Design and Development of a Named Entity Recognizer for an Agglutinative Language

Iñaki Alegria, Olatz Arregi, Irene Balza, Nerea Ezeiza, Izaskun Fernandez, Ruben Urizar
IXA taldea. Euskal Herriko Unibertsitatea.
i.alegria@si.ehu.es

Abstract

This paper presents the conclusions reached from the development of a system for Named Entity recognition in written Basque. The system was designed in four steps: first, the development of a recognizer based on linguistic information represented on finite-state-transducers; second, the generation of semi-automatically annotated corpora from the result of these transducers; third, the achievement of the best possible recognizer by training different ML techniques on these corpora; and finally, the combination of the different recognizers obtained. Being Basque an agglutinative language, a linguistic preprocess previous to these steps was required.

1 Introduction and Related Work

Named Entity Recognition and Classification (NERC) constitutes a very important task in Natural Language Processing (NLP) and more specifically in tasks related to Information Extraction.

As defined in the *Message Understanding Conference* (MUC) (Chinchor, 1998), NE recognition consists in identifying and categorizing entity names (person, organization and location), temporal expressions (dates and times), and some types of numerical expressions (percentages, monetary values and so on). Sometimes two different subtasks are distinguished: Named Entity Recognition (NER) which delimits the boundaries of the entities, and Named Entity Classification (NEC) which assigns the type of entity.

According to (McDonald, 1996), there are two kinds of data that should be taken into account in order to identify and classify the possible NEs: internal evidence and external evidence. The former is provided by the expression itself and the latter by the context in which it occurs.

Among the different techniques used to process these data, we find some systems based on statistical methods, some based on strictly linguistic methods which make use of grammar rules (Magnini et al., 2002), and finally the ones that combine rules and statistics (Mikheev et al., 1998). Before applying these techniques, some previous work involving a more or less deep analysis of the written text is generally required. In the simplest cases, only tokenization is applied, but in other cases, also a morphological analysis, disambiguation, and the attachment of semantic features must be carried out.

The machine-learning (ML) paradigm has been very successful in this task and recently language-independent shared tasks have been celebrated with different systems using the same corpora (Tjong, 2002) (Tjong & De Meulder, 2003). Based on these workshops we can observe that Maximum Entropy, AdaBoost or the combination of several methods offer good results. The features used are quite good defined too, more concretely a window with the previous and next words, including information about spelling (mainly capital letters), part-of-speech (PoS), affixes, chunck tags and appearance in gazetteers. Anyway, results in CoNLL-2003 "*do not reveal a single feature that would be ideal for NER*" (Tjong & De Meulder, 2003).

The paper is structured as follows: Section 2 presents the aims and design of the project. Section 3 deals with the first step, a "linguistic" recognizer based on a grammar. In section 4 we describe the different experiments in order to obtain the best possible tool using ML based methods. Finally, section 5 and 6 present the evaluation and conclusions respectively.

2 Aims and Design

The need of a Named Entity recognizer was observed in two projects of IXA Group (ixa.si.ehu.es) and, being a basic tool for future projects in Language Engineering, we decided to build a system for identifying and classifying NEs. This paper mainly deals with the task of identification, since this is the hardest work.

The main goal of our system is to capture expressions referring to Persons, Locations and Organizations. Numerical and temporal expressions are already captured by the lemmatizer/tagger (Ezeiza et al., 1998) used in the preprocess, which performs the tokenization, the morphosyntactic analysis and the POS disambiguation of the text. Since Basque is an agglutinative language the linguistic preprocess of the texts is more necessary than in languages with a simpler morphology.

When we faced the task we took into account two possibilities:

- to tag directly by hand a corpus and make the tool using ML techniques.
- to build a linguistic tool which will help us tag a corpus.

Although ML techniques offer robustness and good results, we finally decided to build a "linguistic" recognizer, bearing in mind that:

- it can offer good results too.
- it can be combined with ML based methods and they can be complementary.
- the linguistic features defined for this method can be useful for future steps.
- a simple method can be developed in short time if you know the tool.
- the necessary linguistic preprocess has already been prepared.
- it permits a semiautomatic way to annotate the corpora for evaluation (and for learning with the ML methods), making this task simpler and faster.

The project was designed in four steps. In a first step, a recognizer based on linguistic information represented in finite state transducers was developed. Secondly, we generated semi-automatically annotated corpora (afterwards hand-reviewed). Thirdly, we trained different ML techniques on these corpora in order to obtain the best possible recognizer. And finally, the recognizers obtained were combined so as to improve the results.

3 The Linguistic Recognizer

The "linguistic" version, named *Eihera*, result of the first step, consists of a grammar whose rules are based on the morphological information of the text provided by the lemmatizer/tagger. This lemmatizer/tagger is described in depth in (Alegria et al., 2003).

The tool used to develop the grammar for the identification of entities is XFST (*Xerox Finite State Transducer*) (Beesley&Karttunen, 2003). XFST allows us to define both the structure of entity names and the rules for their identification.

3.1 Main elements of the grammar

Among the main elements of our grammar we find entity names and trigger words. Although the latter are not relevant for the identification of the entities, they are helpful for their classification.

The main feature of the whole entity is the use of capital letters, but there are other features that should be taken into account, for instance the PoS and subcategory of the elements and their inflection.

The main PoS/subcategories we must distinguish among the entity elements are the following: IZE (common noun), IZB (proper noun), LIB (location/organization proper noun), ADJ (adjective), SIG (acronym) and BST (particle[1]). Except for the case of some BST, the rest of the elements in the entity must be written in capitals. For the identification of entities we make a distinction between non-case elements, genitive, and others because the first two can indicate the continuation of the entity.

3.2 Main patterns

In the grammar, two patterns of named entities are distinguished: entities containing a single element (*Europan* LOCATION) and entities composed of

[1] BST stands for particles that occur in some entities borrowed from Spanish, such as *Santiago **de** Compostela*

more than one element (*Europako Banku Zentralean* ORGANIZATION).

In the first case the element PoS assigned by the lemmatizer/tagger must be SIG (acronym), IZB (proper noun) or LIB (location/organization proper noun). When the element is declined, it can bear any case.

Examples of one-element entities are *EHUn (SIG+inesive)* or *Bilbora (LIB+adlative)*

The entities with more than one element have a more complex pattern. First, the last element of the entity and the rest are different, since the latter has a more restricted declension (only genitive), while the former can appear in any case. In contrast with one-element entities, the elements contained in this second type, can belong to different parts of speech: IZE (common noun), IZB (proper noun), LIB (location/organization proper noun), ADJ (adjective), SIG (acronym) or BST (particle).

Examples of the second pattern are:

- *Europako* (LIB+GENITIVE) *Banku* (IZE) *Zentralean* (ADJ+INESIVE)
- *Alex* (IZB) *de* (BST) *la* (BST) *Iglesiak* (IZB+ERGATIVE)

In any case, the grammar captures the longest sequence of possible entity elements that matches with any of the patterns defined above.

3.3 Results

Eihera´s grammar works quite well. In a test corpus with 935 entities it identifies 951 entities from which 805 are correct.

	Prec.	Recall	F-score
Eihera	84.65	86.10	85.37

Table 1: Results for Eihera in NER

We have examined the reason for the errors revising 100 incorrect NEs. As shown in Table 2 half of the identification errors are due to external reasons and so they would not occur in accurately written texts. Most of the remaining ones could be corrected if the tagger was improved.

Reason	**Percentage**
Errors in capital letters	35 %
Bad analyses in preprocess	29 %
Errors in the input format	22 %
Nested NEs	8 %
Others	6 %

Table 2: Source of errors in identification

Anyway, as it was planned in the design, the system becomes more robust and better results can be obtained if ML techniques are used.

4 Experiments with ML Methods

The corpus obtained applying *Eihera* on articles from *Euskaldunon Egunkaria* were hand-reviewed and then BIO tagged[2]. In order to use this corpus for ML techniques, we divided it in the following way: 46227 words and 3817 entities for the training corpus and 15960 words and 935 entities for testing[3].

We considered *Weka* (Witten & Eibe, 1999) a good choice for ML experiments, because, apart from being free software with GPL license, it is also a very flexible tool, allowing us to experiment with different methods.

In order to compare the results to state-of-the-art in the task, we trained *Abionet* (Carreras et al., 2003) with our corpus. This method was one of the best in both CoNLL-2002 and CoNLL-2003 for NERC in several languages.

Our work with those tools has been focused on features selection and tuning.

A language independent system[4], based on multiple orthographic tries which are combined with a Hidden Markov model framework (Whitelaw&Patrick, 2003) has been also used. Although this tool achieves poorer results than the methods mentioned before, we thought that a different approach could help improving the results when combining methods is carried out.

Once we have learned from the training corpus and tagged the test corpus, the evaluation can be carried out in different measures:

- Average precision, recall and F-score for all the BIO categories
- Precision, recall and F-score for B and I categories, since they are the aim of the system
- Precision, recall and F-score of the identified entities using the same system as in CoNLL.

The last measure is the most useful in order to compare different methods. So this will be the one employed in the evaluation tables. Time is another

[2] B-begin of entity, I-intermediate, O-out of entity

[3] The same test-corpora is used in all experiments.

[4] We will use *Sydney* in the tables

important factor for evaluating ML techniques, so figures about the learning time are given too[5].

4.1 Features

Based on the related work and the relevant information in the grammar, different features were proposed and tested with the ML learners.

The key-features we have obtained are the following (similar results were obtained with all the methods): word, lemma, PoS, declension case, capitalized word, capitalized lemma, word in capitals, and lemma in capitals. Word and lemma were represented as a number, as required by *Weka*.

Other features, as the explicit indication of punctuation signs, were excluded because they did not improve the results.

A [–3,+3] window was applied so 56 features per word were obtained (8 features in 7 words).

Some systems in CoNLL-2003 use external information in order to extract capitalization features, but we think that the information of the capitalized lemmas obtained from the preprocess is equivalent.

4.2 Methods

We decided to test some available ML methods: Naive Bayes (NB), C4.5 and Support Vector Machines (SVM)[6] from *Weka*, and AdaBoost from *Abionet* (Carreras & Márquez, 2003).

Although the methods from *Weka* are not the ones which provide the best results for the identification task, we wanted to test them with a double objective: to compare our results with the ones obtained by *Abionet*, and to have different recognizers in order to combine them.

The University of Sydney evaluated their language independent system for our corpus. The results of the different system are shown in Table 3.

Some conclusion can be extracted:

- As expected Naive Bayes gives a poor result.
- SVM does not overcome the results of C4.5 and it needs a very long time for training.
- The results obtained by the best method in *Weka* are 3% down from the results of *Abionet* which represent the state-of-art.
- The performance of the linguistic system (*Eihera*) is poorer but close to the best methods in *Weka*.

	Time(s.)[7]	Prec.	Recall	F-score
Sydney		74.74	77.54	76.12
Eihera		84.65	86.10	85.37
Naive Bayes	14	46.08	54.76	50.05
C4.5	110	86.74	82.57	84.60
SVM	35500	85.02	81.93	83.44
Abionet-BIO[8]		89.22	85.88	87.52
Abionet		89.81	85.78	87.75

Table 3: Results of different methods

4.3 Selection of features

The selection of a set of features can improve the results. The use of words and/or lemmas was a challenge that we had worked in documents' classification and we thought it interesting to test on NEs identification. Lemmas concentrate information of several words (specially in agglutinative languages), but they lose other kind of information (declension, time, ...).

	words+ lemmas	words	lemmas
Naive Bayes	50.05	50.02	62.36
C4.5	84.60	84.59	84.94
SVM	83.44		83.38

Table 4: Results (F-score) with words and/or lemmas

In the evaluated version we took advantage from both but none them is a good candidate for testing in feature selection. So, we tested the results with only words, only lemmas and compare them to the previous with words and lemmas. The results for F-score are in Table 4.

Looking to the table we can conclude that the use of only lemmas improves the results fairly in Naive Bayes. The improvement is only in precision, which raises from 46.08 to 72.6 using Bayes, but only from 50 to 62.36 in F-score. Using

[5] The time of test is similar in all the methods

[6] SVM is a binary classifier but in Weka is extended to handle multi-class using pair-wise classification

[7] The time given in all tables is in seconds, and only Weka's training times have been estimated

[8] Abionet-BIO represents the result obtained in *Abionet* without using BIO-tags of previous words. This measure is taken because this features can not be used in *Weka*.

C4.5 the gain is weaker (only in precision too) and in the case of SVM a slight loss happens.

4.4 Selection of instances

Another technique used to improve the results is to try to set a better sampling, discarding errors and avoiding overfitting. As a consequence of this selection, the learning process can be faster too[9].

We thought that the O category may unbalance the corpus. Researches in the MIT (Rennie et al., 2003) show the negative effect of classes with very different number of instances using Naive Bayes. In the NER task this effect can be even more negative because, although the system classifies BIO categories, the evaluation is only done over well classified entities. In fact the I category shows less performance than the others.

That is why we decided to remove some examples of words outside the entities and test the results.

A simple heuristic can discard, with high confidence, words that don't belong to entities. In our case, the elimination of all words with non-capital letters, except for nouns, conjunctions and BST tagged words rules out almost half of the instances (the number of instances falls from 46.227 words to 22.264 in the training corpus), while only 0.4% of the entities are left out. The same percentage disappears in the test corpus, so that the system will be, beforehand, unable to identify four entities.

There are two possible ways of discarding words:

- applying the window so as to obtain the examples, after removing the words.
- removing the examples after applying the window.

	F-score (whole)	F-score[10] (short)	Time(s) (whole)	Time(s) (short)
Naive Bayes	50.05	62.20	14	7
C4.5	84.60	84.30	110	61
SVM	83.44	84.45	35500	18900

Table 5: Effect of reduction of examples

The second strategy provides better results (about 3% more precision and recall) since richer information about the context of the words remains.

The results about F-score and time are shown in Table 5 when words and lemmas are used and in Table 6 when only lemmas are used.

Further conclusions can be extracted from these figures:

- The reduction of the training-set offers good results with Naive Bayes and SVM. In the last case, the results are now similar to those obtained using C4.5. This behavior might be due to the small of the trainig corpus[11].
- The time of training is drastically reduced, and, because time is a limiting factor for experiments, this can be considered as an interesting feature for SVM.

Still these results are 3% poorer than the target ones.

	F-score (whole)	F-score (short)	Time(s) (whole)	Time(s) (short)
Naive Bayes	62.36	69.68	12	6
C4.5	84.94	84.74	75	38
SVM	83.38	84.36	29900	16350

Table 6: Effect of reduction of examples with only lemmas

5 Combining methods

Combining methods is another strategy for improving the results. We had the option of combining the same methods with different features or training-sets and different kinds of methods.

	Prec.	Recall	F-score
1-Sydney	74.74	77.54	76.12
2-C4.5	86.74	82.57	84.60
3-C4.5-lemma	85.73	84.17	84.94
4-Eihera	84.65	86.10	85.37
5-Abionet	89.81	85.78	**87.75**

Table 7: Results of each method

The last choice is the most attractive because we already have three different kinds of methods that offer quite good results:

[9] this is very important in methods like SVM since they are very sensitive to the number of examples

[10] The results are slightly better (about the 0,2%) because the number of correct entities are 935 but after the selection only 931 appear in test corpus.

[11] An anonymous reviewer pointed us.

- Language independent method (Sydney)
- Linguistic method (*Eihera*)
- ML based methods (C4.5 and SVM in *Weka, and* AdaBoost *in Abionet*).

The results achieved with each method are shown in table 7.

Different combinations of methods using simple voting were tested. The most important results are plotted in Table 8.

	Prec.	Recall	F-score
Sydney, C4.5, C4.5-Lem	86.59	84.28	85.42
C4.5, C4.5-Lem, Eihera	86.65	84.71	85.67
Sydney, C4.5-Lem, Eihera	88.71	**87.38**	88.04
Sydney, C4.5, Eihera	89.49	**87.38**	**88.42**
Abionet, Sydney, C4.5	**90.72**	85.78	88.18
Abionet, Sydney, Eihera	90.08	**87.38**	88.71
Abionet, C4.5, Eihera	90.47	87.27	**88.84**

Table 8: Results of combining methods using voting

These results confirm the hypothesis that results can be improved when combining different kinds of recognizers. Combining the same kind of recognizers we got a very slight improvement, but combining the three kinds of methods (Sydney, *Eihera* and *Weka*) more improvement and better results than the state-of-art single method (*Abionet*) are obtained (88.42 vs. 87.75).

When *Abionet* is included in the combination, the best results are obtained, as it was expected, bringing together the three methods which produced the best results separately. However, the best recall (87.38) and precision (90.72) are obtained with *Sydney,* although this gets further worse results when performed individually.

6 Conclusions and Future work

We have presented the methodology and the results in a NE recognizer for Basque. The most important conclusions from this work are the following:

- The development of a linguistic tool is a useful task, because it allows simplifying the annotation, learning about features and getting a good tool to be combined with ML based methods.
- Linguistic features as lemmas are very useful, at least in agglutinative languages.
- Examples' selection can be useful in Named Entities Extraction for both speeding the training and improving results.
- The combination of methods can improve the performance of the system, specially if different kind of classificators are combined

At the time of writing this paper the evaluation for the NEC task is being carried out. In a near future we want to test the results with the maximum entropy model (ME) using a free-software (http://maxent.sourceforge.net/), like *Weka*.

7 Acknowledgements

The research of Iñaki Alegria was carried out during his visit to the University of Sydney supported by a grant from the Basque Government. We would like to thank Jon Patrick, Casey Whitelaw and the *ScamSeek* group for their support and helpful discussions, to Xavier Carreras and Lluis Màrquez for their help with *Abionet* and to the three IJCNLP04 anonymous referees for their constructive feedback.

This research has been partially funded by the Spanish Research Department (HERMES TIC2000-0335-CO3-02), the Basque Government (Etortek-Hizking, Saiotek-Ihardetsi) and the University of the Basque Country (UE02/B11).

8 References

Alegria I., Balza I., Ezeiza N., Fernandez I., Urizar R., 2003: Named Entity Recognition and Classification for texts in Basque. *Proc. of II Jornadas de Tratamiento y Recuperación de Información, JOTRI*, Madrid. 2003.

Beesley K.R., Karttunen L., 2003: *Finite State Morphology*. CSLI.

Carreras X., Màrquez L., Padró L., 2003: A Simple Named Entity Extractor Using AdaBoost. CoNLL 2003 Shared Task Contribution. *Proceedings of CoNLL-2003*

Chinchor N., 1998: Overview of MUC-7. In *Proceedings of the 7th Message Understanding Conference (MUC-7).*

Ezeiza N., Aduriz I., Alegria I., Arriola J.M., Urizar R., 1998: Combining Stochastic and Rule-Based Methods for Disambiguation in Agglutinative Languages. *COLING-ACL'98*, Montreal (Canada).

Magnini B., Negri M., Prevete R., Tanev H. A, 2002: WordNet Approach to Names Entity Recognition. *Proceeding of the Workshop SemaNet'2002: Binding and Using Semantic Networks.*

McDonald D., 1996: Internal and external evidence in the Identification and Semantic Categorization of Proper Names. *Corpus Processing for Lexical Acquisition (Boguraev and Pustejovsky, eds.).* The MIT Press, Massachusetts.

Mikheev A., Grover C., Moens M., 1998: Description f the LTG system used for MUC-7. *Proceeding of Message Understanding Conference (MUC-7).*

Rennie J.D.M., Shih L., Teevan J., Karger D.R., 2003 Tackling the Poor Assumptions of Naive Bayes Text Classifiers. *Proceedings of the 20th Conference on Machine Learning*.

Tjong Kim Sang E, 2002: Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. *Proceedings of CoNLL-2002.*

Tjong Kim Sang E and De Meulder F., 2003: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. *Proceedings of CoNLL-2003.*

Whitelaw C. and Patrick J., 2003: Named Entity Recognition Using a Character-based Probabilistic Approach. *Proceedings of CoNLL-2003.*

Witten I.H. and Eibe F., 1999: *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.