# Using LazyBoosting for Word Sense Disambiguation

**G. Escudero, L. Màrquez and G. Rigau**
TALP Research Center
Universitat Politècnica de Catalunya
Jordi Girona Salgado, 1–3
Barcelona, Catalonia, Spain
{escudero,lluism,g.rigau}@lsi.upc.es

## Abstract

This paper describes the architecture and results of the TALP system presented at the SENSEVAL-2 exercise for the English lexical–sample task. This system is based on the LazyBoosting algorithm for Word Sense Disambiguation (Escudero et al., 2000), and incorporates some improvements and adaptations to this task. The evaluation reported here includes an analysis of the contribution of each component to the overall system performance.

## 1 System Description

The TALP system has been developed on the basis of LazyBoosting (Escudero et al., 2000), a boosting–based approach for Word Sense Disambiguation. In order to better fit the SENSEVAL-2 domain, some improvements have been made on the basic system, including: features that take into account domain information, an specific treatment of multiwords, and a hierarchical decomposition of the multiclass classification problem, similar to that of (Yarowsky, 2000). All these issues will be briefly described in the following sections.

### 1.1 LazyBoosting

The purpose of boosting–based algorithms is to find a highly accurate classification rule by combining many *weak classifiers* (or weak hypotheses), each of which may be only moderately accurate. The weak hypotheses are learned sequentially, one at a time, and, conceptually, at each iteration the weak hypothesis is biased to classify the examples which were most difficult to classify by the preceding weak hypotheses. The learned weak hypotheses are linearly combined into a single rule called the *combined hypothesis*.

The particular algorithm used in our system to perform the classification of senses is the generalized AdaBoost.MH with confidence–rated predictions (Schapire and Singer, 1999). This algorithm is able to deal straightforwardly with multiclass multi–label problems, and has been previously applied, with significant success, to a number of NLP disambiguation tasks, including, among others: Part–of–speech tagging and PP–attachment (Abney et al., 1999), text categorization (Schapire and Singer, 2000), and shallow parsing (Carreras and Màrquez, 2001). The weak hypotheses used in this work are *decision stumps*, which can be seen as extremely simple decision trees with one internal node testing the value of a single binary feature (e.g. "the word *dark* appears in the context of the word to be disambiguated?") and two leaves that give the prediction of the senses based on the feature value.

The "Lazy" Boosting, is a simple modification of the AdaBoost.MH algorithm, which consists of reducing the feature space that is explored when learning each weak classifier. More specifically, a small proportion of attributes are randomly selected and the best weak rule is selected only among them. This modification significantly increases the efficiency of the learning process with no loss in accuracy (Escudero et al., 2000).

### 1.2 Feature Space

Three kinds of information have been used to describe the examples and to train the classifiers. These features refer to local and topical contexts, and domain labels.

More particularly, let "$\ldots\ w_{-3}\ w_{-2}\ w_{-1}\ w\ w_{+1}\ w_{+2}\ w_{+3}\ldots$" be the context of consecutive words around the word $w$ to be disambiguated, and $p_{\pm i}$

($-3 \leq i \leq 3$) be the part–of–speech tag of word $w_{\pm i}$[1]. Feature patterns referring to local context are the following 13:

$$p_{-3}, p_{-2}, p_{-1}, p_{+1}, p_{+2}, p_{+3}, w_{-2}, w_{-1}, w_{+1}, w_{+2}, (w_{-2}, w_{-1}), (w_{-1}, w_{+1}), \text{and } (w_{+1}, w_{+2}),$$

where the last three correspond to collocations of two consecutive words.

The topical context is formed by $c_1, \ldots, c_m$, which stand for the unordered set of open class words appearing in a medium–size 21-word window centered around the target word.

The more innovative use of semantic domain information is detailed in the next section.

### 1.2.1 Domain Information

We have enriched the basic set of features by adding semantic information in the form of domain labels. These domain labels are computed during a preprocessing step using the 164 domain labels linked to the nominal part of WordNet 1.6 (Magnini and Cavaglia, 2000).

For each training example, a program gathers, from its context, all nouns and their synsets with the attached domain labels, and scores them according to a certain scoring function. The weights assigned by this function depend on the number of domain labels assigned to each noun and their relative frequencies in the whole WordNet. The result of this procedure is the set of domain labels that achieve a score higher than a certain experimentally set threshold, which are incorporated as regular features for describing the example.

## 1.3 Preprocessing and Hierarchical Decomposition

We began this exercise by selecting a representative sample, containing the most frequent words of the SENSEVAL-2 training data, and applying the LazyBoosting system straightforwardly on this sample. The results achieved after a 10–fold cross–validation procedure were very bad, mainly due to the fact that most of the words contain too many senses and too few examples per sense to induce reliable classifiers. With the aim of improving the performance of the learning algorithm, we have reduced the number of senses by performing a hierarchical decomposition of the multiclass problem, following the idea of (Yarowsky, 2000).

Two different simplifications have been carried out. Firstly, multiword training examples have been processed separately. During training, multiwords have been saved into a separate file. At test time, all examples found in this multiword file are automatically tagged as multiwords. As an example, the word *bar* appears in the training set with 22 labels. But only the 10 senses showed in the left table of figure 1 are single words. The remaining 12 are multiwords which are considered unambiguous (Yarowsky, 1993).

| Full senses | |
|---|---|
| **Senses** | **Exs.** |
| bar%1:06:04:: | 127 |
| bar%1:06:00:: | 29 |
| bar%1:06:05:: | 28 |
| bar%1:14:00:: | 17 |
| bar%1:10:00:: | 12 |
| bar%1:06:06:: | 11 |
| bar%1:04:00:: | 5 |
| bar%1:06:02:: | 4 |
| bar%1:23:00:: | 3 |
| bar%1:17:00:: | 1 |

| 1st level | |
|---|---|
| **Senses** | **Exs.** |
| bar%1:06 | 199 |
| bar%1:14 | 17 |
| bar%1:10 | 12 |

| 2nd level | |
|---|---|
| **Senses** | **Exs.** |
| 04:: | 127 |
| 00:: | 29 |
| 05:: | 28 |
| 06:: | 11 |

Figure 1: Sense treatment for word 'bar'

Secondly, we have reduced the sense granularity, by hierarchically decomposing the learning process in two steps. In the first level, the learning algorithm is trained to classify between the labels corresponding to the WordNet semantic files, and, additionally the semantic–file labels with less than 10 training examples are automatically discarded. If less than two senses remain, no training is performed and, simply, the *Most-frequent-sense Classifier* is applied.

As an example, for the word *'bar'*, in this first step the system is trained to classify between the labels of the top–right table of figure 1. Note that senses *bar%1:04*, *bar%1:23* and *bar%1:17* have been dropped out because there are not enough training examples.

In the second level, one classifier is trained for each of the resulting semantic–file labels of the first step in order to distinguish between their particular senses. Note that the same simplifying rules of the previous level are also applied. For instance, the bottom–right table of figure 1 shows the labels for *bar%1:06*, where *02::* has been rejected.

When classifying a new test example, the classifiers of the two levels are applied sequentially. That

[1] In this work, the English versions of MACO+ morphological analyzer and RELAX part–of–speech tagger have been used for tagging (Carmona et al., 1998).

is, the semantic–file classifier is applied first. Then, depending on the semantic–file label output by this classifier, the appropriate 2nd level classifier is selected. The resulting label assigned to the test example is formed by the concatenation of the outputs of both previous levels.

In the official competition, labels 'U' and 'P' have been completely ignored. Thus, the examples labelled with these classes have not been considered during the training, and no test examples have been tagged with them.

Despite the simplifying assumptions and the loss of information, we have observed that all these changes together significantly improved the accuracy on the training set. However, the components of the system were not tested separately due to the lack of time. Next section includes some evaluation about this issue.

## 2 Evaluation

The official results achieved by the TALP system are presented in table 1. The evaluation setting corresponding to these results contains all the modifications explained in the previous sections, including the hierarchical approach to all words.

| | **Accuracy** |
|---|---|
| **fine–grained** | 59.4% |
| **coarse–grained** | 67.1% |

Table 1: Official results

After the SENSEVAL-2 event, we added a very simple Named–entity Recognizer to the part–of–speech tagger that was not finished at the time of the event, but the system continues ignoring the 'U' label. We also have evaluated which parts of the system contributed most to the improvement in performance.

Table 2 shows the accuracy results of the four combinations resulting from using (or not) domain–label features and hierarchical decomposition. These results have been calculated over the test set of SENSEVAL-2.

On the one hand, it becomes clear that enriching the feature set with domain labels systematically improves the results in all cases, and that this difference is specially noticeable in the case of nouns (over 3 points of improvement). On the other hand, the use of the hierarchies is unexpectedly useless in all cases. Although it is productive in some particular words (3 nouns, 12 verbs and 5 adjectives) the overall performance is significantly lower. A fact that can explain this situation is that the first–level classifiers do not succeed on classifying semantic–file labels with high precision (the average accuracy of first–level classifiers is only slightly over 71%) and that this important error is dramatically propagated to the second–level, not allowing the greedy sequential application of classifiers. A possible explanation of this fact is the way semantic classes are defined in WordNet. Consider for instance work#1 (activity) and work#2 (production), they seem quite close but a system trying to differentiate among semantic files needs to distinguish among these two senses. On the other extreme, such a classifier should collapse house#2 (legislature) with house#4 (family), which are quite different. Of course, joining both situations makes a pretty hard task.

| | **nouns** | | | |
|---|---|---|---|---|
| | **without dom.** | | **with dom.** | |
| | fine | coarse | fine | coarse |
| **not hier.** | 64.25 | 72.35 | **67.90** | **75.60** |
| **hier.** | 63.00 | 71.10 | 64.31 | 71.49 |
| | **verbs** | | | |
| | **without dom.** | | **with dom.** | |
| | fine | coarse | fine | coarse |
| **not hier.** | 51.61 | 61.63 | **52.10** | **62.62** |
| **hier.** | 50.28 | 60.80 | 51.11 | 61.96 |
| | **adjectives** | | | |
| | **without dom.** | | **with dom.** | |
| | fine | coarse | fine | coarse |
| **not hier.** | 66.17 | 66.17 | **68.90** | **68.90** |
| **hier.** | 65.35 | 65.35 | 68.21 | 68.21 |

Table 2: Fine/coarse–grained evaluation for different settings and part–of–speech

Regarding multiword preprocessing (not included in table 2), we have seen that is slightly useful in all cases. It improves the non–hierarchical scheme with domain information by almost 1 point in accuracy. By part–of–speech, the improvement is about 1 point for nouns, 0.1 for verbs and about 2 points for adjectives.

In conclusion, the best results obtained by our system on this test set correspond to the application of multiword preprocessing and domain–labels for all words, but no hierarchical decomposition at all, achieving a fine–grained accuracy of 61.51% and a coarse–grained accuracy of 69.00%. We know that it is not fair to consider these results for comparison, since the system is tuned over the test set. Our aim is simply to fully inspect the TALP system to know which parts are useful for a real Word Sense Disambiguation system.

## 3 Work in progress

We think that the system presented in this paper still has a large room for improvement. Among all the research lines and developments that we are currently performing on the TALP system for WSD, we would like to mention the following:

- Tuning the preprocessing procedure with improved versions of the Named–entity Recognizer and Domain taggers.
- Studying in more detail the promising use of domain information in the feature set.
- Enriching the set of features with the most relevant features used by the SENSEVAL-2 systems, and using the Minipar[2] parser to obtain dependency and role information.
- Exploring more appropriate ways of making the hierarchical decomposition, not based on semantic files, and improve the sequential application of classifiers in order to reduce the cascade errors.
- Using unlabeled data to obtain larger sets of accurate training data, especially for those words/senses with few training examples.

## 4 Conclusions

This paper has presented the main characteristics and current performance of the TALP system within the framework of SENSEVAL-2 English lexical–sample task competition.

The system is mainly based on LazyBoosting (Escudero et al., 2000), which uses an improved version of the boosting algorithm AdaBoost.MH to perform the WSD classification problem.

We used a common set of features including local and topical context enriched with domain information. We obtained better performance separating multiword examples and also adding domain information.

Due to the small number of examples for training, we also tried to concentrate evidence reducing the fine-grained sense distinctions of WordNet. We perform a hierarchical procedure grouping those senses belonging to the same semantic file, preprocessing multiwords and ignoring 'U' label. After the competition, we have shown that the hierarchical decomposition fails to improve performance in this domain, while preprocessing of multiwords is quite useful. The improved system achieved a fine–grained accuracy of 61.51% and a coarse–grained accuracy of 69.00%.

## 5 Acknowledgements

We would like to thank Xavier Carreras, Lluís Padró, Victoria Arranz, and the anonymous referees for their helpful comments. This research has been partially funded by the European Commission (NAMIC project, IST-1999-12392) and the Spanish Research Department (HERMES project, TIC2000-0335-C03-02).

## References

S. Abney, R. E. Schapire and Y. Singer. 1999. Boosting Applied to Tagging and PP–attachment. In *Proceedings of EMNLP–VLC'99.*

J. Carmona, S. Cervell, L. Màrquez, M. A. Martí, L. Padró, R. Placer, H. Rodríguez, M. Taulé and J. Turmo. 1998. An Environment for Morphosyntactic Processing of Unrestricted Spanish Text. In *Proceedings of LREC'98.*

X. Carreras and L. Màrquez. 2001. Boosting Trees for Clause Splitting. In *Proceedings of CoNLL'01.*

G. Escudero, L. Màrquez and G. Rigau. 2000. Boosting Applied to Word Sense Disambiguation. In *Proceedings of ECML'00.*

B. Magnini and G. Cavaglia. 2000. Integrating Subject Field Codes into WordNet. In *Proceedings of LREC'00.*

R. E. Schapire and Y. Singer. 1999. Improved Boosting Algorithms Using Confidence-rated Predictions. *Machine Learning*, 37(3):297–336.

R. E. Schapire and Y. Singer. 2000. BoosTexter: A Boosting-based System for Text Categorization. *Machine Learning*, 29(3/4):135–168.

D. Yarowsky. 1993. One Sense per Collocation. In *Proceedings of the DARPA Workshop on Human Language Technology.*

D. Yarowsky. 2000. Hierarchical Decision Lists for Word Sense Disambiguation. *Computer and the Humanities*, 34:179–186.

[2] Available at http://www.cs.ualberta.ca/~ lindek .