1 Introduction

This paper presents a Named Entity Extraction (NEE) system for the CoNLL-2003 shared task competition. As in the past year edition (Carreras et al., 2002a), we have approached the task by treating the two main sub–tasks of the problem, recognition (NER) and classification (NEC), sequentially and independently with separate modules. Both modules are machine learning based systems, which make use of binary and multiclass AdaBoost classifiers.

Named Entity recognition is performed as a greedy sequence tagging procedure under the well–known BIO labeling scheme. This tagging process makes use of three binary classifiers trained to be experts on the recognition of B, I, and O labels, respectively. Named Entity classification is viewed as a 4–class classification problem (with LOC, PER, ORG, and MISC class labels), which is straightforwardly addressed by the use of a multiclass learning algorithm.

The system presented here consists of a replication, with some minor changes, of the system that obtained the best results in the CoNLL-2002 NEE task. Therefore, it can be considered as a benchmark of the state–of–the–art technology for the current edition, and will allow also to make comparisons about the training corpora of both editions.

2 Learning the Decisions

We use AdaBoost with confidence rated predictions as learning algorithm for the classifiers involved in the system. More particularly, the basic binary version has been used to learn the I, O, and B classifiers for the NER module, and the multiclass multilabel extension (namely AdaBoost.MH) has been used to perform entity classification.

The idea of these algorithms is to learn an accurate strong classifier by linearly combining, in a weighted voting scheme, many simple and moderately–accurate base classifiers or rules. Each base rule is learned sequentially by presenting the base learning algorithm a weighting over the examples, which is dynamically adjusted depending on the behavior of the previously learned rules.

AdaBoost has been applied, with significant success, to a number of problems in different areas, including NLP tasks (Schapire, 2002). We refer the reader to (Schapire and Singer, 1999) for details about the general algorithms (for both the binary and multiclass variants), and (Carreras and Márquez, 2001; Carreras et al., 2002b) for particular applications to NLP domains.

In our setting, the boosting algorithm combines several small fixed–depth decision trees, as base rules. Each branch of a tree is, in fact, a conjunction of binary features, allowing the strong boosting classifier to work with complex and expressive rules.

3 Feature Representation

A window $W$ anchored in a word $w$ represents the local context of $w$ used by a classifier to make a decision on that word. In the window, each word around $w$ is codified with a set of primitive features, together with its relative position to $w$. Each primitive feature with each relative position and each possible value forms a final binary feature for the classifier (e.g., “the word form at position(-2) is street”). The kind of information coded in those features may be grouped in the following kinds:

- **Lexical**: Word forms and their position in the window (e.g., $W(3)=\text{“bank”}$). When available, word lemmas and their position in the window.
- **Syntactic**: Part-of-Speech tags and Chunk tags.
- **Orthographic**: Word properties with regard to how is it capitalized (initial-caps, all-caps), the kind of characters that form the word (contains-digits, all-digits, alphanumeric, roman-number), the presence of punctuation marks (contains-dots, contains-hyphen, acronym), single character patterns (lonely-
initial, punctuation-mark, single-char), or the membership of the word to a predefined class (functional-word\(^1\)), or pattern (URL).

- **Affixes**: The prefixes and suffixes of the word (up to 4 characters).
- **Word Type Patterns**: Type pattern of consecutive words in the sentence. The type of a word is either functional (\(f\)), capitalized (\(c\)), lowercased (\(l\)), punctuation mark (\(p\)), quote (\(q\)) or other (\(x\)). For instance, the word type pattern for the phrase “John Smith payed 3 euros” would be \(cclxl\).
- **Left Predictions**: The \(\{B,I,O\}\) tags being predicted in the current classification (at recognition stage), or the predicted category for entities in left context (at classification stage).
- **Bag-of-Words**: Form of the words in the window, without considering positions (e.g., “bank” \(\in W\)).
- **Trigger Words**: Triggering properties of window words. An external list is used to determine whether a word may trigger a certain Named Entity (NE) class (e.g., “president” may trigger class PER).
- **Gazetteer Features**: Gazetteer information for window words. An external gazetteer is used to determine possible classes for each word.

4 The NER Module

The Named Entity recognition task is performed as a combination of local classifiers which test simple decisions on each word in the text.

According to a BIO labelling scheme, each word is tagged as either the beginning of a NE (B tag), a word inside a NE (I tag), or a word outside a NE (O tag). We use three binary classifiers for the tagging, one corresponding to each tag. All the words in the train set are used as training examples, applying a one-vs-all binarization. When tagging, the sentence is processed from left to right, greedily selecting for each word the tag with maximum confidence that is coherent with the current solution (e.g., O tags cannot be followed by I tags). Despite its simplicity, the greedy BIO tagging performed very well for the NER task. Other more sophisticated representations and tagging schemes, studied in the past edition (Carreras et al., 2002a), did not improve the performance at all.

The three classifiers use the same information to codify examples. According to the information types introduced in section 3, all the following features are considered for each target word: **lexical, syntactic, orthographic**, and **affixes** in a \{-3,+3\} window; **left predictions** in a \{-3,-1\} window; and all the **word type patterns** that cover the 0 position in a \{-3,+3\} window.

The semantic information represented by the rest of features, namely **bag-of-words, trigger words, and gazetteer features**, did not help the recognition of Named Entities, and therefore was not used.

5 The NEC Module

NEC is regarded as a classification task, consisting of assigning the NE type to each already recognized NE. In contrast to the last year system, the problem has not been binarized and treated in an ECOC (error correcting output codes) combination scheme. Instead, the multiclass multilabel AdaBoost.MH algorithm has been used. The reason is that although ECOC provides slightly better results, its computational cost is also much higher than the required for AdaBoost.MH.

The algorithm has been employed with different parameterizations, by modeling NEC either as a three-class classification problem (in which MISC is selected only when the entity is negatively classified as PER, ORG and LOC) or as a four-class problem, in which MISC is just one more class. The latter turned out to be the best choice (with very significant differences).

The window information described in section 3 is used in the NEC module computing all features for a \{-3,+3\} window around the NE being classified, except for the **bag-of-words** group, for which a \{-5,+5\} window is used. Information relative to **orthographic**, **left predictions**, and **bag-of-words** features is straightforwardly coded as described above, but other requires further detail:

- **Lexical features**: Apart from word form and lemma for each window position, two additional binary features are used: One is satisfied when the focus NE form and lemma coincide exactly, and the other when they coincide after turning both of them into lowercase.
- **Syntactic features**: Part-of-Speech (PoS) and Chunk tags of window words (e.g., \(W(3)\).PoS=NN). PoS and Chunk pattern of the NE (e.g., NNPS_POS JJ for the NE “People’s Daily”)
- **Affix features**: Prefixes and suffixes of all window words. Prefixes and suffixes of the NE being classified and of its internal components (e.g., considering the entity “People’s Daily”, “ly” is taken as a suffix of the NE, “ple” is taken as a suffix of the first internal word, etc.).
- **Trigger Words**: Triggering properties of window words (e.g., \(W(3)\).trig=PER). Triggering properties of components of the NE being classified (e.g., for...
ments are not identical: The German corpus enables the
in the shared task, though German and English environ-
7 Results
by the organization was not used in this work.
The set of unlabelled examples provided
the number of rounds were directly optimized on the de-
decision stumps) to 4. The depth of the base rules and
decision trees per classifier, with depths ranging from 1
(sterm words inside a NE that appeared more than 3 times were
Additionally, binary features encoding the length in
words of the NE being classified are also used.
6 Experimental Setting
The list of functional words for the task has been automatic-
words inside a NE that appeared more than 3 times were
selected as functional words for the language.
Similarly, a gazetteer was constructed with the NEs in
the training set. When training, only a random 40% of the
entries in the gazetteer were considered. Moreover, we
used external knowledge in the form of a list of trigger
words for NEs and an external gazetteer. These knowl-
edge sources are the same that we used in the last year
competition for Spanish NEE. The entries of the trigger–
word list were linked to the Spanish WordNet, so they
have been directly translated by picking the corresponding
synsets of the English WordNet. The gazetteer has
been left unchanged, assuming interlinguality of most of
the entries. The gazetteer provided by the CoNLL-2003
organization has not been used in the work reported in
this paper.
In all cases, a preprocess of attribute filtering was per-
formed in order to avoid overfitting and to speed–up
learning. All features that occur less than 3 times in the
training corpus were discarded.
For each classification problem we trained the corre-
sponding AdaBoost classifiers, learning up to 4,000 base
decision trees per classifier, with depths ranging from 1
(decision stumps) to 4. The depth of the base rules and
the number of rounds were directly optimized on the de-
development set. The set of unlabelled examples provided
by the organization was not used in this work.
7 Results
The described system has been applied to both languages
in the shared task, though German and English environ-
ments are not identical: The German corpus enables the
use of lemma features while English does not. Also, the
used trigger word list is available for English but not for
German.
The results of the BIO model for the NER task on
the development and test sets for English and German
are presented in table 1. As will be seen later for the
whole task, the results are systematically better for En-
lish than for German. As it can be observed, the
behaviour on the development and test English sets is quite
different. While in the development set the NER mod-
ule achieves a very good balance between precision and
recall, in the test set the precision drops almost 4 points,
being the $F_1$ results much worse. On the contrary, de-
development and test sets for German are much more sim-
ilar. In this case, recall levels obtained for the language
are much lower compared to precision ones. This fact is
indicating the difficulty for reliably detecting the begin-
ings of the Named Entities in German (all common and
proper nouns are capitalized). Probably, a non–greedy
tagging procedure would have the chance to improve the
recognition results.

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>$F_{\beta=1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>English devel.</td>
<td>95.65%</td>
<td>95.51%</td>
<td>95.58</td>
</tr>
<tr>
<td>English test</td>
<td>91.93%</td>
<td>94.02%</td>
<td>92.96</td>
</tr>
<tr>
<td>German devel.</td>
<td>88.15%</td>
<td>71.55%</td>
<td>78.99</td>
</tr>
<tr>
<td>German test</td>
<td>85.87%</td>
<td>72.61%</td>
<td>78.68</td>
</tr>
</tbody>
</table>

Table 1: Results of the BIO recognizer for the NER task

Regarding NEC task, optimal feature selection is dif-
ferent for each language: Chunk information is almost
useless in English (or even harmful, when combined with
PoS features), but useful in German. On the contrary, al-
though the use of left predictions for NEC is useful for
English, the lower accuracy of the German system ren-
ders those features harmful (they are very useful when
assumed perfect left predictions). Table 2 presents NEC
accuracy results assuming perfect recognition of entities.

<table>
<thead>
<tr>
<th></th>
<th>English</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>features</td>
<td>accuracy</td>
<td>features</td>
</tr>
<tr>
<td>basic</td>
<td>91.47%</td>
<td>basic</td>
</tr>
<tr>
<td>basic+P</td>
<td>92.14%</td>
<td>basic+P</td>
</tr>
<tr>
<td>basic+C</td>
<td>91.60%</td>
<td>basic+C</td>
</tr>
<tr>
<td>basic+PC</td>
<td>92.12%</td>
<td>basic+PC</td>
</tr>
<tr>
<td>basic+Pg</td>
<td>93.86%</td>
<td>basic+Pg</td>
</tr>
<tr>
<td>basic+PG</td>
<td>95.05%</td>
<td>basic+PG</td>
</tr>
<tr>
<td>basic+PGT</td>
<td>95.14%</td>
<td>basic+PGT</td>
</tr>
</tbody>
</table>

Table 2: NEC accuracy on the development set assuming
a perfect recognition of named entities

The basic feature set includes all lexical, orthographic,
affix and bag–of–words information. P stands for Part-of-
Speech features, C for chunking-related information, T for trigger-words features and g/G for gazetteer-related information. In general, more complex features sets yield better results, except for the C case in English, as commented above.

Table 4 presents the results on the NEE task obtained by pipelining the NER and NEC modules. The NEC module used both knowledge extracted from the training set as well as external sources such as the gazetteer or trigger word lists.

Almost the same conclusions extracted from the NER results apply to the complete task, although here the results are lower due to the cascade of errors introduced by the two modules: 1) Results on English are definitely better than on German; 2) Development and test sets present a regular behaviour in German, while for English they are significantly different. We find the latter particularly disappointing because it is indicating that no reliable conclusions can be extracted about the generalization error of the NEE system constructed, by testing it on a 3,000 sentence corpus. This may be caused by the fact that the training set is no representative enough, or by a too biased learning of the NEE system towards the development set.

Regarding particular categories, we can see that for English the results are not extremely dissimilar (F1 values fall in a range of 10 points for each set), being LOC and PER the most easy to identify and ORG and MISC the most difficult. Comparatively, in the German case bigger differences are observed (F1 ranges from 52.58 to 80.79 in the test set), e.g., recognition of MISC entities is far worse than the rest. Another slight difference against English is that the easiest category is PER instead of LOC.

In order to allow fair comparison with other systems, table 3 presents the results achieved on the development set without using external knowledge. The features used correspond to the basic model plus Part-of-Speech information (plus Chunks for German), plus a gazetteer build with the entities appearing in the training corpus.

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>English devel.</td>
<td>90.34%</td>
<td>90.21%</td>
<td>90.27%</td>
</tr>
<tr>
<td>English test</td>
<td>83.19%</td>
<td>85.07%</td>
<td>84.12%</td>
</tr>
<tr>
<td>German devel.</td>
<td>74.87%</td>
<td>60.77%</td>
<td>67.09%</td>
</tr>
<tr>
<td>German test</td>
<td>74.69%</td>
<td>63.16%</td>
<td>68.45%</td>
</tr>
</tbody>
</table>

Table 3: Overall results using no external knowledge

**Acknowledgments**

This research has been partially funded by the European Commission (Meaning, IST-2001-34460) and the Spanish Research Dept. (Hermes, TIC2000-0335-C03-02; Pe-

---

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>English devel.</td>
<td>95.33%</td>
<td>94.39%</td>
<td>94.86%</td>
</tr>
<tr>
<td>MISC</td>
<td>89.94%</td>
<td>83.41%</td>
<td>86.55%</td>
</tr>
<tr>
<td>ORG</td>
<td>86.98%</td>
<td>88.14%</td>
<td>87.56%</td>
</tr>
<tr>
<td>PER</td>
<td>91.79%</td>
<td>94.68%</td>
<td>93.21%</td>
</tr>
<tr>
<td>Overall</td>
<td>91.51%</td>
<td>91.37%</td>
<td>91.44%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOC</td>
<td>88.14%</td>
<td>90.41%</td>
<td>89.26%</td>
</tr>
<tr>
<td>MISC</td>
<td>82.02%</td>
<td>75.36%</td>
<td>78.54%</td>
</tr>
<tr>
<td>ORG</td>
<td>78.40%</td>
<td>80.43%</td>
<td>79.41%</td>
</tr>
<tr>
<td>PER</td>
<td>86.36%</td>
<td>91.65%</td>
<td>88.93%</td>
</tr>
<tr>
<td>Overall</td>
<td>84.05%</td>
<td>85.96%</td>
<td>85.00%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOC</td>
<td>75.72%</td>
<td>73.67%</td>
<td>74.68%</td>
</tr>
<tr>
<td>MISC</td>
<td>72.34%</td>
<td>42.48%</td>
<td>53.52%</td>
</tr>
<tr>
<td>ORG</td>
<td>76.89%</td>
<td>63.82%</td>
<td>69.75%</td>
</tr>
<tr>
<td>PER</td>
<td>83.84%</td>
<td>68.88%</td>
<td>75.63%</td>
</tr>
<tr>
<td>Overall</td>
<td>77.90%</td>
<td>63.23%</td>
<td>69.80%</td>
</tr>
</tbody>
</table>

Table 4: Final results for English and German

---


References