The Basque Country University system: English and Basque tasks

Eneko Agirre
IXA NLP Group
Basque Country University
Donostia, Spain
eneko@si.ehu.es

David Martinez
IXA NLP Group
Basque Country University
Donostia, Spain
davidm@si.ehu.es

Abstract

Our group participated in the Basque and English lexical sample tasks in Senseval-3. A language-specific feature set was defined for Basque. Four different learning algorithms were applied, and also a method that combined their outputs. Before submission, the performance of the methods was tested for each task on the Senseval-3 training data using cross validation. Finally, two systems were submitted for each language: the best single algorithm and the best ensemble.

1 Introduction

Our group (BCU, Basque Country University), participated in the Basque and English lexical sample tasks in Senseval-3. We applied 4 different learning algorithms (Decision Lists, Naive Bayes, Vector Space Model, and Support Vector Machines), and also a method that combined their outputs. These algorithms were previously tested and tuned on the Senseval-2 data for English. Before submission, the performance of the methods was tested for each task on the Senseval-3 training data using 10 fold cross validation. Finally, two systems were submitted for each language, the best single algorithm and the best ensemble in cross-validation.

The main difference between the Basque and English systems was the feature set. A rich set of features was used for English, including syntactic dependencies and domain information, extracted with different tools, and also from external resources like WordNet Domains (Magnini and Cavagliá, 2000). The features for Basque were different, as Basque is an agglutinative language, and syntactic information is given by inflectional suffixes. We tried to represent this information in local features, relying on the analysis of a deep morphological analyzer developed in our group (Aduriz et al., 2000).

In order to improve the performance of the algorithms, different smoothing techniques were tested on the English Senseval-2 lexical sample data (Agirre and Martinez, 2004), and applied to Senseval-3. These methods helped to obtain better estimations for the features, and to avoid the problem of 0 counts Decision Lists and Naive Bayes.

This paper is organized as follows. The learning algorithms are first introduced in Section 2, and Section 3 describes the features applied to each task. In Section 4, we present the experiments performed on training data before submission; this section also covers the final configuration of each algorithm, and the performance obtained on training data. Finally, the official results in Senseval-3 are presented and discussed in Section 5.

2 Learning Algorithms

The algorithms presented in this section rely on features extracted from the context of the target word to make their decisions.

The **Decision List (DL)** algorithm is described in (Yarowsky, 1995b). In this algorithm the sense with the highest weighted feature is selected, as shown below. We can avoid undetermined values by discarding features that have a 0 probability in the divisor. More sophisticated smoothing techniques have also been tried (cf. Section 4).

$$\arg \max_k w(s_k, f_i) = \log(\frac{Pr(s_k|f_i)}{\sum_{j \neq k} Pr(s_j|f_i)})$$

The **Naive Bayes (NB)** algorithm is based on the conditional probability of each sense given the features in the context. It also requires smoothing.

$$\arg \max_k P(s_k) \prod_{i=1}^{m} P(f_i|s_k)$$

For the **Vector Space Model (V)** algorithm, we represent each occurrence context as a vector, where each feature will have a 1 or 0

value to indicate the occurrence/absence of the feature. For each sense in training, one centroid vector is obtained. These centroids are compared with the vectors that represent testing examples, by means of the cosine similarity function. The closest centroid is used to assign its sense to the testing example. No smoothing is required to apply this algorithm, but it is possible to use smoothed values.

Regarding **Support Vector Machines (SVM)** we utilized SVM-Light (Joachims, 1999), a public distribution of SVM. Linear kernels were applied, and the soft margin (C) was estimated per each word (cf. Section 4).

3 Features

3.1 Features for English

We relied on an extensive set of features of different types, obtained by means of different tools and resources. The features used can be grouped in four groups:

Local collocations: bigrams and trigrams formed with the words around the target. These features are constituted with lemmas, word-forms, or PoS tags[1]. 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[2]. The following relations were used: object, subject, noun-modifier, preposition, and sibling.

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

Domain features: The WordNet Domains resource was used to identify the most relevant domains in the context. Following the relevance formula presented in (Magnini and Cavagliá, 2000), we defined 2 feature types: (1) the most relevant domain, and (2) a list of domains above a predefined threshold[3]. Other experiments using domains from SUMO, the EuroWordNet top-ontology, and WordNet's Semantic Fields were performed, but these features were discarded from the final set.

3.2 Features for Basque

Basque is an agglutinative language, and syntactic information is given by inflectional suffixes. The morphological analysis of the text is a necessary previous step in order to select informative features. The data provided by the task organization includes information about the lemma, declension case, and PoS for the participating systems. Our group used directly the output of the parser (Aduriz et al., 2000), which includes some additional features: number, determiner mark, ambiguous analyses and elliptic words. For a few examples, the morphological analysis was not available, due to parsing errors.

In Basque, the determiner, the number and the declension case are appended to the last element of the phrase. When defining our feature set for Basque, we tried to introduce the same knowledge that is represented by features that work well for English. We will describe our feature set with an example: for the phrase "elizaren arduradunei" (which means "to the directors of the church") we get the following analysis from our analyzer:

eliza	|-ren	|**arduradun**	|-ei
church	|of the	|**director**	|to the +pl.

The order of the words is the inverse in English. We extract the following information for each word:

```
elizaren:
  Lemma:  eliza (church)
  PoS: noun
  Declension Case:  genitive (of)
  Number:  singular
  Determiner mark:  yes
arduradunei:
  Lemma:  arduradun (director)
  PoS: noun
  Declension Case:  dative (to)
  Number:  plural
  Determiner mark:  yes
```

We will assume that eliza (church) is the target word. Words and lemmas are shown in lowercase and the other information in uppercase. As local features we defined different types of unigrams, bigrams, trigrams and a window of ±4 words. The unigrams were constructed combining word forms, lemmas, case, number, and determiner mark. We defined 4

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

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

[3]The software to obtain the relevant domains was kindly provided by Gerard Escudero's group, from Universitat Politecnica de Catalunya

kinds of unigrams:

```
Uni_wf0 elizaren
Uni_wf1 eliza SING+DET
Uni_wf2 eliza GENITIVE
Uni_wf3 eliza SING+DET GENITIVE
```

As for English, we defined bigrams based on word forms, lemmas and parts-of-speech. But in order to simulate the bigrams and trigrams used for English, we defined different kinds of features. For word forms, we distinguished two cases: using the text string (Big_wf0), or using the tags from the analysis (Big_wf1). The word form bigrams for the example are shown below. In the case of the feature type "Big_wf1", the information is split in three features:

```
Big_wf0 elizaren arduradunei
Big_wf1 eliza GENITIVE
Big_wf1 GENITIVE arduradun_PLUR+DET
Big_wf1 arduradun_PLUR+DET DATIVE
```

Similarly, depending on the use of the declension case, we defined three kinds of bigrams based on lemmas:

```
Big_lem0 eliza arduradun
Big_lem1 eliza GENITIVE
Big_lem1 GENITIVE arduradun
Big_lem1 arduradun DATIVE
Big_lem2 eliza_GENITIVE
Big_lem2 arduradun_DATIVE
```

The bigrams constructed using Part-of-speech are illustrated below. We included the declension case as if it was another PoS:

```
Big_pos_-1 NOUN GENITIVE
Big_pos_-1 GENITIVE NOUN
Big_pos_-1 NOUN DATIVE
```

Trigrams are built similarly, by combining the information from three consecutive words. We also used as local features all the content words in a window of ± 4 words around the target. Finally, as global features we took all the content lemmas appearing in the context, which was constituted by the target sentence and the two previous and posterior sentences.

One difficult case to model in Basque is the ellipsis. For example, the word "elizakoa" means "the one from the church". We were able to extract this information from our analyzer and we represented it in the features, using a special symbol in place of the elliptic word.

4 Experiments on training data

The algorithms that we applied were first tested on the Senseval-2 lexical sample task for English. The best versions were then evaluated by 10 fold cross-validation on the Senseval-3 data, both for Basque and English. We also used the training data in cross-validation to tune the parameters, such as the smoothed frequencies, or the soft margin for SVM. In this section we will describe first the parameters of each method (including the smoothing procedure), and then the cross-validation results on the Senseval-3 training data.

4.1 Methods and Parameters

DL: On Senseval-2 data, we observed that DL improved significantly its performance with a smoothing technique based on (Yarowsky, 1995a). For our implementation, the smoothed probabilities were obtained by grouping the observations by raw frequencies and feature types. As this method seems sensitive to the feature types and the amount of examples, we tested 3 DL versions: DL_smooth (using smoothed probabilities), DL_fixed (replacing 0 counts with 0.1), and DL_discard (discarding features appearing with only one sense).

NB: We applied a simple smoothing method presented in (Ng, 1997), where zero counts are replaced by the probability of the given sense divided by the number of examples.

V: The same smoothing method used for NB was applied for vectors. For Basque, two versions were tested: as the Basque parser can return ambiguous analyses, partial weights are assigned to the features in the context, and we can chose to use these partial weights (p), or assign the full weight to all features (f).

SVM: No smoothing was applied. We estimated the soft margin using a greedy process in cross-validation on the training data per each word.

Combination: Single voting was used, where each system voted for its best ranked sense, and the most voted sense was chosen. More sophisticate schemes like ranked voting, were tried on Senseval-2 data, but the results did not improve. We tested combinations of the 4 algorithms, leaving one out, and the two best. The best results were obtained combining 3 methods (leave one out).

Method	Recall
vector	**73,9**
SVM	73,5
DL_smooth	69,4
NB	69,4
DL_fixed	65,6
DL_discard	65,4
MFS	57,1

Table 1: Single systems (English) in cross-validation, sorted by recall.

Combination	Recall
SVM-vector-DL_smooth-NB	73,2
SVM-vector-DL_fixed-NB	72,7
SVM-vector-DL_smooth	**74,0**
SVM-vector-DL_fixed	73,8
SVM-vector-NB	73,6
SVM-DL_smooth-NB	72,4
SVM-DL_fixed-NB	71,3
SVM-vector	73,1

Table 2: Combined systems (English) in cross-validation, best recall in bold.

Method	Recall
SVM	**71,1**
NB	68,5
vector(f)	66,8
DL_smooth	65,9
DL_fixed	65,2
vector(p)	65,0
DL_discard	60,7
MFS	53,0

Table 3: Single systems (Basque) in cross-validation, sorted by recall.

Combination	Recall
SVM-vector-DL_smooth-NB	70,6
SVM-vector-DL_fixed-NB	**71,1**
SVM-vector-DL_smooth	70,6
SVM-vector-DL_fixed	70,8
SVM-vector-NB	**71,1**
SVM-DL_smooth-NB	70,2
SVM-DL_fixed-NB	70,5
SVM-vector	69,0
SVM-NB	69,8

Table 4: Combined systems (Basque) in cross-validation, best recall in bold. Only vector(f) was used for combination.

4.2 Results on English Training Data

The results using cross-validation on the Senseval-3 data are shown in Table 1 for single systems, and in Table 2 for combined methods. All the algorithms have full-coverage (for English and Basque), therefore the recall and the precision are the same. The most frequent sense (MFS) baseline is also provided, and it is easily beaten by all the algorithms.

We have to note that these figures are consistent with the performance we observed in the Senseval-2 data, where the vector method is the best performing single system, and the best combination is SVM-vector-DL_smooth. There is a small gain when combining 3 systems, which we expected would be higher. We submitted the best single system, and the best combination for this task.

4.3 Results on Basque Training Data

The performance on the Senseval-3 Basque training data is given in Table 1 for single systems, and in Table 2 for combined methods. In this case, the vector method, and DL_smooth obtain lower performance in relation to other methods. This can be due to the type of features used, which have not been tested as extensively as for English. In fact, it could happen that some features contribute mostly noise. Also, the domain tag of the examples, which could provide useful information, was not used.

There is no improvement when combining different systems, and the result of the combination of 4 systems is unusually high in relation to the English experiments. We also submitted two systems for this task: the best single method in cross-validation (SVM), and the best 3-method combination (SVM-vector-NB).

5 Results and Conclusions

Table 5 shows the performance obtained by our systems and the winning system in the Senseval-3 evaluation. We can see that we are very close to the best algorithms in both languages.

The recall of our systems is 1.2%-1.9% lower than cross-validation for every system and task, which is not surprising when we change the setting. The combination of methods is useful for English, where we improve the recall in 0.3%, reaching 72.3%. The difference is statistically significant according to McNemar's test.

However, the combination of methods does not improve the results in the the Basque task, where the SVM method alone provides better

Task	Code	Method	Rec.
Eng.	**Senseval-3 Best**	**?**	**72,9**
Eng.	BCU_comb	SVM-vector-DL_smooth	72,3
Eng.	BCU-english	vector	72,0
Basq.	**Senseval-3 Best**	**?**	**70,4**
Basq.	BCU-basque	SVM	69,9
Basq.	BCU-Basque_comb	SVM-vector-NB	69,5

Table 5: Official results for the English and Basque lexical tasks (recall).

results (69.9% recall). In this case the difference is not significant applying McNemar's test.

Our disambiguation procedure shows a similar behavior on the Senseval-2 and Senseval-3 data for English (both in cross-validation and in the testing part), where the ensemble works best, followed by the vector model. This did not apply to the Basque dataset, where some algorithms seem to perform below the expectations. For future work, we plan to study better the Basque feature set and include new features, such as domain tags.

Overall, the ensemble of algorithms provides a more robust system for WSD, and is able to achieve state-of-the-art performance.

6 Acknowledgements

We wish to thank both David Yarowsky's group, from Johns Hopkins University, and Gerard Escudero's group, from Universitat Politecnica de Catalunya, for providing us software for the acquisition of features. This research has been partially funded by the European Commission (MEANING IST-2001-34460).

References

I. Aduriz, E. Agirre, I. Aldezabal, I. Alegria, X. Arregi, J. Arriola, X. Artola, K. Gojenola, A. Maritxalar, K. Sarasola, and M. Urkia. 2000. A word-grammar based morphological analyzer for agglutinative languages. In *Proceedings of the International Conference on Computational Linguistics COLING*, Saarbrucken, Germany.

Eneko Agirre and David Martinez. 2004. Smoothing and word sense disambiguation. *(submitted)*.

T. Joachims. 1999. Making large–scale SVM learning practical. In *Advances in Kernel Methods — Support Vector Learning*, pages 169–184, Cambridge, MA. MIT Press.

Bernardo Magnini and Gabriela Cavagliá. 2000. Integrating subject field codes into WordNet. In *Proceedings of the Second International LREC Conference*, Athens, Greece.

Hwee Tou Ng. 1997. Exemplar-based word sense disambiguation: Some recent improvements. In *Proceedings of the Second EMNLP Conference*. ACL, Somerset, New Jersey.

Grace Ngai and Radu Florian. 2001. Transformation-based learning in the fast lane. *Proceedings of the Second Conference of the NAACL, Pittsburgh, PA, USA*.

Ted Pedersen. 2001. A decision tree of bigrams is an accurate predictor of word sense. *Proceedings of the Second Meeting of the NAACL, Pittsburgh, PA.*

David Yarowsky. 1995a. Three machine learning algorithms for lexical ambiguity resolution. In *PhD thesis, Department of Computer and Information Sciences, University of Pennsylvania*.

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