Procesamiento del Lenguaje Natural, vol. 1, num. 3 (2002), pp.1-15 recibido 1-2-2002; aceptado 20-2-2002

A Proposal for Wide–Coverage Spanish Named Entity Recognition*

M. Arévalo[1], X. Carreras[2], L. Màrquez[2], M.A. Martí[1], L. Padró[2] & M.J. Simón[1]

[1] Centre de Llenguatge i Computació (CLiC)
Universitat de Barcelona
Gran Via de les Corts Catalanes, 585, E-08007, Barcelona
{montse,amarti,simon}@lingua.fil.ub.es

[2] Centre de Recerca TALP
Departament LSI, Universitat Politècnica de Catalunya
Jordi Girona, 1–3, E-08034, Barcelona
{carreras,lluism,padro}@lsi.upc.es

Resumen: Este trabajo presenta una propuesta para el reconocimiento de amplia cobertura de entidades con nombre en castellano. En primer lugar, se propone una descripción lingüística de la tipología de las entidades con nombre. Seguidamente, se describe una arquitectura de procesos secuenciales para abordar el reconocimiento y clasificación de entidades fuertes y débiles. Las primeras se tratan con técnicas de aprendizaje automático (AdaBoost) y atributos simples sobre corpus no etiquetados, complementados con fuentes de información externas (una lista de palabras disparadoras y un gazetteer). Las segundas se abordan mediante una gramática incontextual para el reconocimiento de patrones sintácticos. Se presenta una evaluación en profundidad de la primera tarea sobre corpus reales para validar la adecuación del método. También se expone una valoración cualitativa de la gramática incontextual, con buenos resultados sobre un pequeño corpus de prueba.
Palabras clave: Reconocimiento y clasificación de entidades con nombre, Aprendizaje Automático para PLN, algoritmos de Boosting, Combinación de métodos estadísticos y linguísticos

Abstract: This paper presents a proposal for wide–coverage Named Entity Recognition for Spanish. First, a linguistic description of the typology of Named Entities is proposed. Following this definition an architecture of sequential processes is described for addressing the recognition and classification of strong and weak Named Entities. The former are treated using Machine Learning techniques (AdaBoost) and simple attributes requiring non tagged corpora complemented with external information sources (a list of trigger words and a gazetteer). The latter are approached through a context free grammar for recognizing syntactic patterns. A deep evaluation of the first task on real corpora to validate the appropriateness of the approach is presented. A preliminar version of the context free grammar is qualitatively evaluated with also good results on a small hand–tagged corpus.
Keywords: Named entity recognition and classification, Machine Learning for NLP, Boosting algorithms, Combination of statistical and linguistic approaches

* This research has been partially funded by the Spanish Research Department (HERMES TIC2000-0335-C03-02, PETRA TIC2000-1735-C02-02, Xtract PB98-1226), by the European Comission (NAMIC IST-1999-12392), and by the Catalan Research Department (CIRIT's consolidated research group 1999SGR-150 and CIRIT's grant 1999FI 00773).

ISSN 1135-5948

1 Introduction

There is a wide consensus about that Named Entity recognition and classification (NERC) are Natural Language Processing (NLP) tasks which may improve the performance of many NLP applications, such as Information Extraction, Machine Translation, Query Answering, Topic detection and tracking, etc. Nevertheless, there is a lower agreement in what respects to the definition of what a *Named Entity* is.

From 1987 to 1999, the *Message Understanding Conferences* (MUC), devoted to Information Extraction, included a *Named Entity Recognition* task, which *de facto* determined what we usually refer to with the term *Named Entity*, and established standard measures for the accuracy of a system performing this task.

In MUC, the NERC task is divided into three subtasks: the Name Extraction (ENAMEX), the Time Extraction (TIMEX), and the Number Extraction (NUMEX) tasks. The first consists of recognizing and classifying the names for persons, locations and organizations. The second refers to the extraction of temporal expressions (dates, times), and the last one deals with monetary and percentage quantities.

The TIMEX and NUMEX tasks are much easier than ENAMEX, since temporal and numerical expressions can be detected with high accuracy using a limited amount of patterns. Name Extraction deals with an open domain and thus, presents larger difficulties to obtain comparable accuracy degrees.

Although MUC conferences established accurate evaluation criteria for NERC tasks, in all cases those criteria relied on the comparison of the system output with a hand-annotated test corpus. This has several drawbacks: First, humans annotating the reference corpus do not always agree (human annotators scored $F_1 = 96.4\%$ in MUC-6, and 97% in MUC-7). Second, the domain of the corpus and the criteria used to hand-annotate it varies from one edition to another, resulting in tasks with different levels of difficulty, which makes impossible to compare results obtained on different test corpora (in MUC-7 results were significantly lower than in MUC-6, though many participating systems were the result of improving the ones competing in MUC-6).

The techniques used in these systems cover a wide spectrum of approaches and algorithms traditionally used in NLP and AI.

Some systems rely on heavily data-driven approaches, such as Nymble (Bikel et al., 1997) which uses Hidden Markov Models, or ALEMBIC (Aberdeen et al., 1995), based on Error Driven Transformation Based Learners (Brill, 1992). Others use only hand–coded knowledge, such as FACILE (Black, Rinaldi, and Mowatt, 1998) which relies on hand written unification context rules with certainty factors, or FASTUS (Appelt et al., 1995), PLUM (Weischedel, 1995) and NetOwl Extractor (Krupka and Hausman, 1998) which are based on cascaded finite state transducers or pattern matching. Finally, there are also hybrid systems combining corpus evidence and gazetteer information (Yu, Bai, and Wu, 1998; Borthwick et al., 1998), or combining hand-written rules with Maximum Entropy models to solve correference (Mikheev, Grover, and Moens, 1998).

In this paper we propose a hybrid approach to Named Entity Recognition and Classification for Spanish. We approach the task excluding the equivalent to the NUMEX and TIMEX tasks in MUC (that is, we do not consider time or numerical expressions, which being frequent and easier to detect and classify, have the effect of raising the final accuracy figures). In addition, the task we approach is somewhat more difficult than MUC ENAMEX since we consider not only PERSON, LOCATION, and ORGANIZATION classes, but also a fourth category OTHERS which includes named entities such as documents (*Constitución*, *Ley de Arrendamientos Urbanos*), measures and taxes (*Producto Interior Bruto*, *Nasdaq*, *Impuesto sobre la Renta*), titles of art works —cinema, music, literature, painting, etc.— (*Siete años en el Tibet*, *Las Meninas*), and others.

The system uses Machine Learning (ML) components for the recognition and classification of simple entities, and a hand–written context free grammar to recognize complex entities. The Machine Learning module uses extremely simple contextual and orthographic information, while the context free grammar relies on much richer information. Some experimental evaluation is presented confirming the validity of the approach proposed. We also test whether ML systems significantly improve their performance when using external knowledge sources (such as

gazetteers or lists of trigger words).

The overall organization of the paper is the following: In section 2 the linguistic basis of the NE typology is explained, as well as the overall architecture of the process. Sections 3 and 4 are devoted to describe the algorithms and sources of information used to recognize strong and weak NEs, respectively. Section 5 describes the initial experimental evaluation of the model in a general Spanish corpus from a news agency. Finally, section 6 states the main conclusions of the work and discusses some extensions and directions for the future work.

2 A Particular Approach

2.1 Linguistic Delimitation and Proposed NE Typology

In NLP systems, *Named Entity* (NE) generically refers to expressions involving time, money or numbers, as well as person, location or organization names, which are key elements to capture *what* or *whom* the text is about, which may be important for specific applications.

The linguistic characterization of the NE is difficult, and it seems clear that is a wider set than that of Proper Nouns. There is no agreement between different systems and applications for what respects to the treatment of NE and to their conceptual and syntagmatical delimitation. Strictly speaking, one could consider a Named Entity any noun phrase with a referential value.

In this paper we present some filters in order to establish different categories of weak named entities according to their *strength*.

Many systems rely on a shallow analysis and consider only proper nouns as NE, which results in a loss of information. For instance, in the case of *"el presidente de la Cámara de Comercio"*, a typical system could extract *"Cámara de Comercio"* as an organization name, when the actual entity involved in the described event is in fact a person.

We have defined some restrictions to linguistically delimit this kind of entities. We distinguish two main types of Named Entities:

- **Strong** Named Entities, which consist of a proper noun. For instance in the sentence *"El presidente del partido conservador Renovación Nacional, Alberto Cardemil, ..."*, we find two strong NEs: *"Renovación Nacional"* (organization), and *"Alberto Cardemil"* (person).
- **Weak** Named Entities, which may contain a trigger word (e.g., *"el* **presidente** *Bush"*) and, optionally, a strong NE and a determiner (e.g., *"*__el__ *Museo de Arte Moderno"*).

2.2 Weak Named Entities and Trigger Words

The concept of trigger words is a central issue in this work. It comes from McDonald's concept of internal/external evidence (McDonald, 1996). Internal evidence consists on abbreviations accompanying the proper noun (*Inc.*, *Ltd.*, *S.A.*, *Mr.*, *Sra.*, etc.). External evidence is provided by words that point out the presence of a NE of a certain type. These words are known as trigger words and provide clues for the semantic classification of proper nouns (e.g., *"el* **juez** *Liaño"*, *"el* **atleta** *Abel Antón"*).

We extracted a list of trigger words from corpora and encyclopedic sources and classified them according to a semantic classification (Arévalo, 2001; Simón, 2001). The list contains a total of 3,759 trigger words (lemmas) and 2,500 geographical origin adjectives. These words have been considered as the referential set for identifying, classifying and analyzing weak NEs.

As we will see in the next section, trigger words are the only required elements of the weak NE, and thus, they play a central role in their processing, since they enable us not only to detect possible NEs, but also semantically and morphologically classify them, providing a rich information source for later correference resolution.

Relying on semantic and morphological criteria, two kinds of trigger words may be described:

- Internal to NE. Trigger words integrated into the strong NE. They are characterized by:
  - Being capitalized
  - Not admitting sinonymy
  - Expressing static knowledge (Nirenburg and Raskin, 1996)

  Samples of this kind of trigger words are: Geographical features (*"*__Golfo__ *Pérsico"*,

*"**Monte** Everest"*), institutions and organizations (*"**Museo** de Bellas Artes"*, *"**Ministerio** de Defensa"*), titles of documents (*"**Tratado** de Esgrima"*) and others (*"**Síndrome** de Down"*, *"**Teorema** de Pitágoras"*).

- External to NE. Trigger words not integrated into the strong NE, which express dynamic, modifiable knowledge, and assign a semantic type to the NE (e.g., *"el* **socialista** *Rodríguez Zapatero"*, *"la* **ministra** *Celia Villalobos"*, etc.).

2.2.1 Types of NEs

Weak NEs are an open conceptual class about which there is no agreement on how differentiate them from other discourse entities. In other words, the problem is which kind of noun phrases belong to NE class and which do not.

In order to cope with the problem of recognizing and classifying weak NEs we have established several degrees of *named–entitiness* taking into account semantic and syntactic filters.

From a semantic point of view we have distinguished three kinds of NEs:

- Core NEs, which have as nucleus a trigger word belonging to the referential set. We have assigned to each trigger word its EuroWordNet synset.
- Related NEs, which have as nucleus a hyponym/hyperonym/synonym of a trigger word coming from the referential set.
- General NEs, any noun phrase.

Syntactically speaking, we distinguish two types of NEs:

- Syntactically simple weak NEs: those formed by a single noun phrase (*"la ciudad asturiana de Gijón"*, *"el trío norteamericano Hanson"*) and very simple cases of coordination (*"los sindicatos USO y FSIE"*). The prototypical structure of this type of weak NEs is formed by a determiner, a trigger word and, optionally, a complement that normally includes a proper noun or a prepositional phrase. The presence of the determiner is compulsory except for the case of appositive construction, in which the determiner usually drops. See examples in table 1.
- Syntactically complex weak NEs: those formed by complex noun phrases including plurals combined with ellipsis, anaphora, relative phrases, etc. (*"Los alcaldes socialistas de Badalona, Maite Arqué, y de Sant Adrià, Jesús María Canga"*, *"los mediocampistas Paul Okon (Fiorentina, Italia), Stan Lazaridis (Birmingham City, Inglaterra) y Danny Tiatto (Manchester City, Inglaterra)"*).

Up to now we have focused on core and syntactically simple NEs. The MICE module (Module for the Identification and Classification of Entities) recognizes and classifies NEs having one of the trigger words that we have previously defined. In section 5.4 we present the results of evaluating this grammar over a subcorpus of 120 examples.

The MICE module can be extended allowing the treatment of related and general NEs (semantic extension). The treatment of syntactically complex NEs should be treated in a module or a parser that would take into account contextual information as well as complex structures.

2.3 System Architecture

In the proposed system, the Named Entity recognition task is divided in two main parts, corresponding to the processing of strong and weak entities. The first is solved using ML techniques based on context features. The second is based on syntactic patterns described in a context free grammar (CFG).

In what respects to strong NE processing, two sub-tasks must be approached: Named Entity Recognition (NER) —consisting of detecting the boundaries for each entity— and Named Entity Classification (NEC) — consisting of deciding whether the NE refers to a person, a location, an organization, etc.

We follow the approach of performing each task as soon as the necessary information is available. In this direction, NER is performed during morphological analysis, since it requires only context information on word forms, capitalization patterns, etc. In addition, NER performed at this early stage improves tagging performance, since it reduces the morphological ambiguity of the sequence and provides simpler sequences (e.g. the tagger would rarely find a sequence of proper nouns, since NER would have recognized them before as a single token).

Pattern	example
`Det + tw + PP`	*El presidente de USA*
`Det + tw + ADJ`	*El presidente ruso*
`Det + tw`	*El tenista*
`..., tw + PP, ...`	*..., campeón de Europa, ...*

Table 1: Sample patterns of syntactically simple weak NEs

The NEC task may be performed either before the tagger or after it. If performed after the tagger, it may take advantage of the morphosyntactical information provided by the tagger (lemmas and Part–of–Speech tags). In our case, we have chosen to perform it after the morphological analysis and before the tagging because there was no significant improvement in NEC performance when using the information provided by the tagger. This is because word context information required to classify a NE usually involve the existence of prepositions, determiners, and punctuation marks, all of them features that can be easily captured by the word form instead of by lemma or PoS.

Finally, the weak NE recognition and classification, since it is performed through a CFG specifying syntactical patterns, must necessarily be performed after tagging, in order to have PoS tags available.

Summarizing, the architecture of our system, integrated in the whole language processor system is presented in Figure 1.

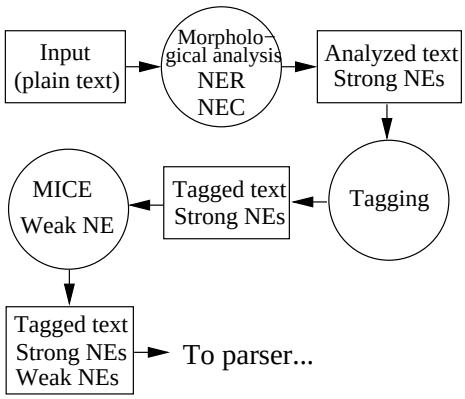

Figure 1: Architecture of the Language Processing system, including weak and strong NE recognition and classification

3 Recognition and Classification of Strong Named Entities

3.1 Description of the NER and NEC tasks

NER and NEC tasks will be performed sequentially but independently inside the morphological analysis of the text. It is also possible to approach a joint resolution of both tasks with the aim of allowing both processes to collaborate, but up to now we have restricted ourselves to analyse separately both processes.

Formally, NER can be seen as the task of segmenting a sequence of words in non–overlapping and non–recursive chunks (i.e., the NEs). From this point of view, a NE is characterized by its starting and ending points, and, in terms of binary decisions, we need only a classifier to decide if a certain word in the sentence opens a new NE and another one to decide if a certain word closes an already opened NE. From now on, we will refer to this scheme as OpenClose. Alternatively, NER can be seen as the task of tagging a sequence of symbols codifying whether each particular word is the beginning of a NE (B tag), if it is a component of a NE, but not the first word (I tag), or if it is outside a NE (O tag). Note that at least three tags are necessary to unambiguously annotate an arbitrary sentence, and, in particular, the existence of consecutive NEs. This annotation scheme is a variant of the one introduced by (Ramshaw and Marcus, 1995) which has been widely used for syntactic chunking. From now on, we will refer to this approach as IOB. It is worth noting that, in this case, a single classifier is needed to perform the IOB task. Such classifier must decide which is the correct tag (I, O, or B) for each word in the sentence.

See figure 2 for an example of both types of annotation. Note that while in the IOB annotation each word has exactly one tag, in the OpenClose annotation a single word can be an open and close point simultaneously.

There are several ways of using the classifiers so far described to perform the NER task. The most simple and efficient consists in exploring the sequence of words in a certain direction (e.g., from left to right) and applying the *open* and *close* classifiers (or, alternatively, the IOB classifier) coherently.

"Las críticas del expresidente del (_*Gobierno*_) (_*Felipe González*_) *a la política antiterrorista del* (_*PP*_) *, ..."*

*"Las*_O *críticas*_O *del*_O *expresidente*_O *del*_O *Gobierno*_B *Felipe*_B *González*_I *a*_O *la*_O *política*_O *antiterrorista*_O *del*_O *PP*_B *,*_O *..."*

Figure 2: Annotation of the NEs of a training sentence, according to the OpenClose and IOB tagging schemes, respectively

This greedy approach is linear in the number of words of the sentence.

Given that the classifiers are able to provide predictions, which may be translated into probabilities, another possibility is to use dynamic programming (in the style of the Viterbi algorithm) for assigning, the sequence of tags (and thus, the set of NEs) that maximize the probability of the sequence of observed words. However, it has to be noted that the nature of the problem imposes some coherence constraints inside the annotation of the sequence (e.g., a tag I is not admissible after a tag O, the open–close parentheses cannot overlap nor define embedded structures, etc.) that have to be considered inside the tagging algorithm. In this direction, the work on syntactic chunking (Punyakanok and Roth, 2000) describes two approaches (an extension of traditional HMM models, and an extension of the constraint satisfaction formalism) to make sequential inference with the outputs of classifiers under certain structural constraints imposed by the problem. Yet another approach consists of learning more specialized classifiers in order to guide the heuristic search towards the best annotation sequence (Carreras and Màrquez, 2001). For instance, a classifier that predicts whether the sequence of words between a pair of potential starting and ending points forms a NE or not, can be used to score alternative sequence taggings.

In all these approaches, the interactions between the decisions taken by the classifiers along the word sequence is something that has to be carefully considered, since even in the simplest case, the tagging of a certain word propagates relevant information to the left context of the following word to be tagged (e.g., tagging a word as the beginning of a NE increases the probability for the next word being inside a NE).

In this work, for simplicity and efficiency reasons, the simplest approach discussed above has been followed. The benefits and drawbacks of the rest of tagging schemes will be studied in future work.

Finally, it is worth noting that NEC is simply a classification task, consisting of assigning the NE type to each potential, and already recognized NE. In this case, all the decisions are taken independently, and the classification of a certain NE cannot influence the classification of the following ones.

3.2 Learning the Decisions

The AdaBoost algorithm (Freund and Schapire, 1997) has been used to learn all the binary decisions involved in the annotation of strong named entities.

AdaBoost is a general method for obtaining a highly accurate classification rule by combining many *weak* classifiers, each of which may be only moderately accurate. In designing our system, a generalized version of the AdaBoost algorithm has been used (Schapire and Singer, 1999). This algorithm, which uses weak classifiers with confidence–rated predictions, is briefly sketched in this section. We assume that the reader is familiar with the related concepts —see (Schapire and Singer, 1999), otherwise.

This particular boosting algorithm is able to work efficiently in very high dimensional feature spaces, and has been applied, with significant success, to a number of practical problems, including text filtering and routing, "ranking" problems, several Natural Language Processing tasks, image retrieval, and medical diagnosis. See (Schapire, 2001) for details. Among the NLP and text related problems, the following deserve a special mention: Part–of–speech tagging and PP–attachment (Abney, Schapire, and Singer, 1999), text categorization (Schapire and Singer, 2000), word sense disambiguation (Escudero, Màrquez, and Rigau., 2000), statistical parsing (Haruno, Shirai, and Ooyama, 1999), and clause identification (Carreras and Màrquez, 2001).

3.2.1 AdaBoost Algorithm

As previously said, the purpose of AdaBoost is to find a highly accurate classification rule by combining many *weak hypotheses* (or weak classifiers). The weak hypotheses are learned sequentially, one at a time, and, conceptually, at each iteration the weak hypothesis is bi-

ased to classify the examples which were most difficult to classify by the preceding weak hypotheses. The final weak hypotheses are linearly combined into a single rule called the *combined hypothesis*.

Let $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ be the set of m training examples, where each instance x_i belongs to an instance space $\mathcal{X}$ and $y_i \in \{-1, +1\}$ is the class or label associated to x_i. The generalized AdaBoost algorithm for binary classification (Schapire and Singer, 1999) maintains a vector of weights as a distribution D_t over examples. At round t, the goal of the weak learner algorithm is to find a weak hypothesis $h_t : \mathcal{X} \rightarrow \mathbb{R}$ with moderately low error with respect to the weights D_t. In this setting, weak hypotheses $h_t(x)$ make real–valued confidence–rated predictions. Initially, the distribution D_1 is uniform, but after each iteration, the boosting algorithm increases (or decreases) the weights $D_t(i)$ for which $h_t(x_i)$ makes a bad (or good) prediction, with a variation proportional to the confidence $|h_t(x_i)|$. The final hypothesis, $f : \mathcal{X} \rightarrow \mathbb{R}$, computes its predictions using a weighted vote of the weak hypotheses $f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$. For each example x, the sign of $f(x)$ is interpreted as the predicted class (-1 or $+1$), and the magnitude $|f(x)|$ is interpreted as a measure of confidence in the prediction. Such a function can be used either for classifying new unseen examples or for ranking them according to the confidence degree.

3.2.2 Weak Rules

In this work we have used domain–partitioning (or decision–tree–like) weak hypotheses with real–valued predictions. In the most simple case, such hypotheses are rules that test the value of a boolean predicate and make a prediction based on that value. The predicates used refer to the attributes that describe the training examples (e.g. "the word *street* appears to the left of the named entity to be classified"). Formally, based on a given predicate p, weak hypotheses h are considered that make predictions of the form: $h(x) = c_0$ if p holds in x, and c_1 otherwise. Where the c_0 and c_1 are real numbers. See (Schapire and Singer, 1999) for the details about how to calculate the c_i values given a certain predicate p in the AdaBoost framework.

This type of weak hypotheses can be seen as extremely simple decision trees with one internal node and two leafs, which are sometimes called *decision stumps*. Furthermore, the criterion for finding the best weak hypothesis (with a single feature) can be seen as a natural splitting criterion and used to perform decision–tree induction. In this work, we have extended weak hypotheses to arbitrarily deep decision trees following the idea suggested in (Schapire and Singer, 1999), and considering an additional parameter in the learning algorithm that accounts for the depth of the decision trees induced at each iteration.

For instance, figure 3 contains two real examples of weak classifiers learned for the NER task described in section 3.1. In this particular case, the weak hypotheses are depth–3 decision trees learned by the *open* (left) and *close* (right) classifiers. Each internal node contains a test on a binary feature. The descendant node for the positive answer is drawn to the right, while the descendant for the negative answer is drawn below (and also to the right). The leafs contain the real–valued predictions for opening (or closing) a NE in the particular example tested. For instance, the second branch of the left tree states that if that the focus word is capitalized, its previous word is outside a NE and it is not the case that the word two positions to the left is not available (out of the sentence), and thus the current word is at least the third word in the sentence, then the prediction gives positive evidence (1.058) to the decision of opening a NE in this point. Similarly, the fifth branch of the right tree states that if the first word to the right of the focus word is not capitalized but it is a functional word, and the second word to the right is capitalized, then the NE should not be finished in the current position (prediction -0.976). Note that if, instead, the second word to the right is not capitalized then the prediction is the reverse (sixth branch, positive prediction for closing 0.252).

These more complex weak hypotheses allow the algorithm to work in a higher dimensional feature space that contains conjunctions of simple features, and this fact has turned out to be crucial for improving results in the present domain.

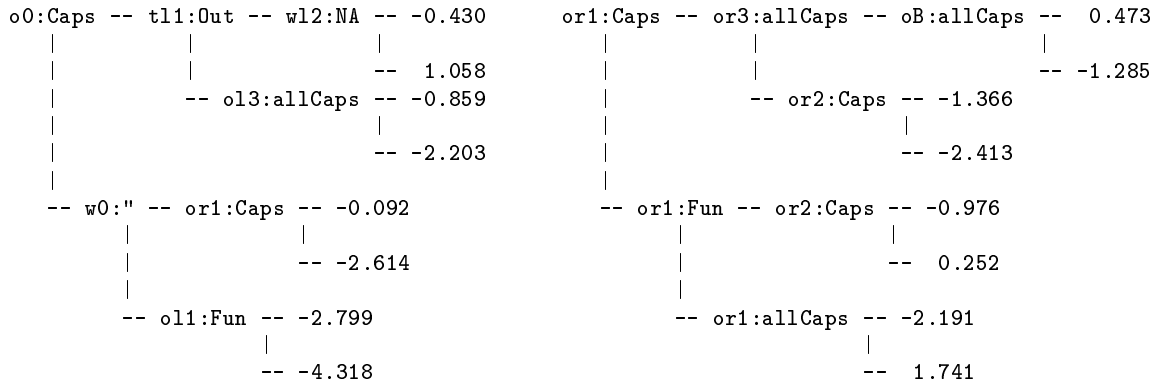

Figure 3: Example of weak rules (depth–3 decision trees) for the *open* and *close* decisions (left and right, respectively)

3.3 Information Sources and Features

The AdaBoost algorithm described above are applied both to NER and NEC tasks. As described in section 2.3, both tasks use basically context information which can be found in untagged text.

Nevertheless, it seems logical that further generalization may be achieved if the learners can use external knowledge sources such as lists of person or location names, or lists of specific trigger words. For instance, if the corpus contains the sequences "*el ministro de Economia*" and "*el secretario de Interior*", it may lack evidence to be able to infer a rule stating that something after the word "*ministro*" is a Named Entity, but if the system is provided with a list of words referring to politician positions, the corpus may be regarded as "*el* `<politician>` *de Economia*" and "*el* `<politician>` *de Interior*", which is more likely to enable the inference of a rule stating that after a `<politician>` trigger word, is probable the occurrence of a Named Entity.

Thus, the features used to take the decision in NER and NEC tasks may be divided into two broad classes:

- Context features, referring to orthographic properties of the words in context, word forms, word patterns, context patterns, etc.
- External knowledge features, relying on lists of person and location names, and lists of classified trigger words.

The trigger word list used in this work consists of 5,007 geographical origin adjectives (*catalán*, *inglesas*, *árabe*, ...), 281 location words (*río*, *selva*, *villa*, ...), 189 organization words (*comité*, *fábrica*, *sucursal*, ...), 1,483 person words (*actor*, *escritora*, *soberano*, ...) and 573 words denoting other NE classes (*estatua*, *gaceta*, *huracán*, ...). Since some forms are ambiguous, (e.g. *productora* may denote a person or an organization) the total number of different forms is 7,427.

The gazetteer contains 10,560 names, 2,760 of which are multiwords (e.g *alcalá_de_henares*). 6,233 are marked as geographical names, 2,060 as person first names, and 2,780 as person family names. Up to 496 entries are ambiguous, most of them between geographical and surname categories[1].

Due to the nature of the AdaBoost, all features are binarized, that is, there is a feature for each possible word form appearing at each position in the context window. The feature is *active* or *inactive* in a given example depending on whether the context contains that word in that position. Although this creates large feature spaces, the AdaBoost algorithm is able to deal with such dimensionality appropriately (i.e., efficiently and preventing overfitting to the training examples).

The features used by each of both tasks are more detailed below.

3.3.1 NER Features

Extremely simple features have been used for training the classifiers involved in the NER task, all of them referring to the context[2].

[1] This is quite logical, since many Spanish surnames are location names.

[2] Contrary to the NEC task, it has been empirically observed that the addition of knowledge from gazetteers and trigger words provides only very weak evidence in deciding the correct segmentation of a NE.

However, slightly different features and example definitions have been followed depending on the annotating scheme: IOB or OpenClose.

Since the basic AdaBoost algorithm is designed for binary classification problems, we have binarized the 3–class IOB problem by creating one binary problem for each tag. Therefore, each word in the training set, labelled with the X tag (X in {I,O,B}), defines an example, which is taken as positive for the X–classifier, and negative for the rest. The following features are taken into account for representing these examples:

- The form and the position of all the words in a window covering three words to the left and three words to the right, and including the focus word (e.g., *the word* estadio *appears one position to the left of the focus word*).
- An orthographic feature and the position of all the words in the same [-3,+3] window. These orthographic features are binary and not mutually exclusive, and take into account whether the $\pm i$_th word: *"is capitalized"*, *"contains only uppercase letters"*, *"contains numbers"*, *"contains only numbers"*, *"is an alphanumeric expression"*, *"is a roman number"*, *"contains dots"*, *"contains dashes"*, *"is an acronym"*, *"is an initial"*, *"is a punctuation mark"*, *"is a single character"*, *"is a functional word"* (some fixed prepositions and determiners), and *"is a web or an email address"*.
- The I, O, B tags of the three preceding words.

For the OpenClose scheme, two classifiers must be learned.

The *open* classifier is trained with the words at the beginning of the NEs as positive examples, and the words that are outside the NEs as negative examples. Note that the words inside a NE are not codified as examples. The feature codification of these examples is the same as for the IOB case.

The *close* classifier is trained only with the examples coming from words that are components of a NE, taking the last word of each NE as a positive example, and the rest as negative examples. In this case, the decision of whether a certain word should close a NE strongly depends on the sequence of words between the word in which the NE starts and the current word (i.e., the structure of the partial NE). For the words in a [-2,+3] window outside this sequence, exactly the same features as in the IOB case have been considered. The specific features for the inner sequence are the following:

- Word form and orthographic features of the current word and of the word starting the NE.
- Word form and orthographic features of the words inside the sequence taking its position with respect to the current word.
- Length in words of the sequence.
- Pattern of the partial entity, with regard to capitalized (M) or non-capitalized (m) words, functional words (F), punctuation marks (P), numbers (#), quotations (C), and others (a). Sample patterns can be found in table 2.

Pattern	example
M_M	*San* **Sebastián**
C_M_M_m_F	*" Congreso Europeo sobre* **la**
M_F_M_M_F	*Basílica de Santa María* **de**
M_#	*Eurocopa* **2000**

Table 2: Sample patterns for training the *close* classifier

3.3.2 NEC Features

The NEC classifier uses a set of attributes that can be grouped in the following classes:

- Context word features: Form and position of each word in a window of three words left and right of the entity being classified (e.g. *the word* presidente *appears two positions to the left of the NE*).
- Bag-of-words features: form of each word in a window of five words left and right of the entity being classified. (e.g. *the word* banco *appears in the context*).
- NE features: Length (in words) of the entity being classified, pattern of the entity, with regard to acronyms (A), numbers (D), capitalized words (M), preposi-

Since the results of the whole NER task were not improved at all, we have not included these experiments in the paper.

tions (p), determiners (d), and punctuation marks. Some sample patterns can be found in Table 3.

Pattern	example
M_M	*José Pérez*
M_p_M	*Banco de Inglaterra*
A	*IBM*
M_d_M_,_A	*Mudanzas el Rapido, S.A.*
M_DDDD	*Eurocopa 2000*

Table 3: Sample Named Entity patterns.

- Trigger word features: Class and position of trigger words in a window of three words left and right of the entity being classified. The pattern, with regard to punctuation marks, prepositions (p), determiners (d), trigger words denoting person, location, organization, or other entities (PERS, LOC, ORG, OTH), and trigger words denoting geographical origin (GENT), of the immediate left context of the entity being classified. Sample context patterns are presented in Table 4.

Pattern	example
PERS_p	*presidente de* *ministro de*
d_ORG	*la empresa* *la promotora*
ORG_GENT	*empresa catalana* *promotora alemana*
LOC_p	*ciudad de* *cordillera de*

Table 4: Sample trigger word patterns for NE left context.

- Gazetteer features: Class (geographical, first name, or surname) and position of gazetteer words in a window of three words left and right of the entity being classified. (e.g. *a location name appears three positions to the right of the NE*). Class in gazetteer of the NE being classified and class in the gazetteer of the NE components.

4 Parsing of Weak Named Entities

4.1 Linguistic Motivation

As we have seen in section 2.2, NEs present syntactic and semantic features that can be detected by a computational grammar. We have developed a grammar for the recognition and classification of weak NEs that takes into account all these features. The main purpose that we had when we developed this grammar was to improve the quality of the semantic and morphological tagging of NEs, detecting not only the strong NEs (recognized in a previous stage) but the whole of the entity involved (*el presidente de EEUU*).

4.2 A Grammar for Recognizing Complex Named Entities

The task of recognizing and classifying Complex NEs is solved with the creation of a context free grammar, which has been enriched with the semantic information of the trigger words. The grammar builds up chunks, corresponding to weak NEs, and assigns them a tag containing semantic information coming from the trigger word and morphological information coming from the determiner.

The grammar basically consists of the combination of the morphological tags, literals (trigger words) and strong proper nouns. We distinguish two kinds of rules: lexical and syntactic rules.

Lexical rules allow us to assign semantic tags to the trigger words:

```
tw-politician ==> ncms000(alcalde).
tw-politician ==> ncfs000(alcaldesa).
tw-politician ==> ncfs000(diputada).
tw-politician ==> ncfp000(diputadas).
tw-politician ==> ncms000(diputado).
tw-politician ==> ncmp000(diputados).
```

Syntactic rules are the basic morphosyntactic patterns that allow us to recognize weak NEs. For instance, the following rule, states that a weak entity referring to a politician person (npmss10) can be composed by a determiner followed by a *politician* trigger word, a preposition, and a strong named entity denoting a location.

```
npmss10 ==> det, tw-politician,
            prep, pn-loc.
```

For instance, the expression *El alcalde de Barcelona* would be analyzed by the morphological analyzer as:

El$^{\text{det}}$ alcalde$^{\text{noun}}$ de$^{\text{prep}}$ **Barcelona**$^{\text{LOC}}$

After applying the lexical rules of the grammar it would be rewritten as:

$$[\text{El}^{\text{det}}\ \text{alcalde}^{\text{tw-POLITICIAN}}\ \text{de}^{\text{prep}}\ \textbf{Barcelona}^{\text{pn-LOC}}]_{\text{npmss10}}$$

Finally, after applying the syntactic rules the whole NE is recognized and classified with the tag **npmss10**[3]: [El alcalde de Barcelona]$_{\text{npmss10}}$

5 *Evaluation*

5.1 Spanish Corpus

The corpus used for the evaluation of the whole Named Entity processing system is a collection of over 3,000 news agency articles totalling 802,729 words, which contain over 86,000 hand tagged strong Named Entities (Arévalo and Buffi, 2001).

A corpus subset containing 65,000 words (4,800 strong Named Entities) is reserved for evaluation tests and the remaining is used as training material.

For the NER task a subset of the training set of slightly less than 100,000 words has been used for training. It has been empirically observed that using a bigger corpus does not result in a better performance for the task, while the number of examples and features greatly increase. The precise number of examples and features derived from the training corpus for each binary decision of the NER task is described in table 5. It has to be noted that the features occurring less than 3 times in the training corpus have been filtered out.

For the NEC task, the whole training set, consists of some 81,000 NE occurrences, has been used. According to the features defined in section 3.3, these examples produce near 89,000 features. Again, only the features occurring three or more times in the training corpus are considered, reducing the feature space to a figure between 21,000 and 23,000 depending on the feature set used in the experiment.

5.2 Experimental Methodology

We trained the system using different feature sets and number of learning rounds, learning models of different complexities, ranging from stumps (simple decision rules, i.e. decision trees of depth 1) to decision trees of depth 4.

Each variant was evaluated on the hand tagged 65,000-word test corpus both for NER and NEC classifiers. The evaluation measures for NER are: number of NE beginnings correctly identified (B), number of NE endings correctly identified (E), and number of complete NEs correctly identified. In the last case, recall (R, number of entities correctly identified over the number of expected entities), precision (P, number of entities correctly identified over the number of identified entities), and F-measure ($F_1 = 2 \cdot P \cdot R/(P + R)$) are computed.

The evaluation measures for NEC task include the evaluation of the binary classifiers for each category, which is measured in terms of accuracy (percentage of entities correctly accepted/refused), as well as the evaluation of the combined classifier, which proposes the final decision based on the predictions of all binary classifiers. The combination performance is measured in terms of recall (R, number of correctly classified entities over total number of entities to classify), precision (P, number of correctly classified entities over number of proposed classes), and F-measure ($F_1 = 2 \cdot P \cdot R/(P + R)$). The accuracy ($Acc$) of the system when forced to choose exactly one class per entity is also evaluated.

Finally, the complete system is evaluated by testing the performance of the NEC classifier on the output of the NER task, and checking how the errors introduced by the latter affect the performance of the former.

It is important to notice that all the evaluations are made under a worst-case approach, that is, the NER is not considered to produce a correct output unless the entity boundaries are recognized exactly as they are annotated in the corpus. Similarly, the NEC task is considered to incorrectly classify an entity whose boundaries have been misdetected by the NER .

5.3 Strong Named Entities

In this section we present the results obtained in the experiments and draw some conclusions

5.3.1 NER Results

Figure 4 contains the performance plots (F_1 measure) with respect to the number of rounds of the AdaBoost algorithm (i.e., the

[3]These tags provide morphological and semantic information about the expression. In particular, **npmss10** means that this expression is a proper noun (**np**mss10) of masculine gender (np**m**ss10) and single number (npm**ss**10) and that its semantic type is politician person (npmss**10**).

Problem	#Examples	#Features	#Pos.examples
open	91,625	19,215	8,126 (8.87%)
close	8,802	10,795	4,820 (54.76%)
I	97,333	20,526	5,708 (5.86%)
O	97,333	20,526	83,499 (85.79%)
B	97,333	20,526	8,126 (8.35%)

Table 5: Sizes and proportions of positive examples in the NER binary decisions

number of weak hypotheses combined) in all the binary decisions of the NER task. As can be observed in this figure, decision trees perform significantly better than stumps on taking NER binary decisions. However, increasing the depth of the trees provides smaller gain. Also, it can be noticed that all NER binary classifiers are quite accurate, with F_1 measure over 96% (see Figure 4). The learning curves present a satisfactory behaviour, with no significant overfitting with larger number of rounds, and achieving maximum performance after a quite reduced number of rounds.

Table 6 contains the results of the OpenClose and IOB approaches on the whole NER task, using depth–3 weak rules. It can be observed that both variants perform significantly better than the baseline MACO+ NE module (Carmona et al., 1998). The MACO+ NE module is a heuristic rule based NE recognizer. The rules, which have been manually developed, basically mark as a NE any (intrasentence) sequence of capitalized words (possibly) connected by some functional words (such as *el*, *la*, *de*, *del*, ...). Words at the beginning of a sentence (and thus capitalized) are considered possible NEs either if they are not found in a form morphological dictionary, or if they are found with noun or adjective part-of-speech. The rules of the MACO+ module cover a significant part of the NE cases of the test corpus with a quite high accuracy. However, they are very general rules and the nature of the recognizer makes difficult to manually include exceptional cases and more particular rules.

The performance of the OpenClose scheme is slightly worse than the IOB. This can be explained by the fact that when the *close* classifier makes a mistake by deciding not to end a NE when it should, then the prediction of the same classifier in the following words can be very bad, since, in general, they will be words outside a NE that the *close* classifier has not seen in the training set. This is confirmed by an additional experiment consisting of combining the OpenClose scheme with the I binary classifier of the IOB scheme. This new tagging scheme, labelled OpenClose&I in table 6, consists of applying the OpenClose scheme but after each time the *close* classifier makes a negative prediction, the I classifier is asked to confirm if the following word is still inside the NE. If the I classifier gives a positive answer then the process continues normally, otherwise it is assumed that the *close* classifier was wrong. The positive answers of the *close* classifier are never questioned, since it is very accurate in its predictions. As it can be seen in the table, this third scheme achieves the best results on the task.

Finally, table 7 presents the NER results depending on the length of the NE to recognize, as well as depending on whether the entity begins with uppercase or lowercase letter. As it could be expected, the performance degrades with the length of the sequence to be detected (specially with respect to the recall level). However, a reasonable high accuracy can be expected for NEs of length up to six words. The set of NEs that begin with a lowercase word represents a very challenging problem for the NER module, specially due to the very shallow semantic treatment of the training examples (captured only through the word forms, without any kind of generalization). We find very remarkable the accuracy achieved by the system on this subset of words (85.40%). The recall level is significantly lower (63.93%), basically because in many occasions the *open* classifier does not have enough evidence to start a NE in a lowercase word. Probably, a better precision–recall balance can be obtained by thresholding the *open* classifier in a less conservative value.

5.3.2 NEC Results

The binarization of the NEC problem used in this work is the simplest possible, and con-

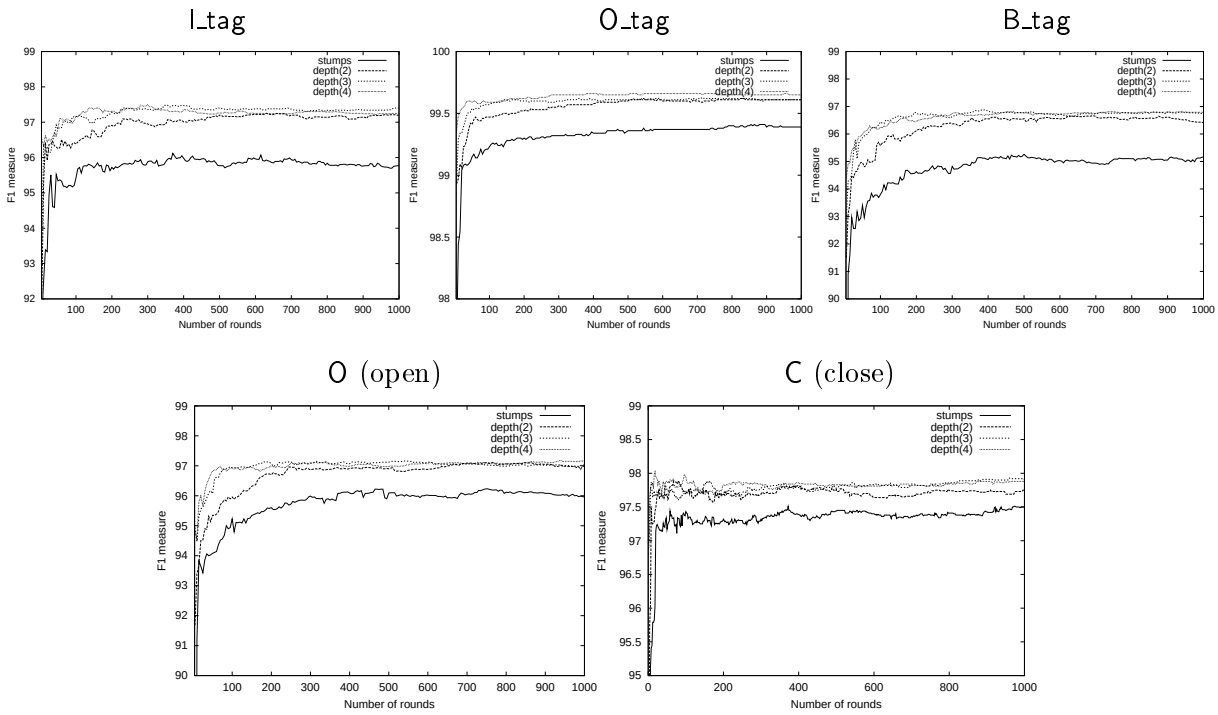

Figure 4: F_1 measure with respect to the number of rounds of all binary decisions involved in the NER task.

Method	B	E	P	R	F_1
MACO+	90.83%	87.51%	89.94%	87.51%	88.71%
OpenClose	94.61%	91.54%	92.42%	91.54%	91.97%
IOB	95.20%	91.99%	**92.66%**	91.99%	92.33%
OpenClose&I	**95.31%**	**92.14%**	92.60%	**92.14%**	**92.37%**

Table 6: Results of all methods in the NER task

sists of a binary classifier for each class (*one–per–class* scheme). The binary classifiers accept/reject (with a confidence degree) the NE as belonging to each class. Then, the confidence degrees are combined in a final decision. The NEC binary decisions present all an accuracy between 91% and 97% (see Figure 5). As in the NER task, decision trees give significantly better results than stumps.

With respect to the complete NEC system, the combination of binary decisions is performed selecting the classes to which binary predictors assigned a positive confidence degree. The system can be forced to give exactly one prediction per NE by selecting the class with higher confidence degree. The results of all NEC systems are presented in Table 8. As a baseline we include the results that a dumb *most-frequent-class* classifier would achieve.

The same table presents the results when the models include features obtained from external knowledge sources, such as lists of trigger words and gazetteers. In all cases, the use of this extra information improves the performance of the system, both in the binary decisions and in the final combination. It is remarkable that the best result is achieved when both external resources are used, pointing out that each of them provides information not included in the other.

The *basic* row refers to the model using context word, bag-of-words and NE features as described in section 3.3. Results obtained when adding trigger word features and gazetteer features are found in rows coded *tw* and *gaz*, respectively. Note that although the individual performance of the binary classifiers was over 91%, the combined classifier achieves an accuracy of about 88%.

Finally, Table 9 presents the result of the whole system, acting as a NER–NEC pipeline in which entities recognized by the first component are classified by the second. Per-

Subset	#NE	B	E	P	R	F_1
length=1	2,807	97.04%	95.80%	94.64%	95.65%	95.15%
length=2	1,005	99.30%	93.73%	94.01%	93.73%	93.87%
length=3	495	93.74%	88.69%	91.65%	88.69%	90.14%
length=4	237	89.45%	84.81%	84.81%	84.81%	84.81%
length=5	89	87.64%	76.40%	77.27%	76.40%	76.84%
length=6	74	93.24%	79.73%	81.94%	79.73%	80.82%
length=7	22	59.09%	54.55%	60.00%	54.55%	57.14%
length=8	22	77.27%	68.18%	88.24%	68.18%	76.92%
length=9	11	90.91%	72.73%	80.00%	72.73%	76.19%
length=10	3	66.67%	33.33%	50.00%	33.33%	40.00%
uppercase	4,637	96.42%	93.25%	92.81%	93.25%	93.03%
lowercase	183	67.21%	63.93%	85.40%	63.93%	73.13%
TOTAL	4,820	95.31%	92.14%	92.60%	92.14%	92.37%

Table 7: Results of OpenClose&I on different subsets of the NER task

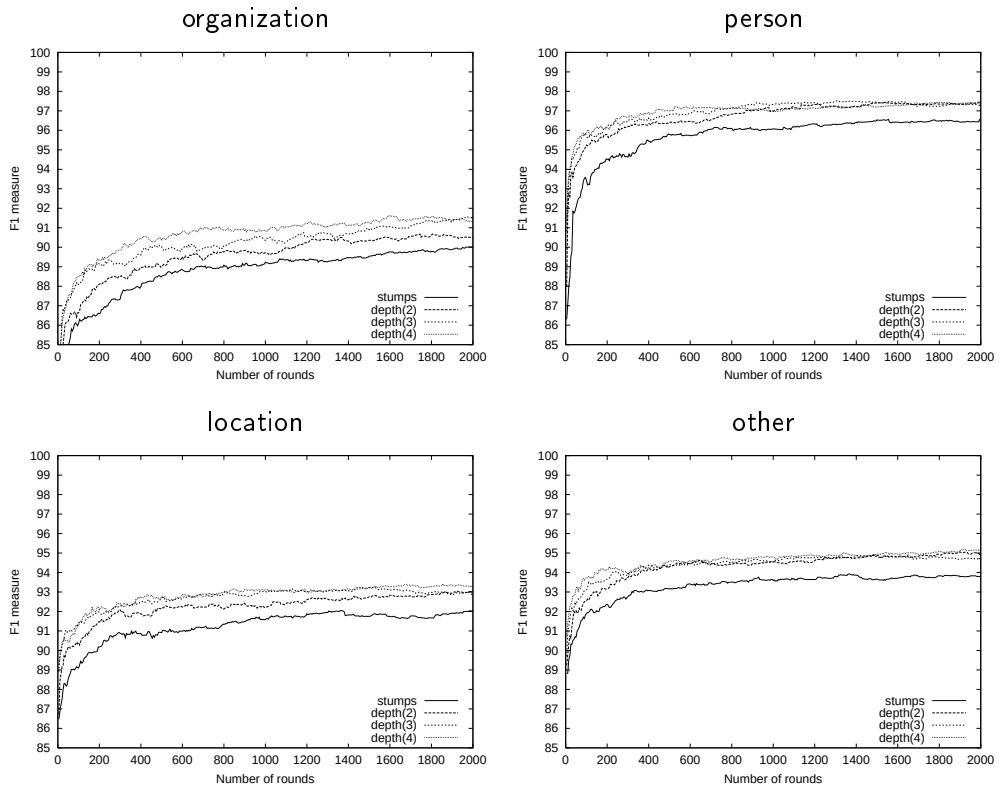

Figure 5: F_1 measure with respect to the number of rounds of all binary decisions involved in the NEC task.

formance is rather lower, due to the error propagation and to the worst-case evaluation, which counts as misclassifications the entities incorrectly recognized.

5.3.3 Some Comments on the Corpus Annotation

The presented results are not easy to compare with other systems, since the train and test corpus used may greatly affect the final results, specially in two aspects: the criteria used to annotate them and the amount of noise they contain. As a sample we present some particular cases found in the corpus we used.

In Tables 10 and 11, sample NE occurring in the corpus are presented. The NE are typed in boldface, with a superindex indicating their class. The different components of a same NE are connected with underscores.

Table 10 illustrates some cases of what could be called *non-standard* named en-

Features	*P*	*R*	*F*1	*Acc*
Most frequent	39.78%	39.78%	39.78%	39.78%
basic	90.19%	84.44%	87.22%	87.51%
basic+tw	90.11%	84.77%	87.36%	88.17%
basic+gaz	90.25%	**85.31**%	87.71%	88.60%
basic+tw+gaz	**90.61**%	85.23%	**87.84**%	**88.73**%

Table 8: Results of all feature sets in the NEC task

Features	*P*	*R*	*F*1	*Acc*
Most frequent	37.47%	37.47%	37.47%	37.47%
basic	83.84%	79.39%	81.55%	81.85%
basic+tw	85.08%	79.30%	82.09%	82.15%
basic+gaz	85.05%	79.57%	82.22%	82.15%
basic+tw+gaz	**85.32**%	**79.80**%	**82.47**%	**82.31**%

Table 9: Results of all feature sets in the NEC task when run after the NER classifier

tities. The top rows show entities beginning with or including non-capitalized words. The lower rows present some cases in which several entities are involved causing the boundaries to be difficult to establish, even for humans, such as cases where a location entity appears after another entity (*la* **Calle_Balmes** *de* **Barcelona**), or when the location name is a part of the entity (**Ayuntamiento_de_Madrid**), or cases of several consecutive entities with no punctuation marks in between.

Table 11 illustrates some noise found in the corpus due to annotation errors, which not only difficult the model learning phase, but also affect the evaluation. Noise appears in the form of ill-bounded entities, also as well-bounded ill-classified entities, and finally as entities which actually fall into the definition of *weak entity* but that have been annotated as *strong*.

5.4 Weak Named Entities

The weak NE processing has been evaluated qualitatively due to its early development stage. For doing that, the same test corpus used for the strong NE evaluation has been used as a reference.

The corpus used for the evaluation of the MICE module is a subset of 120 examples randomly chosen from the test corpus. The subset consists of 45 expressions referring to persons, 45 to organizations, 27 to geographical places and 3 ambiguous entities, and contains a good variety of the different syntactic structures for named entities found in the test corpus.

When evaluating the results of analyzing this subcorpus we have observed the following cases:

- Errors due to previous processes, which can be caused by PoS tagging (3 occurrences) or by the strong NE recognition and classification modules (7 occurrences).
- Errors made by MICE module, due to non-covered syntactic patterns (3 occurrences) or caused by the occurrence of trigger words not included in the used lists (4 occurrences).
- Expressions correctly detected and classified (103 occurrences). We have considered as correctly classified those expressions that contained trigger words that were semantically ambiguous. For instance, "*colegio*" in Spanish is an ambiguous trigger word that can refer to a geographical or to an organization entity. Since a context free grammar does not permit disambiguation in this case, because world knowledge is required, we decided to tag all these types of entities with a top tag that indicates that the expression detected is an entity, without specifying its type.

From this evaluation we can conclude that when working over restricted data (all the examples of the subcorpus were weak NEs manually tagged) the performance of MICE is quite good. To evaluate it over unrestricted text, a hand tagged corpus must be anno-

Non-capitalized words
camping_Bellavista^{org}
pinturas_API_,_S.A.^{org}
glorieta_de_Tanos^{loc}
rotonda_de_Campuzano^{loc}
Derecho_civil^{oth}

Many entities involved
colegio_de_la_Asunción^{org} *de* **Valladolid**^{loc}
Tribunal_Superior_de_Justicia_de_Galicia^{org}
Delegación_Provincial_de_Educación_de_Guadalajara^{org}
el defensa del **Córdoba_CF**^{org} **Juan_González_Maestre**^{per} **"_Juanito_"**^{per}

Table 10: Examples of *non-standard* entities in the corpus

Boundary/Classification errors
Comandancia_de_La_Rioja^{org}
should be: **Comandancia**^{org} *de* **La_Rioja**^{loc}
Puesto_de_Murillo^{loc} *de* **Río_Leza**^{loc}
should be: **Puesto**^{loc} *de* **Murillo_de_Río_Leza**^{loc}
red europea de parques naturales **"_Natura_2000_"**^{oth}
should be: *red europea de parques naturales "* **Natura_2000**^{org} *"*
Ley_del_Procurador^{oth} *del* **Común**^{org} *de* **Castilla_León**^{loc}
should be: **Ley_del_Procurador_del_Común**^{oth} *de* **Castilla_León**^{org}
delegado para **Galicia**^{loc} *del* **Poder_General**^{oth} *del* **Poder_Judicial**^{oth}
should be: *delegado para* **Galicia**^{loc} *del* **Poder_General_del_Poder_Judicial**^{org}
ministro de **Trabajo**^{oth} *y* **Asuntos_Sociales**^{oth}
should be: *ministro de* **Trabajo_y_Asuntos_Sociales**^{org}
conselleiro de **Justicia_,_Interior_y_Relaciones_Laborales**^{oth}
should be: *conselleiro de* **Justicia_,_Interior_y_Relaciones_Laborales**^{org}

Actually weak NE
estación_del_metro_de_Joan_XXIII^{loc}
Ministerio_de_Defensa_canadiense^{org}

Table 11: Examples of noisy entities in the corpus

tated, according to the linguistic criteria defined in section 2.2.1.

6 Conclusions and Further Work

We have presented a Named Entity recognition and classification system based on the distinction between *strong* and *weak* entities. The processing of the former is approached via Machine Learning techniques, while the latter is covered by hand coded linguistic knowledge in the form of a CFG.

Also, a linguistically motivated typology for Named Entities has been defined. We distinguished two main types of NEs. Strong entities are basically those considered in MUC and weak entities have a more complex syntactic structure and contain a non-integrated trigger word.

From an internal point of view, weak NEs have been semantically and syntactically classified. Semantically we have distinguished core, related and general NEs. The first class contains prototypical trigger words, while the second contains trigger words related through different EuroWordNet relationships, and the latter has as nucleus any noun not marked as trigger word. This classification allows us to consider different degrees of named-entitiness, being those in the core class the most paradigmatic ones. This information could be used for assigning relevance degrees to the entities found in a text. From a syntactic point of view, we distinguished between single and complex weak NEs. Currently, MICE detects core and syntactically simple weak NEs.

The performance of the learning algorithms is quite good, providing state of the art NE recognizers and classifiers, though the comparison with other systems results is difficult since the used evaluation corpora as well as the criteria used to annotate them are not homogeneous.

As further lines of work to improve the performance and quality of the system, we can sketch the following:

- NER: The recognizer use a greedy algorithm in which each word is tagged as beginning (**B**), internal (**I**) or out (**O**) a named entity. A wrong decision may affect the following words and cause a whole entity to fail to be recognized. Non-greedy approaches may palliate these effects trying to maximize the consistency of the whole tag sequence via a Dynamic Programming, Viterbi algorithm, etc.
- NEC: The combination of the four binary classifiers obtains lower performance than any of them. Further combination schemes must be explored, as well as the use of multi-label AdaBoost algorithms instead of binary classifiers.
- Trigger words are linked to its EuroWordNet sense, so they can be expanded with its synonyms, hyponyms or hyperonyms, increasing their coverage as well as providing a valuable information for correference resolution tasks.
- Syntactically complex weak NEs: We plan to develop a corpus where the syntactically complex weak NEs will be annotated. This corpus will be used as an information source to build a grammar which will treat this kind of entities, as well as to evaluate system performance.
- Adaptation of MICE to other languages such as Catalan or English.

References

Aberdeen, J., J. Burger, D. Day, L. Hirschman, P. Robinson, and M. Vilain, 1995. MITRE: *Description of the ALEMBIC System Used for MUC-6*, pages 141–155. Proceedings of the 6th Messsage Understanding Conference. Morgan Kaufmann Publishers, Inc., Columbia, Maryland.

Abney, S., R.E. Schapire, and Y. Singer. 1999. Boosting Applied to Tagging and PP–attachment. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.*

Appelt, D.E., J.R. Hobbs, J. Bear, D. Israel, M. Kameyama, A. Kehler, D. Martin, K. Myers, and M. Tyson, 1995. SRI *International* FASTUS *System MUC-6 Test Results and Analysis*, pages 237–248. Proceedings of the 6th Messsage Understanding Conference. Morgan Kaufmann Publishers, Inc., Columbia, Maryland.

Arévalo, M. 2001. Gramática para la detección y clasificación de entidades con nombre. D.E.A. Report, Departament de Lingüística, Universitat de Barcelona, Barcelona.

Arévalo, M. and N. Bufí. 2001. Manual de etiquetación semántica del corpus de la Agencia EFE. X-TRACT Working Paper, WP-01/06, Centre de Llenguatge i Computació, CLiC, Barcelona.

Bikel, D.M., S. Miller, R. Schwartz, and R. Weischedel. 1997. Nymble: A High Performance Learning Name-Finder. In *Proceedings of the 5th Conference on Applied Natural Language Processing, ANLP*, Washington DC. ACL.

Black, W.J., F. Rinaldi, and D. Mowatt. 1998. FACILE: Description of the NE System Used for MUC-7. In *Proceedings of the 7th Message Understanding Conference.*

Borthwick, A., J. Sterling, E. Agichtein, and R. Grishman. 1998. NYU: Description of the MENE Named Entity System as Used in MUC-7. In *Proceedings of the 7th Message Understanding Conference.*

Brill, E. 1992. A Simple Rule–Based Part–of–speech Tagger. In *Proceedings of the 3rd Conference on Applied Natural Language Processing, ANLP*, pages 152–155. ACL.

Carmona, J., 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 the 1st International Conference*

on Language Resources and Evaluation, LREC, pages 915–922, Granada, Spain.

Carreras, X. and L. Màrquez. 2001. Boosting Trees for Clause Splitting. In *Proceedings of the 5th Conference on Computational Natural Language Learning, CoNLL'01*, Tolouse, France.

Escudero, G., L. Màrquez, and G. Rigau. 2000. Boosting Applied to Word Sense Disambiguation. In *Proceedings of the 12th European Conference on Machine Learning, ECML*, Barcelona, Spain.

Freund, Y. and R.E. Schapire. 1997. A Decision–Theoretic Generalization of On–line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

Haruno, M., S. Shirai, and Y. Ooyama. 1999. Using Decision Trees to Construct a Practical Parser. *Machine Learning*, 34(1/2/3):131–151.

Krupka, G.R. and K. Hausman. 1998. IsoQuest, Inc.: Description of the NetOwl^{TM} Extractor System as Used for MUC-7. In *Proceedings of the 7th Message Understanding Conference.*

McDonald, D., 1996. *Internal and External Evidence in the Identification and Semantic Categorization of Proper Names*, pages 21–39. Corpus Processing for Lexical Acquisition (Boguraev and Pustejovsky, eds.). The MIT Press, Massachussets, MA.

Mikheev, A., C. Grover, and M. Moens. 1998. Description of the LTG System Used for MUC-7. In *Proceedings of the 7th Message Understanding Conference.*

Nirenburg, S. and V. Raskin. 1996. Ten Choices for Lexical Semantics. CRL MCCS-96-304, New Mexico State University, Las Cruces, NM.

Punyakanok, V. and D. Roth. 2000. The Use of Classifiers in Sequential Inference. In *Proceedings of the 13th Conference on Neural Information Processing Systems, NIPS'00.*

Ramshaw, L. and M. Marcus. 1995. Text Chunking using Tranformation–based Learning. In *Proceedings of the 3rd ACL Workshop on Very Large Corpora*, Cambridge, MA, USA.

Schapire, R.E. 2001. The boosting approach to machine learning. an overview. In *Proceedings of the MSRI Workshop on Nonlinear Estimation and Classification*, Berkeley, CA.

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

Schapire, R.E. and Y. Singer. 2000. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

Simón, M.J. 2001. Gazetteer para la categorización de entidades con nombre. D.E.A. Report, Departament de Lingüística, Universitat de Barcelona, Barcelona.

Weischedel, R., 1995. BBN*: Description of the* Plum *System as Used for MUC-6*, pages 55–69. Proceedings of the 6th Message Understanding Conference. Morgan Kaufmann Publishers, Inc., Columbia, Maryland.

Yu, S., S. Bai, and P. Wu. 1998. Description of the Kent Ridge Digital Labs System Used for MUC-7. In *Proceedings of the 7th Message Understanding Conference.*