Journal of Artificial Intelligence Research 33 (2008) 79-107 Submitted 07/07; published 09/08

On the Use of Automatically Acquired Examples for All-Nouns Word Sense Disambiguation

David Martinez DAVIDM@CSSE.UNIMELB.EDU.AU
University of Melbourne
3010, Melbourne, Australia
Oier Lopez de Lacalle OIER.LOPEZDELACALLE@EHU.ES
University of the Basque Country
20018, Donostia, Basque Country
Eneko Agirre E.AGIRRE@EHU.ES
University of the Basque Country
20018, Donostia, Basque Country

Abstract

This article focuses on Word Sense Disambiguation (WSD), which is a Natural Language Processing task that is thought to be important for many Language Technology applications, such as Information Retrieval, Information Extraction, or Machine Translation. One of the main issues preventing the deployment of WSD technology is the lack of training examples for Machine Learning systems, also known as the Knowledge Acquisition Bottleneck. A method which has been shown to work for small samples of words is the automatic acquisition of examples. We have previously shown that one of the most promising example acquisition methods scales up and produces a freely available database of 150 million examples from Web snippets for all polysemous nouns in WordNet. This paper focuses on the issues that arise when using those examples, all alone or in addition to manually tagged examples, to train a supervised WSD system for all nouns. The extensive evaluation on both lexical-sample and all-words Senseval benchmarks shows that we are able to improve over commonly used baselines and to achieve top-rank performance. The good use of the prior distributions from the senses proved to be a crucial factor.

1. Introduction

This paper is devoted to the Word Sense Disambiguation (WSD) task for Natural Language Processing (NLP). The goal of this task is to determine the senses of the words as they appear in context. For instance, given the sentence *"He took all his money from the* ***bank.****"*, if we focus on the word *bank*, the goal would be to identify the intended sense, which in this context would be some *"financial"* sense, instead of other possibilities like the *"edge of river"* sense. The senses can be defined in a dictionary, knowledge-base or ontology. This task is defined as an intermediate step towards natural language understanding. The construction of efficient algorithms for WSD would benefit many NLP applications such as Machine Translation (MT), or Information Retrieval (IR) systems (Resnik, 2006). For instance, if an MT system was to translate the previous example into French, it would need to choose among the possible translations of the word *bank*. This word should be translated as *"banque"* when it is used in the financial sense (as in the example), but as *"rive"* when it is used in the *"edge of river"* sense. See the work of Vickrey, Biewald, Teyssier, and Koller (2005) for a recent evaluation of cross-lingual WSD in MT. For IR engines, it would also be useful to determine which is the sense of the word in the query in order to retrieve relevant documents, specially when working with multilingual documents in Cross-Language Information Retrieval (CLIR), or other IR scenarios where recall is a key performance factor, such as retrieving images by their captions. Some evidence in favor of using WSD in IR has been gathered lately (Kim, Seo, & Rim, 2004; Liu, Liu, Yu, & Meng, 2004; Stevenson & Clough, 2004; Vossen, Rigau, Alegría, Agirre, Farwell, & Fuentes, 2006).

WSD techniques can also fill an important role in the context of the Semantic Web. The Web has grown focusing on human communication, rather than automatic processing. The Semantic Web has the vision of automatic agents working with the information in the Web at the semantic level, achieving interoperability with the use of common terminologies and ontologies (Daconta, Obrst, & Smith, 2005). Unfortunately most of the information in the Web is in unstructured textual form. The task of linking the terms in the texts into concepts in a reference ontology is paramount to the Semantic Web.

Narrower domains like Biomedicine are also calling for WSD techniques. The Unified Medical Language System (UMLS) (Humphreys, Lindberg, Schoolman, & Barnett, 1998) is one of the most extensive ontologies in the field, and studies on mapping terms in medical documents to this resource have reported high levels of ambiguity, which calls for WSD technology (Weeber, Mork, & Aronson, 2001).

WSD has received the attention of many groups of researchers, with general NLP books dedicating separate chapters to WSD (Manning & Schütze, 1999; Jurafsky & Martin, 2000; Dale, Moisl, & Somers, 2000), special issues on WSD in NLP journals (Ide & Veronis, 1998; Edmonds & Kilgarriff, 2002), and books devoted specifically to the issue (Ravin & Leacock, 2001; Stevenson, 2003; Agirre & Edmonds, 2006). The interested reader can start with the dedicated chapter by Manning and Schütze (1999) and the WSD book (Agirre & Edmonds, 2006). The widespread interest motivated the Senseval initiative[1], which has joined different research groups in a common WSD evaluation framework since 1998. The goal is to follow the example of other successful competitive evaluations, like DUC (Document Understanding Conference) or TREC (Text Retrieval Conference).

WSD systems can be classified according to the knowledge they use to build their models, which can be derived from different resources like corpora, dictionaries, or ontologies. Another distinction is drawn on corpus-based systems, distinguishing between those that rely on hand-tagged corpora (*supervised systems*), and those that do not require this resource (*unsupervised systems*). This distinction is important because the effort required to hand-tag senses is high, and it would be costly to obtain tagged examples for all word senses and all languages, as some estimations show (Mihalcea & Chklovski, 2003). In spite of this drawback (referred to as "the knowledge acquisition bottleneck"), most of recent efforts have been devoted to the improvement of supervised systems, which are the ones that obtain the highest performance, even with the current low amounts of training data. These systems rely on sophisticated Machine Learning (ML) algorithms that construct their models based on the features extracted from the training examples.

Alternatively, Senseval defines two kinds of WSD tasks: lexical-sample and all-words. In a lexical-sample task the systems need to disambiguate specific occurrences of a handful

[1] http://www.senseval.org

of words for which relatively large numbers of training examples are provided (more than 100 examples in all cases). In the all-words task, no training data is provided, and testing is done for whole documents. Systems need to tag all content words occurring in the texts, even if only small amounts of external training data are available.

The analysis of the results for the English lexical-sample exercise in the third edition of Senseval (Mihalcea & Edmonds, 2004) suggested that a plateau in performance had been reached for ML methods. For this task, where the systems had relatively large amounts of training data, there were many systems on the top, performing very close to each other. The systems were able to significantly improve the baselines and attained accuracies above 70% (Mihalcea, Chklovski, & Killgariff, 2004).

The case was different in the all-words task (Snyder & Palmer, 2004), where supervised systems also performed best. They used training examples from Semcor (Miller, Leacock, Tengi, & Bunker, 1993), which is the only sizable all-words sense-tagged corpus at the time of writing this paper. The scarcity of examples and the use of test documents from corpora unrelated to Semcor heavily affected the performance, and only a few systems scored above the baseline method of assigning the most frequent sense in Semcor. In order to be useful for NLP applications, WSD systems have to address the knowledge acquisition bottleneck for all (or at least a significant part) of the word types, as evaluated by all-words tasks. Lexical-sample tasks are useful for evaluating WSD systems under ideal conditions (i.e. regarding availability of training data), but they do not show systems to be scalable to all the words in the vocabulary. In this work we will use a lexical-sample task in order to adjust some parameters of our system, but the main evaluation is on an all-words task. Our experiments are designed accordingly: the lexical-sample tests show empirical evidence on specific parameters, and the all-words evaluation compares our systems to the state of the art.

In this article, we explore a method to alleviate the knowledge acquisition bottleneck at a large scale. We use WordNet (Fellbaum, 1998) to automatically acquire examples from the Web. The seminal work of Leacock, Chodorow, and Miller (1998) showed that the approach was promising, with good results on a small sample of nouns. Other works in the field of automatic acquisition of examples have focused on exploring different approaches to the acquisition process (Agirre, Ansa, Martinez, & Hovy, 2000; Mihalcea, 2002; Cuadros, Padró, & Rigau, 2006), with a straightforward application to WSD. Those explorations typically required costly querying over the Web, and thus tried a limited number of variations for a handful of words. Our approach is different in spirit: we want to go through the whole process for all nouns, from the acquisition of examples itself to their use on WSD and the thorough evaluation on the Senseval 2 lexical-sample and Senseval 3 all-words datasets. This comes at the cost of not exploring all the different possibilities at each step, but has the advantage of showing that the results are extensive, and not limited to a small set of nouns.

For these reasons, and given the prior work on acquisition techniques, we use the most efficient and effective example acquisition method according to independent experiments performed by Agirre et al. (2000) and Cuadros et al. (2006). The focus of this paper is thus on the issues that arise when using those examples as training data of a supervised ML system. This paper will show that the automatically acquired examples can be effectively used with or without pre-existing data, and that deciding the amount of examples to use for each sense (the prior distribution) is a key issue.

The objectives of this paper are to show that existing methods to acquire examples from the Web scale-up to all nouns, and to study other issues that arise when these examples are to be used as training data in an all-nouns WSD system. Our goal is to build a state-of-the-art WSD system for all nouns using automatically retrieved examples.

Given the cost of large-scale example acquisition, we decided to limit the scope of our work only to nouns. We think that noun disambiguation on its own can be a useful tool in many applications, specially in the IR tasks mentioned above. Our method can be easily adapted to verbs and adjectives (Cuadros et al., 2006), and we plan to pursue this line in the future.

The work reported here has been partially published in two previous conference papers. The method for the automatic acquisition of examples was described by Agirre and Lopez de Lacalle (2004). A first try on the application of those examples to Word Sense Disambiguation was presented in Agirre and Martinez (2004b). In this paper we present a global view of the whole system, together with a more thorough evaluation, which shows that the automatically acquired examples can be used to build state-of-the-art WSD systems in a variety of settings.

The article is structured as follows. After this introduction, related work on the knowledge acquisition bottleneck in WSD is described in Section 2, with a focus on automatic example acquisition. Section 3 introduces the method to automatically build SenseCorpus, our automatically acquired examples for WordNet senses. Section 4 describes the experimental setting. Section 5 explores some factors on the use of SenseCorpus and evaluates them on a lexical-sample task. The final systems are thoroughly evaluated on an all-nouns task in Section 6. Finally, Section 7 provides some discussion, and the conclusions and further work are outlined in Section 8.

2. Related Work

The construction of WSD systems applicable to all words has been the goal of many research initiatives. In this section we will describe related work that looks for ways to alleviate the knowledge acquisition bottleneck using the following techniques: bootstrapping, active learning, parallel corpora, automatic acquisition of examples and acquisition of topic signatures. Sections 5 and 6, which evaluate our proposed system in public datasets, will review the best performing systems in the literature.

Bootstrapping techniques consist on algorithms that learn from a few instances of labeled data (seeds) and a big set of unlabeled examples. Among these approaches, we can highlight co-training (Blum & Mitchell, 1998) and their derivatives (Collins & Singer, 1999; Abney, 2002). These techniques are very appropriate for WSD and other NLP tasks because of the wide availability of untagged data and the scarcity of tagged data. However, these systems have not been shown to perform well for fine-grained WSD. In his well-known work, Yarowsky (1995) applied an iterative bootstrapping process to induce a classifier based on Decision Lists. With a minimum set of seed examples, disambiguation results comparable to supervised methods were obtained in a limited set of binary sense distinctions, but this success has not been extended to fine-grained senses.

Recent work on bootstrapping applied to WSD is also reported by Mihalcea (2004) and Pham, Ng, and Lee (2005). In the former, the use of unlabeled data significantly increases the performance of a lexical-sample system. In the latter, Pham et al. apply their WSD classifier to the all-words task in Senseval-2, but targeting words over a threshold of frequency in the Semcor and WSJ corpora. They observe a slight increase in accuracy relying on unlabeled data.

Active learning is used to choose informative examples for hand-tagging, in order to reduce manual cost. In one of the few works directly applied to WSD, Fujii, Inui, Tokunaga, and Tanaka (1998) used selective sampling for the acquisition of examples for the disambiguation of verb senses, in an iterative process with human taggers. The informative examples were chosen following two criteria: maximum number of neighbors in unsupervised data, and minimum similarity with the supervised example set. Another active learning approach is the Open Mind Word Expert (Mihalcea & Chklovski, 2003), which is a project to collect sense-tagged examples from Web users. The system selects the examples to be tagged applying a selective sampling method based on two different classifiers, choosing the unlabeled examples where there is disagreement. The collected data was used in the Senseval-3 English lexical-sample task.

Parallel corpora is another alternative to avoid the need of hand-tagged data. Recently Chan and Ng (2005) built a classifier from English-Chinese parallel corpora. They grouped senses that share the same Chinese translation, and then the occurrences of the word on the English side of the parallel corpora were considered to have been disambiguated and "sense tagged" by the appropriate Chinese translations. The system was successfully evaluated in the all-words task of Senseval-2. However, parallel corpora is an expensive resource to obtain for all target words. A related approach is to use monolingual corpora in a second language and use bilingual dictionaries to translate the training data (Wang & Carroll, 2005). Instead of using bilingual dictionaries, Wang and Martinez (2006) applied machine translation to text snippets in foreign languages back into English and achieved good results on English lexical-sample WSD.

In the automatic acquisition of training examples, an external lexical resource (WordNet, for instance) or a sense-tagged corpus is used to obtain new examples from a very large untagged corpus (e.g. the Web). Leacock et al. (1998) present a method to obtain sense-tagged examples using monosemous relatives from WordNet. Our approach is based on this early work (cf. Section 3). In their algorithm, Leacock et al. (1998) retrieve the same number of examples per each sense, and they give preference to monosemous relatives that consist on a multiword containing the target word. Their experiment is evaluated over 14 nouns with coarse sense-granularity and few senses. The results showed that the monosemous corpus provided precision close to that of hand-tagged data.

Another automatic acquisition approach (Mihalcea & Moldovan, 1999) used information in WordNet (e.g. monosemous synonyms and glosses) to construct queries, which were later fed into the Altavista[2] search engine. Four procedures were used sequentially, in a decreasing order of precision, but with increasing levels of coverage. Results were evaluated by hand, showing that 91% of the examples were correctly retrieved among a set of 1,080 instances of 120 word senses. However, the corpus resulting from the experiment was not used to

[2] http://www.altavista.com

train a real WSD system. Agirre and Martinez (2000), in an early precursor of the work presented here, tried to apply this technique to train a WSD system with unsatisfactory results. The authors concluded that the examples themselves were correct, but that they somehow mislead the ML classifier, providing biased features.

In related work, Mihalcea (2002) generated a sense tagged corpus (GenCor) by using a set of seeds consisting of sense-tagged examples from four sources: (i) Semcor, (ii) WordNet, (iii) examples created using the method above, and (iv) hand-tagged examples from other sources (e.g. the Senseval-2 corpus). By means of an iterative process, the system obtained new seeds from the retrieved examples. In total, a corpus with about 160,000 examples was gathered. However, the evaluation was carried out on the lexical-sample task, showing that the method was useful for a subset of the Senseval-2 testing words (results for 5 words were provided), and without analysing which were the sources of the performance gain. Even if the work presented here uses other techniques, our work can be seen as an extension of this limited study, in the sense that we evaluate on all-words tasks.

These previous works focused on the use of two different kinds of techniques for the automatic acquisition of examples, namely, the use of monosemous relatives alone (Leacock et al., 1998) and the use of a combination of monosemous relatives and glosses (Mihalcea & Moldovan, 1999; Mihalcea, 2002). In all cases the examples are directly used to feed a supervised ML WSD system, but with limited evaluation and no indication that the methods can scale-up. Unfortunately, no direct comparison of the alternative methods and parameters to automatically acquire examples for WSD exists, but we can see a preference to use the Web, as existing corpora would contain very few occurrences of the monosemous terms or gloss fragments.

A closely related area to that of automatic acquisition of examples for WSD is that of enriching knowledge bases with topic signatures. For instance, Agirre et al. (2000) and Agirre, Ansa, Martinez, and Hovy (2001) used the combined monosemous-relatives plus glosses strategy to query Altavista, retrieve the original documents and build lists of related words for each word sense (so called topic signatures). The topic signatures are difficult to evaluate by hand, so they were applied as context vectors to WSD in a straightforward way. Note that the authors did not train a ML algorithm, but rather combined all the examples in one vector per sense. They showed that using the Web compared favorably to using a fixed corpus, but was computationally more costly: the system first needs to query a search engine and then retrieve the original document in order to get an example for the sense. As an alternative, Agirre and Lopez de Lacalle (2004) showed that it is possible to scale up and gather examples for all nouns in WordNet if the query is limited to using monosemous relatives and if the snippets returned by Google are used instead of the whole document.

At this point, Cuadros et al. (2006) set up a systematic framework for the evaluation of the different parameters that affect the construction of topic signatures, including the methods to automatically acquire examples. The study explores a wide range of querying strategies (monosemous synonyms, monosemous relatives at different distances, and glosses, combined using either *and* or *or* operators) on both a particular corpus (the British National Corpus) and the Web. The best results were obtained using Infomap[3] on the British National Corpus and our monosemous relatives method on the Web (Agirre & Lopez de

[3] `http://infomap-nlp.sourceforge.net`

Lacalle, 2004). Contrary to our method, Infomap returns only lists of related words, and thus can not be used to retrieve training examples. These results are confirmed in other experiments reported by Cuadros and Rigau (2006).

All in all, the literature shows that using monosemous relatives and snippets from the Web (Agirre & Lopez de Lacalle, 2004) provides a method to automatically acquire examples which scales up to all nouns in WordNet, and provides topic signatures of better quality than other alternative methods. We will now explain how these examples were acquired.

3. Building a Sense-Tagged Corpus for all Nouns Automatically

In order to build this corpus (which we will refer to as SenseCorpus) we acquired 1,000 Google snippets for each monosemous noun in WordNet 1.6 (including multiwords, e.g. *church building*). Then, for each word sense of an ambiguous noun, we gathered the examples of its monosemous relatives (e.g. for sense #2 of *church*, we gather examples from its relative *church building*). The way to collect the examples is simply by querying the corpus with the word or string of words (e.g. "*church building*"). This method is inspired in the work by Leacock et al. (1998) and, as already mentioned in Section 2, it has been shown to be both efficient and effective in experiments on topic signature acquisition.

The basic assumption of this method is that for a given word sense of the target word, if we had a monosemous synonym of the word sense, then the examples of the synonym should be very similar to those of the target word sense, and could therefore be used to train a classifier of the target word sense. The same idea , to a lesser extent, can be applied to other monosemous relatives, such as direct hyponyms, direct hypernyms, siblings, indirect hyponyms, etc. The expected reliability decreases with the distance in the hierarchy from the monosemous relative to the target word sense.

The actual method to build SenseCorpus is the following. We collected examples from the Web for each of the monosemous relatives. The relatives have an associated number (type), which correlates roughly with the distance to the target word, and indicates their relevance: the higher the type, the less reliable the relative. Synonyms have type 0, direct hyponyms get type 1, and distant hyponyms receive a type number equal to the distance to the target sense. Direct hypernyms get type 2, because they are more general than the target sense, and can thus introduce more noise than direct hyponyms. We also decided to include less reliable siblings, but with type 3. More sophisticated schemes could be tried, such as using WordNet similarity to weight the distance from the target to the relative word. However, we chose this approach to capture the notion of distance for its simplicity, and to avoid testing too many parameters. A sample of monosemous relatives for different senses of *church*, together with its sense inventory in WordNet 1.7 is shown in Figure 1.

In the following subsections we will describe step by step the method to construct the corpus. First we will explain the acquisition of the highest possible amount of examples per sense, and then we will explain different ways to limit the number of examples per sense for better performance.

3.1 Collecting the Examples

The method to collect the examples has been previously published (Agirre & Lopez de Lacalle, 2004), and comprises the following steps:

- **Sense inventory (church)**
  - **Sense 1:** A group of Christians; any group professing Christian doctrine or belief.
  - **Sense 2:** A place for public (especially Christian) worship.
  - **Sense 3:** A service conducted in a church.
- **Monosemous relatives for different senses (of church)**
  - **Synonyms (Type 0)**: *church building* (sense 2), *church service* (sense 3) ...
  - **Direct hyponyms (Type 1)**: *Protestant Church* (sense 1), *Coptic Church* (sense 1) ...
  - **Direct hypernyms (Type 2)**: *house of prayer* (sense 2), *religious service* (sense 3) ...
  - **Distant hyponyms (Type 2,3,4...)**: *Greek Church* (sense 1), *Western Church* (sense 1)...
  - **Siblings (Type 3)**: *Hebraism* (sense 2), *synagogue* (sense 2) ...

Figure 1: Sense inventory and a sample of monosemous relatives in WordNet 1.7 for *church*.

1: We query Google[4] with the monosemous relatives for each sense, and extract the snippets returned by the search engine. All snippets are used (up to 1,000), but some of them are dropped out in the next step.

2: We try to detect full meaningful sentences in the snippets which contain the target word. We first detect sentence boundaries in the snippet and extract the sentence that encloses the target word. Some of the sentences are filtered out, according to the following criteria: length shorter than 6 words, having more non-alphanumeric characters than words divided by two, or having more words in uppercase than in lowercase.

3: The automatically acquired examples contain a monosemous relative of the target word. In order to use these examples to train the classifiers, the monosemous relative (which can be a multiword term) is substituted by the target word. In the case of the monosemous relative being a multiword that contains the target word (e.g. *Protestant Church* for *church*) we can choose not to substitute, because *Protestant*, for instance, can be a useful feature for the first sense of church. We tried both alternatives, and Section 5 will show that we obtain slightly better results if no substitution is applied for such multiwords.

4: For a given word sense, we collect the desired number of examples (see the following section) in order of their type: we first retrieve all examples of type 0, then type 1, etc. up to type 3 until the necessary examples are obtained. We did not collect examples from type 4 upwards. We did not make any distinctions between the relatives from each type. Contrary to Leacock et al. (1998) we do not give preference to multiword relatives containing the target word.

All in all, we have acquired around 150 million examples for the nouns in WordNet using this technique, which are publicly available[5].

[4] We use the off-line XML interface kindly provided by Google for research.

[5] `http://ixa.si.ehu.es/Ixa/resources/sensecorpus`.

3.2 Number of Examples per Sense (Prior)

Previous work (Agirre & Martinez, 2000) has reported that the distribution of the number of examples per word sense (prior for short) has a strong influence in the quality of the results. That is, the results degrade significantly whenever the training and testing samples have different distributions of the senses. It has also been shown that a type-based approach that predicts the majority sense of a word in the domain can provide good performance by itself (McCarthy, Koeling, Weeds, & Carroll, 2004).

As we are extracting examples automatically, we have to decide how many examples we will use for each sense. In order to test the impact of the prior, different settings have been tried:

- No prior: we take an equal amount of examples for each sense.
- Web prior: we take all examples gathered from the Web.
- Automatic ranking: the number of examples is given by a ranking obtained following the method by McCarthy et al. (2004).
- Sense-tagged prior: we take a number of examples proportional to the relative frequency of the word senses in some hand-tagged corpus.

The first method assumes uniform priors. The second assumes that the number of monosemous relatives and their occurrences are correlated to sense importance, that is, frequent senses would have more occurrences of their monosemous relatives. The fourth method uses the information in some hand-tagged corpus, typically Semcor. Note that this last kind of prior requires hand-tagged data, while the rest (including the third method below) are completely unsupervised.

The third method is more sophisticated and deserves some further clarification. McCarthy et al. (2004) present a method to acquire sense priors automatically from a domain corpus. This is a two-step process. The first step is a corpus-based method, which given a target word builds a list of contextually similar words (Lin, 1998) with weights. In this case, the co-occurrence data was gathered from the British National Corpus. For instance, given a target word like *authority*, the list of the topmost contextually similar words include *government*, *police*, *official* and *agency*[6]. The second step ranks the senses of the target word, depending on the scores of a WordNet-based similarity metric (Patwardhan & Pedersen, 2003) relative to the list of contextually similar words. Following with the example, the pairwise WordNet similarity between *authority* and *government* is greater for sense 5 of *authority*, which is evidence that this sense has some prominence in the corpus. The pairwise similarity scores are added, yielding a ranking for the 7 senses of *authority.* Table 2 shows in the column named Auto.MR the normalized scores assigned to each of the senses of *authority* according to this technique.

Table 1 shows the number of examples per type (0,1,...) that are acquired for *church* following the Semcor prior. The last column gives the number of examples in Semcor. Note that the number of examples is sometimes smaller than 1,000 (maximum number of snippets returned by Google in one query). This can be due to rare monosemous relatives, but is

[6] Actual list of words taken from the demo in `http://www.cs.ualberta.ca/~lindek/demos/depsim.htm`.

Sense	**0**	**1**	**2**	**3**	**Total**	**Semcor**
church#1	0	476	524	0	1,000	60
church#2	306	100	561	0	967	58
church#3	147	0	20	0	167	10
Overall	453	576	1,105	0	2,134	128

Table 1: Examples per type (0,1,2,3) that are acquired from the Web for the three senses of *church* following the Semcor prior, and total number of examples in Semcor.

Sense	**Semcor**		**SenseCorpus**								**Senseval test**	
			Web PR		**Auto. MR**		**Semcor PR**		**Semcor MR**			
	#ex	**%**	**#ex**	**%**	**#ex**	**%**	**#ex**	**%**	**#ex**	**%**	**#ex**	**%**
authority#1	18	60.0	338	0.5	138	19.3	338	33.7	324	59.9	37	37.4
authority#2	5	16.7	44932	66.4	75	10.5	277	27.6	90	16.6	17	17.2
authority#3	3	10.0	10798	16.0	93	13.0	166	16.6	54	10.0	1	1.0
authority#4	2	6.7	886	1.3	67	9.4	111	11.1	36	6.7	0	0.0
authority#5	1	3.3	6526	9.6	205	28.6	55	5.5	18	3.3	34	34.3
authority#6	1	3.3	72	0.1	71	9.9	55	5.5	18	3.3	10	10.1
authority#7	0	0.0	4106	6.1	67	9.4	1	0.1	1	0.2	0	0.0
Overall	30	100.0	67657	100.0	716	100.0	1003	100.0	541	100.0	99	100.0

Table 2: Distribution of examples for the senses of *authority* in different corpora. PR (proportional) and MR (minimum ratio) columns correspond to different ways to apply Semcor prior.

usually caused by the sentence extraction and filtering process, which discards around 50% of the snippets.

The way to apply the prior is not straightforward. For illustration, we will focus on the Semcor prior. In our first approach for Semcor prior, we assigned 1,000 examples to the major sense in Semcor, and gave the other senses their proportion of examples. We call this method **proportional** (PR). But in some cases the number of examples extracted will be less than expected by the distribution of senses in Semcor. As a result, the actual number of examples available would not follow the desired distribution.

As an alternative, we computed, for each word, the **minimum ratio** (MR) of examples that were available for a given target distribution *and* a given number of examples extracted from the Web. We observed that this last approach would reflect better the original prior, at the cost of having less examples.

Table 2 presents the different distributions of examples for *authority*. There we can see the Senseval-testing and Semcor distributions, together with the total number of examples in the Web (Web PR); the Semcor proportional distribution (Semcor PR) and minimum ratio (Semcor MR); and the automatic distribution with minimum ratio (Auto MR). Getting a maximum of one thousand examples per monosemous relative allows to get up to 44,932 examples for the second sense (Web PR column), but only 72 for the sixth sense.

Semcor Word	Web prior	Automatic prior	Semcor prior	Semcor Word	Web prior	Automatic prior	Semcor prior
art	15,387	2,610	10,656	grip	20,874	277	2,209
authority	67,657	716	541	hearth	6,682	2,730	1,531
bar	50,925	5,329	16,627	holiday	16,714	1,846	1,248
bum	17,244	4,745	2,555	lady	12,161	884	2,959
chair	24,625	2,111	8,512	material	100,109	6,385	7,855
channel	31,582	10,015	3,235	mouth	648	464	287
child	47,619	791	3,504	nation	608	608	594
church	8,704	6,355	5,376	nature	32,553	9,813	24,746
circuit	21,977	5,095	3,588	post	34,968	8,005	4,264
day	84,448	3,660	9,690	restraint	33,055	2,877	2,152
detention	2,650	511	1,510	sense	10,315	2,176	2,059
dyke	4,210	843	1,367	spade	5,361	2,657	2,458
facility	11,049	1,196	8,578	stress	10,356	3,081	2,175
fatigue	6,237	5,477	3,438	yew	10,767	8,013	2,000
feeling	9,601	945	1,160				
Average	24,137	3,455	4,719				
Total	699,086	100,215	136,874				

Table 3: Number of examples following different sense distributions for the Senseval-2 nouns. Minimum ratio is applied both for the Semcor and automatic priors.

The sixth sense has a single monosemous relative, which is a rare word with few hits in Google, while the second sense has many and frequent monosemous relatives.

Regarding the use of minimum ratio, the table illustrates how MR allows to better approximate the distribution of senses in Semcor: the first sense[7] has 60% in Semcor, but only gets 33.7% in SenseCorpus with the proportional Semcor prior because there are only 338 examples in SenseCorpus for the first sense. In contrast SenseCorpus with minimum ratio using Semcor does assign 59.9% of the examples to the first sense. This better approximation comes at the cost of getting 541 examples for *authority*, in contrast to 1,003 with PR. Note that *authority* occurs only 30 times in Semcor.

The table also shows that for this word the distributions of senses in Semcor and Senseval-test have important differences (sense 5 gets 3.3% and 34.3% respectively), although the most frequent sense is the same. For the Web and automatic distributions, the most salient sense is different from that in Semcor, with the Web prior (Web PR column) assigning only 0.5% to the first sense. Note that the automatic method is able to detect that sense 5 is salient in the test corpus, while Semcor ranks it only 5th. In general, distribution discrepancies similar to those in the table can be observed for the other words in the test set.

To conclude this section, Table 3 shows the number of examples acquired automatically for each word in the Senseval-2 lexical-sample following three approaches: the Web prior, the Semcor prior with minimum ratio, and the Automatic prior with minimum ratio. We can see that retrieving all the examples (Web prior) we get 24,137 examples in average per word; and respectively 4,700 or 3,400 if we apply the Semcor prior or the Automatic prior.

[7] The senses in WordNet are numbered according to their frequency in Semcor, so the first sense in WordNet is paramount to the most frequent sense in Semcor.

3.3 Decision Lists

The supervised learning method used to measure the quality of the corpus is **Decision Lists** (DL). This simple method performs reasonably well in comparison with other supervised methods in Senseval all words (as we will illustrate in Table 6.4), and preliminary experiments showed it to perform better with the automatically retrieved examples than more sophisticated methods like Support Vector Machines or the Vector Space Model. It is well known that learning methods perform differently according to several conditions, as showed for instance by Yarowsky and Florian (2003), who analyzed in depth the performance of various learning methods (including DL) in WSD tasks.

We think that the main reason for DL to perform better in our preliminary experiments is that SenseCorpus is a noisy corpus with conflicting features. Decision Lists use the single most powerful feature in the test context to make predictions, in contrast to other ML techniques, and this could make them perform better in this corpus. Specially in the all-words task, with only a few hand-tagged examples per word in most cases, even the most sophisticate ML algorithms cannot deal with the problem by themselves. While the best systems in the Senseval-3 lexical-sample rely on complex kernel-based methods, in the all-words task the top systems are those that find external ways to deal with the sparseness of data and then apply well-known methods, such as memory based learning or decision trees (Mihalcea & Edmonds, 2004).

The DL algorithm is described by Yarowsky (1994). In this method, the sense s_k with the highest weighted feature f_i is selected, according to its log-likelihood (see Formula 1). For our implementation, we applied a simple smoothing method: for the cases where the denominator is zero, we use 0.1 as the denominator. This is roughly equivalent to assigning a 0.1 probability mass to the rest of senses, and has been shown to be effective enough compared to more complex methods (Yarowsky, 1994; Agirre & Martinez, 2004a).

$$weight(s_k, f_i) = \log(\frac{Pr(s_k|f_i)}{\sum_{j \neq k} Pr(s_j|f_i)}) \tag{1}$$

3.4 Feature Types

The feature types that we extracted from the context can be grouped in three main sets:

Local collocations: bigrams and trigrams formed with the words around the target. These features are constituted by lemmas, word-forms, or PoS tags[8]. Other local features are those formed with the previous/posterior lemma/word-form in the context.

Syntactic dependencies: syntactic dependencies were extracted using heuristic patterns, and regular expressions defined with the PoS tags around the target[9]. The following relations were used: object, subject, noun-modifier, preposition, and sibling.

Topical features: we extract the lemmas of the content words both in the whole sentence and in a ±4-word window around the target. We also obtain salient bigrams in the context, with the methods and the software described by Pedersen (2001).

[8] The PoS tagging was performed with the fnTBL toolkit (Ngai & Florian, 2001).

[9] This software was kindly provided by David Yarowsky's group, from the Johns Hopkins University.

The complete feature set was applied for our main experiments on the all-words Senseval-3 corpus. However, for our initial experiments in the lexical-sample task only local features and topical features (without salient bigrams) were applied.

4. Experimental Setting

We already noted in the introduction that lexical-sample evaluations as defined in Senseval are not realistic: relatively large amounts of training examples are available, those are drawn from the same corpus as the test examples, and both train and test examples are tagged by the same team. Besides, developing a system for a handful of words does not necessarily show that it is scalable. In contrast, all-words evaluations do not provide training data. Supervised WSD systems typically use **Semcor** (Miller et al., 1993) for training. This corpus offers tagged examples for all open-class words occurring in a 350.000 word subset of the balanced Brown corpus, tagged with WordNet 1.6 senses. In contrast to lexical-sample, some polysemous words like *authority* only get a handful of examples (30 in this case, cf. Table 2). Note that the test examples (from Senseval) and Semcor come from different corpora and thus might be related to different domains, topics or genres. An added difficulty is posed by the fact that they have been tagged by different teams of annotators from distinct institutions.

With all this on mind, we designed two sets of experiments: the first set was performed on a sample of nouns (lexical-sample), and it was used to develop and fine-tune the method in basic aspects like the effect of the kinds of features and the importance of the prior. We did not use the training examples, except to measure the impact of the priors. We provide a comparison with state-of-the-art systems.

The second set of experiment was used to show that our method is scalable, useful for any noun, and performs in the state-of-the art of WSD in a realistic setting. We thus selected to apply WSD on all the nouns in running text (all-nouns). In this setting we apply the best configurations obtained from the first set of experiments, and explore the use of SenseCorpus alone, combined with priors from Semcor, and also with training data from Semcor. We provide a comparison of our results with those of state-of-the-art systems.

For lexical-sample evaluation, the test part of the **Senseval-2 English lexical-sample** task was chosen, which consisted on instances of 29 nouns, tagged with WordNet 1.7 senses. The advantage of this corpus was that we could focus on a word-set with enough examples for testing. Besides, it is a different corpus, and therefore the evaluation is more realistic than that made using cross-validation over Semcor. In order to factor out pre-processing and focus on WSD, the test examples whose senses were multiwords or phrasal verbs were removed. Note that they are not as problematic since they can be efficiently detected with other methods in a preprocess.

It is important to note that the training part of Senseval-2 lexical-sample was not used in the construction of the systems, as our goal was to test the performance we could achieve with minimal resources (i.e. those available for any word). We only relied on the Senseval-2 training prior in preliminary experiments on local/topical features, and as an upperbound to compare the performance with other types of priors.

For the all-words evaluation we relied on the **Senseval-3 all-words** corpus (Snyder & Palmer, 2004). The test data for this task consisted of 5,000 words of text. The data was extracted from two Wall Street Journal articles and one excerpt from the Brown Corpus. The texts represent three different domains: editorial, news story, and fiction. Overall, 2,212 words were tagged with WordNet 1.7.1. senses (2,081 if we do not include multiwords). From these, 695 occurrences correspond to polysemous nouns that are not part of multiwords, and these comprise our testing set.

As the rest of Senseval participants, we had an added difficulty in that WordNet versions do not coincide. We therefore used one of the freely available mappings between WordNet versions (Daude, Padró, & Rigau, 2000) to convert the training material from Semcor (tagged with WordNet 1.6 senses) into WordNet 1.7 and WordNet 1.7.1 versions (depending on the target corpus). We preferred to use this mapping rather that relying on other available mappings or converted Semcors. To our knowledge, no comparative evaluation among mappings has been performed, and Daude et al. show that their mapping obtained very high scores in an extensive manual evaluation. Note that the versions of Semcor available in the Web (other than the original one, tagged with WordNet 1.6) have also been obtained using an automatic mapping.

In both lexical-sample and all-nouns settings, we provide a set of baselines, which are based on the most frequent heuristic. This heuristic is known to be hard to beat in WSD, specially for unsupervised systems that do not have access to the priors, and even for supervised systems in the all-nouns setting.

5. Lexical-Sample Evaluation

We performed four sets of experiments in order to study different factors, and compare our performance to other state-of-the-art unsupervised systems in the Senseval-2 lexical-sample task. First we analyzed the results of the systems when using different sets of local and topical features, as well as substituting or not multiwords. The next experiments were devoted to measure the effect of the prior on the performance. After that, we compared our approach with unsupervised systems that participated in Senseval-2. As we mentioned in the introduction, the results obtained in lexical-sample evaluations are not realistic, in that we cannot expect to have hand-tagged data for all words in any target corpus. For this reason we do not report results of supervised systems (which do use the training data). The next section on all-nouns evaluation, which is more realistic, does compare to supervised systems

5.1 Local vs. Topical Features, Substitution

Previous work on automatic acquisition of examples (Leacock et al., 1998) has reported lower performance when using local collocations formed by PoS tags or closed-class words. In contrast, Kohomban and Lee (2005), in a related approach, used only local features for WSD because they discriminated better between senses. Given the fact that SenseCorpus has also been constructed automatically, and the contradictory results on those previous works, we performed an initial experiment comparing the results using local features, topical features, and a combination of both. In this case we used SenseCorpus with Senseval training prior, distributed according to the MR approach, and always substituting the target word. The results (per word and overall) are given in Table 4.

Word	**Local Feats.**			**Topical Feats.**	**Combined Subst.**	**Combined No Subst.**
	Coverage	**Precision**	**Recall**	**Recall**	**Recall**	**Recall**
art	94.4	57.4	**54.2**	45.6	47.0	44.9
authority	93.4	51.2	**47.8**	43.2	46.2	46.2
bar	98.3	53.0	52.1	55.9	**57.2**	**57.2**
bum	100.0	81.2	81.2	**87.5**	85.0	85.0
chair	100.0	88.7	**88.7**	**88.7**	**88.7**	**88.7**
channel	73.5	54.0	39.7	53.7	55.9	**57.4**
child	100.0	56.5	56.5	55.6	56.5	**58.9**
church	100.0	67.7	**67.7**	51.6	54.8	51.6
circuit	88.7	51.1	45.3	54.2	56.1	**58.0**
day	98.6	60.2	59.4	54.7	56.8	**60.4**
detention	100.0	87.5	**87.5**	**87.5**	**87.5**	**87.5**
dyke	100.0	89.3	**89.3**	**89.3**	**89.3**	**89.3**
facility	98.2	29.1	**28.6**	21.4	21.4	21.4
fatigue	100.0	82.5	**82.5**	**82.5**	**82.5**	**82.5**
feeling	100.0	55.1	55.1	**60.2**	**60.2**	**60.2**
grip	100.0	19.0	19.0	38.0	**39.0**	38.0
hearth	100.0	73.4	73.4	**75.0**	**75.0**	**75.0**
holiday	100.0	96.3	**96.3**	**96.3**	**96.3**	**96.3**
lady	100.0	80.4	**80.4**	73.9	73.9	73.9
material	100.0	43.2	43.2	**44.2**	43.8	42.9
mouth	100.0	36.8	36.8	38.6	**39.5**	**39.5**
nation	100.0	80.6	**80.6**	**80.6**	**80.6**	**80.6**
nature	100.0	44.4	**44.4**	39.3	40.7	40.7
post	98.3	44.7	**43.9**	40.5	40.5	40.5
restraint	79.5	37.1	29.5	**37.5**	37.1	37.1
sense	93.0	62.5	**58.1**	37.2	38.4	48.8
spade	100.0	74.2	**74.2**	72.6	**74.2**	**74.2**
stress	100.0	53.9	**53.9**	46.1	48.7	48.7
yew	100.0	81.5	**81.5**	**81.5**	**81.5**	**81.5**
Overall	96.7	58.5	56.5	56.0	57.0	**57.5**

Table 4: Results per feature type (local, topical, and combination), using SenseCorpus with Senseval-2 training prior (MR). Coverage and precision are given only for local features (topical and combination have full coverage). Combination is shown for both substitution and no substitution options. The best recall per word is given in bold.

In this experiment, we observed that local collocations achieved the best precision overall, but the combination of all features obtained the best recall. Local features achieve 58.5% precision for 96.7% coverage overall[10], while the topical and combined features have full-coverage. The table shows clear differences in the results per word, a fact which is also known for other algorithms using real training data (Yarowsky & Florian, 2003). This variability is another important factor to focus on all-words settings, where large numbers of different words are involved.

We also show the results for not substituting the monosemous relative by the target word when the monosemous relative is a multiword. We can see that the results are mixed, but that there is an slight overall improvement if we choose not to substitute in those cases. For the following experiments, we chose to work with the combination of all features with no substitution, as it achieved the best overall recall.

5.2 Impact of Prior

In order to evaluate the acquired corpus, our first task was to analyze the impact of the prior. As we mentioned in Section 3.2, when training Decision Lists with the examples in SenseCorpus, we need to decide the amount of examples for each sense (what can be seen as the estimation of the prior probabilities of the senses).

Table 5 shows the recall[11] attained by DL with each of the four proposed methods to estimate the priors for each target word, plus the use of the training part of Senseval-2 lexical sample to estimate the prior. Note that this last estimation method is not realistic, as one cannot expect to have hand-tagged data for all words in a given target corpus, and should thus be taken as an upperbound. In fact it is presented in this section for completeness, and will not be used for comparison with other systems.

The results show constant improvement from the less informative priors to the most informed ones. Among the three unsupervised prior estimation methods, the best results are obtained with the automatic ranking, and the worst by the uniform distribution ("no prior" column), with the distribution of examples as returned by SenseCorpus ("Web prior") in the middle. Estimating the priors from hand-tagged data improves the results considerably, even when the target corpus and estimation corpus are different ("Semcor"), but the best results overall are obtained when the priors are estimated from the training part of Senseval-2 lexical-sample dataset. The results word by word show that each word behaves differently, which is a well-known behavior in WSD. Note that for all priors except the most informed one a number of words have performances below 10%, which might indicate that DL trained on SenseCorpus is very sensitive to badly estimated priors.

Table 6 shows the overall results from Table 5, together with those obtained using the prior on its own ("prior only"). The results show that the improvement attained by training on SenseCorpus is most prominent for the unsupervised priors (from 6.5 to 19.7 percentage points), with lower improvements (around 2.0 percentage points) for the priors estimated from hand-tagged corpora. These results show clearly that the acquired corpus has use-

[10] Note that due to the sparse data problem, some test examples might not have any feature in common with the training data. In those cases the DL algorithm does not return any result, and thus the coverage can be lower than 100%

[11] All the results in the following tables are given as recall, as the coverage is always 100% and precision equals to recall in this case.

Word	Unsupervised			Minimally-Supervised	
	No prior	Web prior	Autom. ranking	Semcor prior	Senseval-2 prior
art	34.0	**61.1**	45.6	55.6	44.9
authority	20.9	22.0	40.0	41.8	**46.2**
bar	24.7	52.1	26.4	51.6	**57.2**
bum	36.7	18.8	57.5	5.0	**85.0**
chair	61.3	62.9	69.4	**88.7**	**88.7**
channel	42.2	28.7	30.9	16.2	**57.4**
child	40.3	1.6	34.7	54.0	**58.9**
church	43.8	**62.1**	49.7	48.4	51.6
circuit	44.3	52.8	49.1	41.5	**58.0**
day	15.3	2.2	12.5	48.0	**60.4**
detention	52.1	16.7	**87.5**	52.1	**87.5**
dyke	**92.9**	89.3	80.4	**92.9**	89.3
facility	19.6	**26.8**	22.0	**26.8**	21.4
fatigue	58.8	73.8	75.0	**82.5**	**82.5**
feeling	27.2	51.0	42.5	**60.2**	**60.2**
grip	11.3	8.0	28.2	16.0	**38.0**
hearth	57.8	37.5	60.4	**75.0**	**75.0**
holiday	70.4	7.4	72.2	**96.3**	**96.3**
lady	24.3	79.3	23.9	**80.4**	73.9
material	51.7	50.8	52.3	**54.2**	42.9
mouth	39.5	39.5	46.5	**54.4**	39.5
nation	**80.6**	**80.6**	**80.6**	**80.6**	**80.6**
nature	21.9	44.4	34.1	**46.7**	40.7
post	36.8	**47.4**	**47.4**	34.2	40.5
restraint	26.3	9.1	31.4	27.3	**37.1**
sense	44.8	18.6	41.9	47.7	**48.8**
spade	74.2	66.1	**85.5**	67.7	74.2
stress	38.6	**52.6**	27.6	2.6	48.7
yew	70.4	**85.2**	77.8	66.7	81.5
Overall	38.0	39.8	43.2	49.8	**57.5**

Table 5: Performance (recall) of SenseCorpus on the 29 nouns in Senseval-2 lexical-sample, using different priors to train DL. Best results for each word in bold.

ful information about the word senses, and that the estimation of the prior is extremely important.

Prior	**Type**	**Only prior**	**SenseCorpus**	**Diff.**
no prior	unsupervised	18.3	38.0	+19.7
Web prior		33.3	39.8	+6.5
autom. ranking		36.1	43.2	+7.1
Semcor prior	minimally-supervised	47.8	49.8	+2.0
Senseval2 prior		55.6	57.5	+1.9

Table 6: Performance (recall) on the nouns in Senseval-2 lexical-sample. In each row, results for a given prior on its own, of SenseCorpus using that prior, and the difference between both.

Method	Type	Recall
SenseCorpus (Semcor prior)	minimally-supervised	**49.8**
UNED		45.1
SenseCorpus (Autom. prior)	unsupervised	**43.3**
Kenneth_Litkowski-clr-ls		35.8
Haynes-IIT2		27.9
Haynes-IIT1		26.4

Table 7: Results for nouns of our best minimally supervised and fully unsupervised systems (in bold) compared to the unsupervised systems that took part in Senseval-2 lexical-sample.

5.3 Comparison with other Systems

At this point, it is important that we compare the performance of our DL-based approach to other systems in the state of the art. In this section we compare our best unsupervised system (the one using Automatic ranking) and the minimally unsupervised system (using Semcor prior) with those systems participating on Senseval-2 that were deemed as unsupervised. In order to have the results of the other systems, we used the resources available from the Senseval-2 competition, where the answers of the participating systems in the different tasks were available[12]. This made possible to compare the results on the same test data, set of nouns and occurrences.

From the 5 systems presented in the Senseval-2 lexical-sample task as unsupervised, the WASP-Bench system relied on lexicographers to hand-code information semi-automatically (Tugwell & Kilgarriff, 2001). This system does not use the training data, but as it uses manually coded knowledge we think it falls in the supervised category.

The results for the other 4 systems and our own are shown in Table 7. We classified the UNED system (Fernandez-Amoros, Gonzalo, & Verdejo, 2001) as minimally supervised. It does not use hand-tagged examples for training, but some of the heuristics that are applied by the system rely on the prior information available in Semcor. The distribution of senses is used to discard low-frequency senses, and also to choose the first sense as a back-off strategy. On the same conditions, our minimally supervised system attains 49.8% recall, nearly 5 points better.

The rest of the systems are fully unsupervised, and they perform significantly worse than our unsupervised system.

6. All Nouns Evaluation

As we explained in the introduction, the main goal of this research is to develop a WSD system that is able to tag all nouns in context, not only a sample of them. In the previous section we explored different settings for our system, adjusting them according to the results for a handful of words on a lexical-sample task.

[12] http://www.senseval.org

In this section we will test SenseCorpus in the 695 occurrences of polysemous nouns present in the Senseval-3 all-words task, and compare our results with the performance of the systems that participated in the competition. We also present an analysis of the results according to the frequency of the target nouns.

We have developed three different systems, all based on SenseCorpus, but with different requirements of external information. The less informed system is the unsupervised system (called SenseCorpus-U), which does not use any hand-coded corpus or prior extracted therein. This system relies on the examples in SenseCorpus following the Automatic Ranking (McCarthy et al., 2004) to train the DL (see Section 3.2). The following system is minimally-supervised (SenseCorpus-MS), in the sense that it uses the priors obtained from Semcor to define the distribution of examples from SenseCorpus that are fed into the DL. Lastly, the most informed system trains the DL with the hand-tagged examples from Semcor and SenseCorpus (following the Semcor prior), and is known as SenseCorpus-S. The three systems follow a widely used distinction among unsupervised, minimally-supervised and supervised systems, and we will compare each of them to similar systems that participated on Senseval-3.

These systems respond to realistic scenarios. The unsupervised system is called for in case of languages for which no all-words hand-tagged corpus exists, or in cases where the priors coming from Semcor are not appropriate, as in domain-specific corpora. The minimally supervised system is useful when there is no hand-tagged corpora, but when there is some indication of the distribution of senses. Lastly, the supervised system (SenseCorpus-S) shows the performance of SenseCorpus on the currently available conditions for English, that is, when an all-words corpus of limited size is available.

In order to measure the real contribution of SenseCorpus, we compare our three systems to each of the following baselines: SenseCorpus-U vs. the first sense according to the automatically obtained ranking, SenseCorpus-MS vs. the most frequent sense in Semcor, and SenseCorpus-S vs. the Decision Lists trained on Semcor. In order to judge the significance of the improvements, we applied one-tail paired t-test.

6.1 Comparison with Unsupervised Systems in Senseval-3

¿From the systems that participated in the all-words task only three did not rely on any hand-tagged corpora (not even for estimating prior information). We compare the performance of those systems with our unsupervised system SenseCorpus-U in Table 8. In order to make a fair comparison with respect to the participants, we removed the answers that did not correctly guess the lemma of the test instance (discarding errors when pre-processing the Senseval-3 XML data).

We can see that one of the participating systems was the automatic ranking by McCarthy et al. (2004) that we used as a baseline. Although we were able to improve this system, our results are below the best unsupervised system (IRST-DDD-LSI) (Strapparava, Gliozzo, & Giuliano, 2004). Surprisingly, this unsupervised method is able to obtain better performance on this dataset than the version that relies on Semcor frequencies (IRST-DDD-0, see next subsection), but this discrepancy is not explained by the authors. The reasons for the remarkable results of IRST-DDD-LSI are not clear, and subsequent publications by the authors do not shed any light on it.

Code	Method	Attempt.	Prec.	Rec.	F	p-value
IRST-DDD-LSI	LSI	570	64.6	52.9	58.2	0.001
SenseCorpus-U	**Decision Lists**	**680**	**45.5**	**44.4**	**45.0**	–
AutoPS (Baseline)	Automatic Rank.	675	44.6	43.3	43.9	0.001
DLSI-UA	WordNet Domains	648	27.8	25.9	26.8	0.000

Table 8: Performance of all unsupervised systems participating in Senseval-3 all-words for the 695 polysemous nouns, accompanied by p-values of the one tailed paired t-test with respect to our unsupervised system (in bold).

Code	Method	Attempt.	Prec.	Rec.	F	p-value
SenseCorpus-MS	**DL**	**695**	**63.9**	**63.9**	**63.9**	–
MFS (Baseline)	MFS	695	62.7	62.7	62.7	0.044
IRST-DDD-00	Domain-driven	669	55.6	53.5	54.5	0.000
Clr04-aw	Dictionary clues	576	58.7	48.6	53.2	0.000
KUNLP	Similar relative in WordNet	628	54.2	49.0	51.5	0.000
IRST-DDD-09	Domain-driven	346	69.7	34.7	46.3	0.000

Table 9: Performance of all minimally supervised systems participating in Senseval-3 all-words for the 695 polysemous nouns, accompanied by p-values of the one tailed paired t-test with respect to SenseCorpus-MS (in bold).

The improvement over the baseline is lower here than in the lexical-sample case, but it is significant at the 0.99 level (significance is 1−p-value). In order to explore the reasons for this, we performed further experiments separating the words in different sets according to their frequency in Semcor, as reported below in Section 6.4.

6.2 Comparison with Minimally Supervised Systems in Senseval-3

There were four systems in Senseval that used Semcor to estimate the sense distribution, without using the examples of each word for training. We show the performance of these systems, together with our own and the most frequent sense baseline in Table 9.

The results show that the SenseCorpus examples are able to obtain the best performance of this kind of systems, well above the rest. The improvement over the Semcor MFS baseline is significant at the 0.96 level.

6.3 Comparison with Supervised Systems in Senseval-3

Most of the systems that participated in the all-words task were supervised systems that relied mainly on Semcor. In Table 10 we present the results of the top 10 competing systems and our system, trained on SenseCorpus and Semcor. We also include the DL system when trained only in Semcor, as a baseline.

The results show that using SenseCorpus we are able to obtain a significant improvement of 2.9% points in F-score over the baseline. This score places our system as second, close

Code	Method	Attempt.	Prec.	Rec.	F	p-value
SenseLearner	Syntactic Patterns	695	65.9	65.9	65.9	0.313
SenseCorpus-S	**DL**	**695**	**65.3**	**65.3**	**65.3**	–
LCCaw		695	65.3	65.3	65.3	0.166
kuaw.ans		695	64.8	64.7	64.7	0.115
R2D2English	Ensemble	695	64.5	64.5	64.5	0.054
GAMBL-AW	Optim.,TiMBL	695	63.3	63.3	63.3	0.013
upv-eaw.upv-eaw2		695	63.3	63.3	63.3	0.014
Meaning	Ensemble	695	63.2	63.2	63.2	0.009
upv-eaw.upv-eaw		695	62.9	62.9	62.9	0.007
Prob5		691	62.8	62.4	62.6	0.007
Semcor baseline	**DL**	**695**	**62.4**	**62.4**	**62.4**	0.006
UJAEN2		695	62.4	62.4	62.4	0.002

Table 10: Performance of the top 10 supervised systems participating in Senseval-3 all-words for the 695 polysemous nouns, accompanied by p-values of the one tailed paired t-test with respect to SenseCorpus-S (in bold).

to the best system for all-nouns. The statistical significance tests score below 90% for the top 4 systems, and over 95% for the rest of systems. This means that our system performs similar to the top three systems, but significantly better than the rest.

6.4 Analysis of the Performance by Word Frequency

In previous sections we observed that different words achieve different rates of accuracy. For instance, the lexical-sample experiments showed that the precision of the unsupervised system ranged between 12.5% and 87.5% (cf. Table 5). Clearly, there are some words whose performance is very low when using SenseCorpus. In this section, we will group the nouns in the Senseval-3 all-nouns task according to their frequency to see whether there is a correlation between the frequency of the words and the performance of our system. Our goal is to identify sets of words that can be disambiguated with higher accuracy by this method. This process would allow us to previously detect the type of words our system can be applied to, thus providing a better tool to work in combination with other WSD systems that exploit other properties of language.

For this study, we created separate word sets according to their frequency of occurrence in Semcor. Table 11 shows the different word-sets, with their frequency ranges, the number of nouns in each range, and the average polysemy. We can see that the most frequent words tend to be also the most polysemous. In the case of supervised systems, polysemy and number of training examples tend to compensate each other, yielding good results for those kinds of words. That is, polysemous words are more difficult to disambiguate, but they also have more examples to train in Semcor (Agirre & Martinez, 2000).

Table 12 shows the results for different frequency ranges for the top unsupervised systems in Senseval-3, together with our method. We can see that for all the systems the performance is very low in the high-frequency range. The best performing system (IRST-DDD-LSI) profits from the use of a threshold and leaves many of these instances unanswered. Regarding the improvement of SenseCorpus-U over the Automatic Ranking baseline (Au-

Range	#Nouns	Avg. Polysemy
0–10	207	3.6
11–20	101	5.1
21–40	89	6.1
41–60	88	6.6
61–80	54	6.9
81–100	31	9.3
101–	125	9.6
Overall	695	5.4

Table 11: Number of noun occurrences in each of the frequency ranges (in Semcor), with average polysemy.

	DLSI-UA		IRST-DDD-LSI		SenseCorpus-U		AutoPS	
	Att.	F-sc.	Att.	F-sc.	Att.	F-sc.	Att.	F-sc
0–10	188	**35.98**	195	67.13	198	**62.77**	198	**62.68**
11–20	96	34.50	91	**69.77**	98	58.90	98	49.25
21–40	75	15.82	81	57.65	89	25.50	86	26.24
41–60	82	19.97	75	55.21	85	42.84	85	42.75
61–80	54	22.20	50	57.69	54	35.80	54	31.50
81–100	31	9.70	19	36.02	31	23.70	31	29.00
101–	122	24.30	59	35.85	125	29.60	123	31.44
overall	648	26.83	570	58.22	680	45.00	675	43.95

Table 12: Results of each of the unsupervised systems in Senseval-3 all words, as evaluated on the nouns in each Semcor frequency range. Att. stands for number of words attempted at each range. Best F-score per system given in bold.

toPS), the best results are obtained in the low-frequency range (0-20), when the baseline scores in the 50-60% F-score range. The results of SenseCorpus are lower than the baseline for words with frequency higher than 80. This suggests that the system is more reliable for low-frequency words, and a simple threshold that takes into account the frequency of words would be indicative of the performance we can expect. The same behavior is also apparent in the other unsupervised systems, which shows that this is a weak spot for this kind of systems. We think that future research should focus on those high frequency words.

7. Discussion

In this work we have implemented and evaluated an all-words WSD system for nouns that is able to reach state-of-the-art performance in all three supervised, unsupervised and semi-supervised settings. We have produced different systems combining SenseCorpus with different priors and the actual examples from Semcor. The supervised system, trained with both hand-tagged (Semcor) and automatically obtained corpora, reaches an F-score of 65.3%, and would rank second in the Senseval-3 all-nouns test data. The semi-supervised system, using the priors from Semcor but no manually-tagged examples, would rank first on its class, and the unsupervised system second. In all cases, SenseCorpus improves over the baselines.

The results are remarkable. If we compare our system to those which came out first in the unsupervised and supervised settings, we see that each of them uses a completely different strategy. On the contrary, our system, using primarily the automatically acquired examples, is able to perform in the top ranks in all three settings.

In any case, a deep gap exists among the following three kinds of systems: (i) Supervised systems with specific training (e.g. Senseval lexical-sample systems), (ii) Supervised systems with all-words training (e.g. those trained using Semcor), and (iii) Unsupervised systems. Our algorithm has been implemented as an all-words supervised system, and as an unsupervised system. Although our implementations obtain state-of-the-art performance in their categories, there are different issues that could be addressed in order to close these gaps, and make all-words unsupervised performance closer to those of the supervised systems.

We identified three main sources of error: the low quality of the relatives applied to some words, the different distributions of senses in training and testing data, and the low performance on high-frequency (and highly polysemous) words. We examine each of them in turn.

The algorithm suffers from the noise introduced by relatives that are far from the target word, and do not share the local context with it. Better filtering would be required to alleviate this problem, and one way to do this could be to retrieve examples only when they share part of the local context with the target word and discard other examples. Another interesting aspect of this problem would be to identify the type of words that achieve low performance with SenseCorpus. We already observed that high-frequency words obtain low performance, and another study on performance according to the type of relatives would be useful for a better application of the algorithm.

In order to deal with words that do not have close WordNet relatives, another source of examples would be to use distributionally similar words. The words would be obtained by methods such as the one presented by Lin (1998), and the retrieved examples would be linked to the target senses using the WordNet similarity package (Patwardhan & Pedersen, 2003).

The second main problem of systems that rely on automatic acquisition is the fact that the sense distributions in training and test data can be very different, and this seriously affects the performance. Our system relies on an automatically-obtained sense ranking to alleviate this problem. However, some words still get too many examples for senses that are not relevant in the domain. In preliminary experiments, we observed the benefit of using heuristics to filter out these senses, such as using the number of close relatives in WordNet, with promising results.

Finally, a third problem is observed in Section 6.4, which is the fact that high-frequency words do not profit from automatically acquired examples. For most unsupervised methods, these frequent (and very polysemous) words get low performances, and threshold-based systems usually discard answering them. A straightforward way to improve the F-score of our system would be to apply a threshold to discard these words and apply another method or back-off strategy on them.

All in all, detecting the limitations of the system can give us important clues to work towards an accurate unsupervised all-words system. The literature shows that no single unsupervised system is able to perform well for all words. If we were able to identify the type of words that were more suited to different algorithms and heuristics, the integration of these algorithms into one single combined system could be the way to go. For instance, we could detect in which cases the relatives of a target word are too different to apply the SenseCorpus approach, or the cases where the automatic ranking has not enough evidence. We have also observed that simple heuristics such as the number of close relatives in WordNet can be successfully applied to some sets of words. Meta-learning techniques (Vilalta & Drissi, 2002) could be very useful to exploit the strengths of unsupervised systems.

8. Conclusions and Future Work

This paper presents evidence showing that a proper use of automatically acquired examples allows for state-of-the-art performance on WSD of nouns. We have gathered examples for all nouns in WordNet 1.6 in a resource called SenseCorpus, amounting to 150 million examples, and made this resource publicly available to the community.

We have used the examples to train a supervised WSD system, in a variety of settings: on its own, combined with prior information coming from different sources, or combined with training examples from Semcor. Depending on the knowledge used, we are able to build, respectively, an unsupervised system that has not seen any hand-labeled training data, a semisupervised one that only sees the priors in a generic hand-labeled corpus (Semcor), and a fully-supervised system that also uses the generic hand-labeled corpus (Semcor) as training data.

The evaluation in both lexical-sample and all-words settings has shown that SenseCorpus improves over commonly used baselines in all combinations, and achieves state-of-the-art performance in the all-words Senseval-3 evaluation set for nouns. Previous work on automatic example acquisition has been evaluated on a handful of words. In contrast we have shown that we are able to scale up to all nouns producing excellent results. In the way, we have learned that the use of the prior of the senses is crucial to apply the acquired examples effectively.

In the discussion we have outlined different ways to overcome the limitations of our system, and each of the proposed lines could improve significantly the current performance. Although the recent literature shows that there is no unsupervised system that performs with high precision for all words, we believe that the different systems complement each other, as they usually perform well for different sets of words. From a meta-learning perspective, we could build a word-expert system that is able to apply the best knowledge source for the problem: SenseCorpus, hand-tagged examples, simple heuristics, or other unsupervised algorithms that can be incorporated.

For future work, aside from the proposed improvements, we think that it would be interesting to apply the method to other testbeds. In order to be applied, the monosemous relative method requires an ontology and a raw corpus. Such resources can be found in many specific domains, such as Biomedicine, that do not have the fine-grainedness of WordNet, and could lead to more practical applications.

Acknowledgments

This work has been partially financed by the Ministry of Education (KNOW project, ICT-2007-211423) and the Basque Government (consolidated research groups grant, IT-397-07). Oier Lopez de Lacalle was supported by a PhD grant from the Basque Government. David Martinez was funded by the Australian Research Council, grant no. DP0663879.

References

Abney, S. (2002). Bootstrapping. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, ACL'02*, Philadelphia.

Agirre, E., Ansa, O., Martinez, D., & Hovy, E. (2000). Enriching very large ontologies using the WWW. In *Proceedings of the Ontology Learning Workshop, organized by ECAI*, Berlin (Germany).

Agirre, E., Ansa, O., Martinez, D., & Hovy, E. (2001). Enriching WordNet concepts with topic signatures. In *Procceedings of the SIGLEX workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations. In conjunction with NAACL.*

Agirre, E., & Edmonds, P. (Eds.). (2006). *Word Sense Disambiguation: Algorithms and Applications*. Springer.

Agirre, E., & Lopez de Lacalle, O. (2004). Publicly available topic signatures for all WordNet nominal senses. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, Lisbon, Portugal.

Agirre, E., & Martinez, D. (2000). Exploring automatic word sense disambiguation with Decision Lists and the Web. In *Procedings of the COLING 2000 Workshop on Semantic Annotation and Intelligent Content*, Luxembourg.

Agirre, E., & Martinez, D. (2004a). Smoothing and word sense disambiguation. In *Proceedings of Expaa for Natural Language Processing (EsTAL)*, Alicante, Spain.

Agirre, E., & Martinez, D. (2004b). Unsupervised WSD based on automatically retrieved examples: The importance of bias. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the 11h Annual Conference on Computational Learning Theory*, pp. 92–100, New York. ACM Press.

Chan, Y., & Ng, H. (2005). Scaling up word sense disambiguation via parallel texts. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI 2005)*, Pittsburgh, Pennsylvania, USA.

Collins, M., & Singer, Y. (1999). Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, EMNLP/VLC'99*, College Park, MD, USA.

Cuadros, M., Padró, L., & Rigau, G. (2006). An empirical study for automatic acquisition of topic signatures. In *Proceedings of Third International WordNet Conference*, Jeju Island (Korea).

Cuadros, M., & Rigau, G. (2006). Quality assessment of large scale knowledge resources. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pp. 534–541, Sydney, Australia. Association for Computational Linguistics.

Daconta, M., Obrst, L., & Smith, K. (2005). *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management.* John Wiley & Sons.

Dale, R., Moisl, H., & Somers, H. (2000). *Handbook of Natural Language Processing.* Marcel Dekker Inc.

Daude, J., Padró, L., & Rigau, G. (2000). Mapping WordNets using structural information. In *38th Anual Meeting of the Association for Computational Linguistics (ACL'2000)*, Hong Kong.

Edmonds, P., & Kilgarriff, A. (2002). *Natural Language Engineering, Special Issue on Word Sense Disambiguation Systems.* No. 8 (4). Cambridge University Press.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database.* MIT Press.

Fernandez-Amoros, D., Gonzalo, J., & Verdejo, F. (2001). The UNED systems at Senseval-2. In *Proceedings of the SENSEVAL-2 Workshop. In conjunction with ACL*, Toulouse, France.

Fujii, A., Inui, K., Tokunaga, T., & Tanaka, H. (1998). Selective sampling for example-based word sense disambiguation. In *Computational Linguistics*, No. 24 (4), pp. 573–598.

Humphreys, L., Lindberg, D., Schoolman, H., & Barnett, G. (1998). The Unified Medical Language System: An informatics research collaboration. *Journal of the American Medical Informatics Association*, *1*(5).

Ide, N., & Veronis, J. (1998). Introduction to the special issue on word sense disambiguation: The state of the art. *Computational Linguistics*, *24*(1), 1–40.

Jurafsky, D., & Martin, J. (2000). *An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice-Hall, Upper Saddle River, NJ 07458.

Kim, S.-B., Seo, H.-C., & Rim, H.-C. (2004). Information retrieval using word senses: root sense tagging approach. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 258–265, New York, NY, USA. ACM.

Kohomban, U., & Lee, W. (2005). Learning semantic classes for word sense disambiguation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05).*

Leacock, C., Chodorow, M., & Miller, G. A. (1998). Using corpus statistics and WordNet relations for sense identification. In *Computational Linguistics*, Vol. 24, pp. 147–165.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of COLING-ACL*, Montreal, Canada.

Liu, S., Liu, F., Yu, C., & Meng, W. (2004). An effective approach to document retrieval via utilizing WordNet and recognizing phrases. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 266–272, New York, NY, USA. ACM.

Manning, C. D., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

McCarthy, D., Koeling, R., Weeds, J., & Carroll, J. (2004). Finding predominant word senses in untagged text. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, Barcelona, Spain.

Mihalcea, R. (2002). Bootstrapping large sense tagged corpora. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC)*, Las Palmas, Spain.

Mihalcea, R. (2004). Co-training and self-training for word sense disambiguation. In *Proceedings of the Conference on Natural Language Learning (CoNLL 2004)*, Boston, USA.

Mihalcea, R., & Chklovski, T. (2003). Open Mind Word Expert: Creating large annotated data collections with Web users' help. In *Proceedings of the EACL 2003 Workshop on Linguistically Annotated Corpora (LINC 2003)*, Budapest, Hungary.

Mihalcea, R., Chklovski, T., & Killgariff, A. (2004). The Senseval-3 English lexical sample task. In *Proceedings of the 3rd ACL workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL)*, Barcelona, Spain.

Mihalcea, R., & Edmonds, P. (2004). *Senseval-3, Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. The Association for Computational Linguistics.

Mihalcea, R., & Moldovan, D. (1999). An automatic method for generating sense tagged corpora. In *Proceedings of AAAI-99*, Orlando, FL.

Miller, G. A., Leacock, C., Tengi, R., & Bunker, R. (1993). A semantic concordance. In *Proceedings of the ARPA Human Language Technology Workshop*, pp. 303–308, Princeton, NJ. distributed as *Human Language Technology* by San Mateo, CA: Morgan Kaufmann Publishers.

Ngai, G., & Florian, R. (2001). Transformation-Based Learning in the fast lane. In *Proceedings of the Second Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 40–47, Pittsburgh, PA, USA.

Patwardhan, S., & Pedersen, T. (2003). The cpan wordnet::similarity package. In *http://search.cpan.org/author/SID/WordNet-Similarity-0.03/*.

Pedersen, T. (2001). A decision tree of bigrams is an accurate predictor of word sense. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-01)*, Pittsburgh, PA.

Pham, T. P., Ng, H. T., & Lee, W. S. (2005). Word sense disambiguation with semi-supervised learning. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI 2005)*, pp. 1093–1098, Pittsburgh, Pennsylvania, USA.

Ravin, Y., & Leacock, C. (2001). *Polysemy: Theoretical and Computational Approaches.* Oxford University Press.

Resnik, P. (2006). Word sense disambiguation in natural language processing applications. In Agirre, E., & Edmonds, P. (Eds.), *Word Sense Disambiguation*, chap. 11, pp. 299–337. Springer.

Snyder, B., & Palmer, M. (2004). The English all-words task. In *Proceedings of the 3rd ACL workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL)*, Barcelona, Spain.

Stevenson, M. (2003). *Word Sense Disambiguation: The Case for Combining Knowledge Sources.* CSLI Publications, Stanford, CA.

Stevenson, M., & Clough, P. (2004). Eurowordnet as a resource for cross-language information retrieval. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, Lisbon, Portugal.

Strapparava, C., Gliozzo, A., & Giuliano, C. (2004). Pattern abstraction and term similarity for word sense disambiguation: IRST at Senseval-3. In *Proceedings of the 3rd ACL workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL)*, Barcelona, Spain.

Tugwell, D., & Kilgarriff, A. (2001). WASP-Bench: a lexicographic tool supporting word sense disambiguation. In *Proceedings of the SENSEVAL-2 Workshop. In conjunction with ACL-2001/EACL-2001*, Toulouse, France.

Vickrey, D., Biewald, L., Teyssier, M., & Koller, D. (2005). Word-sense disambiguation for machine translation. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing.*

Vilalta, R., & Drissi, Y. (2002). A perspective view and survey of meta-learning. In *Artificial Intelligence Review*, No. 18 (2), pp. 77–95.

Vossen, P., Rigau, G., Alegría, I., Agirre, E., Farwell, D., & Fuentes, M. (2006). Meaningful results for information retrieval in the MEANING project. In *Proceedings of Third International WordNet Conference*, Jeju Island, Korea.

Wang, X., & Carroll, J. (2005). Word sense disambiguation using sense examples automatically acquired from a second language. In *Proceedings of the joint Human Language Technologies and Empirical Methods in Natural Language Processing conference*, Vancouver, Canada.

Wang, X., & Martinez, D. (2006). Word sense disambiguation using automatically translated sense examples. In *Proceedings of EACL 2006 Workshop on Cross Language Knowledge Induction*, Trento, Italy.

Weeber, M., Mork, J., & Aronson, A. (2001). Developing a test collection for biomedical word sense disambiguation. In *Proceedings of AMIA Symposium*, pp. 746–750.

Yarowsky, D. (1994). Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pp. 88–95, Las Cruces, NM.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pp. 189–196, Cambridge, MA.

Yarowsky, D., & Florian, R. (2003). Evaluating sense disambiguation across diverse parameter spaces. *Natural Language Engineering*, *8*(2), 293–310.