examined.

References

[Black, 1988] Black E., "An experiment in computational discrimination of English word senses." *IBM Journal of Research and Development*, v.32, n.2, pp. 185-194, 1988.

[Bruce and Guthrie, 1992] Bruce, R. and Guthrie, L., "Genus Disambiguation: A study in weighted preference." In *Proceedings of the International Conference on Computational Linguistics,* pp. 1187-1191, 1992.

[Cowie, *et al.*, 1992] Cowie, J., Guthrie, J. A. and Guthrie, L., "Lexical disambiguation using simulated annealing." In *Proceedings of the International Conference on Computational Linguistics*, pp. 359-365, 1992.

[Gale *et al.*, 1993] Gale, W. A., Church, K. W. and Yarowsky, D., "A method for disambiguating word senses in a large corpus." *Computers and the Humanities*, v.26, pp.415-439, 1993.

[Hawkins and Nettleton, 1999] Hawkins, P. and Nettleton, D., "A Hybrid Word Sense Disambiguation Algorithm.", *Journal of Natural Language Engineering*, 1999. (to appear)

[Ide and Veronís, 1998] Ide, N. and Veronís, J., "Introduction to the special issue on Word Sense Disambiguation: The state of the art." *Computational Linguistics*, v.24, n.1, pp. 1-40, 1998.

[Kilgariff, 1998] Kilgariff, A., "SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs." In *Proceedings of the Language Resources and Evaluation Conference,* pp. 581-588, 1998.

[Leacock *et al.*, 1993] Leacock, C., Towell, G. and Voorhees, E. M., "Corpus-based statistical sense resolution." In *Proceedings of the ARPA Human Languages Technology Workshop*, 1993.

[Leacock *et al.*, 1998] Leacock, C., Chodrow, M. and Miller, G. A., "Using corpus statistics and WordNet relations for sense identification." *Computational Linguistics*, v.24, n.1, pp. 147-165, 1998.

[Lesk, 1986] Lesk, M., "Automated sense disambiguation using machine-readable dictionaries: How to tell an pine cone from an ice cream cone." In *Proceedings of the SIGDOC Conference*, pp. 24-26, 1986.

[Quinlan, 1993] Quinlan, J. R., *C4.5: Programs for machine learning*, Morgan-Kaufmann, 1993.

[Schütze, 1998] Schütze, H., "Automatic word sense discrimination." *Computational Linguistics*, v.24, n.1, pp. 97-124, 1998.

[Towell and Voorhees, 1998] Towell, G. and Voorhees, E. M., "Disambiguating highly ambiguous words." *Computational Linguistics*, v.24, n.1, pp. 125-146, 1998.

[Wilks *et al.*, 1990] Wilks, Y. A., Fass, D. C., Guo, C. M., MacDonald, J. E., Plate, T. and Slator, B. M., "Providing machine tractable dictionary tools." *Machine Translation*, v.5, pp. 99-154, 1990.

[Yarowsky, 1992] Yarowsky, D., "Word-sense disambiguation using statistical models of Roget's categories trained on large corpora." In *Proceedings of the International Conference in Computational Linguistics,* pp. 454-460, 1992.

[Yarowsky, 1993] Yarowsky, D., "One sense per collocation." In *Proceedings of the ARPA Human Languages Technology Workshop*, pp. 266-271, 1993.

[Yarowsky, 1994] Yarowsky, D., "Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French." In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 88-95, 1994.

[Yarowsky, 1995] Yarowsky, D., "Unsupervised word sense disambiguation rivaling supervised methods." In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 189-196, 1995.