Boosting for Named Entity Recognition

Dekai Wu[†] dekai@cs.ust.hk
Grace Ngai[‡] grace@intendi.com
Marine Carpuat[†] eemarine@ust.hk
Jeppe Larsen[†] egljo@ust.hk
Yongsheng Yang[†] ysyang@cs.ust.hk

[†] Human Language Technology Center
HKUST
Clear Water Bay
Hong Kong

[‡] Intendi Inc.
Hong Kong

Abstract

This paper presents a system that applies boosting to the task of named-entity identification. The CoNLL-2002 shared task, for which the system is designed, is language-independent named-entity recognition. Using a set of features which are easily obtainable for almost any language, the presented system uses boosting to combine a set of weak classifiers into a final system that performs significantly better than that of an off-the-shelf maximum entropy classifier.

1 Introduction

Named entity recognition (NER) has emerged as an important step for many natural language processing applications, such as question answering and information extraction. As a result, systems have been developed that give impressive results but they are usually limited to performing NER for a specific domain and only on one language. This allows the systems to be tailored for the specific task by using knowledge about a particular language.

This paper presents a system that introduces boosting techniques to the identification and classification of named entities, achieving significantly higher performance than our strongest baseline system employing an off-the-shelf maximum entropy model. The system was designed for the CoNLL-2002 shared task competition; the goal of which was to perform named-entity recognition on four types of named entities, PERSON, LOCATION, ORGANIZATION and MISCELLANEOUS. In addition, the task specifications were that two European languages would be involved, but one of them would not be specified until after the submission deadline.

The requirement of the system, given the goal of the competition, was then to achieve a high performance without relying too heavily on knowledge that is very specific for a particular language or domain. In that spirit, the system described avoids using language-dependent knowledge and instead relies on a number of features which are easily obtainable for any language. The approach is based on selecting a number of language independent features, which are used to train several weak classifiers. Using boosting, which has shown to perform well on other NLP problems and is a theoretically proven method, the weak classifiers are then combined to perform an accurate classifier. The final performance achieved is significantly better than that of an off-the-shelf maximum entropy tagger.

2 Boosting

The main idea behind boosting algorithms is that a combination of many simple and moderately accurate weak classifiers will yield a single and highly accurate classifier. Each weak classifier is trained sequentially and the idea is that the weak classifier in the current iteration will be forced, by the weak classifiers in the previous iterations, to train on instances that they found to be hard to classify.

This paper presents a system that uses the familiar AdaBoost algorithm (Freund and Schapire, 1997) as the boosting framework. AdaBoost has shown its usefulness on standard machine-learning tasks through extensive theoretical and empirical study, where different standard machine-learning methods have been used as the weak classifier (Schapire, 2002; Bauer and Kohavi, 1999; Opitz and Maclin, 1999).

The original AdaBoost algorithm was designed for binary classification problems, which does not fulfill the requirements of the NER task. Therefore, the actual algorithm used by the presented system is the AdaBoost.MH algorithm (Schapire and Singer, 1999) which is a generalization of the original AdaBoost algorithm for multiclass classification. AdaBoost.MH has also been shown to perform well on various machine-learning problems and in particular it has performed well on numerous natural language problems. Furthermore, AdaBoost.MH has been successfully applied to Text Categorization (Schapire and Singer, 2000) and for Word Sense Disambiguation (Escudero et al., 2000).

The weak classifiers utilized in the boosting procedure can come from a wide range of machine learning methods. Previously-used methods have included decision trees and neural nets. For our application of boosting we have chosen to utilize a simple classifier called a *Decision Stump*. A decision stump is basically a one-level decision tree where the split at the root level is based on a specific attribute/value pair. For example a possible attribute/value pair could be: $W_{-2} = "Taiwan"$.

It has been suggested that WSD and Text Categorization are problems that have several similar properties (Escudero et al., 2000). On reflection, the NER task also has similar properties:

- High number of features many of which are tested for presence or absence.
- Many irrelevant and highly interdependent features.
- The learned concepts and the examples are sparse in the feature space.

Since AdaBoost.MH has been shown to perform well on the two aforementioned problems, it seems reasonable to hypothesize that AdaBoost.MH would be well suited to NER.

3 Experiment Details

To our knowledge, this is the first attempt at using boosting to solve the named-entity recognition problem.

The experiments detailed in this section are all performed with the publicly available AT&T BoosTexter software (Schapire and Singer, 2000), which implements boosting on top of decision stumps. In addition, a publicly available maximum entropy part-of-speech tagger (Ratnaparkhi, 1996) was used to provide a reasonable baseline for the task.

3.1 Preprocessing of the Data

In order to get more information for our decision stumps, more preprocessing had to be performed on the data. For both languages, publicly available part-of-speech taggers were used to obtain the corresponding POS tags for the words. The Spanish data was tagged with the CRATER tagger (Sanchez, 1995) which assigned a set of 475 tags to the words; while the Dutch data was run through MBT, the online memory-based tagger demo (Daelemans et al., 1996). (Coarse part-of-speech tags had been provided for the Dutch data; however, it should be noted that the disagreement rate between the result of the online demo and the original tags is almost 10%.)

As Spanish and Dutch are both highly inflected languages, an approximation at morphological analysis was also made during the preprocessing step. Prefixes and suffixes of lengths of up to 4 characters were extracted automatically from the words and added to the feature set.

For the Spanish data, an additional source of information was available to us in the form of lemmas, which were automatically extracted via a language-independent lemmatizer (Yarowsky and Wicentowski, 2000). This information, however, was not available for the Dutch since the lemmas were not obtained in time for the final training.

It should be noted that, in the spirit of language-independence, these preprocessing steps can be relatively easily performed for almost any language in which enough language analysis has been performed. In the case of Asian languages such as Chinese or Japanese, the lemmatization can be replaced by a decomposition of words at the character level.

3.2 Feature set

For the maximum entropy model, since the only software available was the part-of-speech tagger, it was not possible to provide it with any information other than the words and the chunk tags. For any given word, the default features used by the maxent tagger were words within a window of 2, the previous two tags, the prefixes and suffixes, and some extra information indicating whether the word is a number, is capitalized, or contains a hyphen.

The boosting/decision stumps were able to accommodate a large number of features. The features used in the final experiments were:

1. Lexical (words and lemmas) and syntactic (part-of-speech) information within a window of 2 words surrounding the current word;
2. Prefixes and suffixes of up to a length of 4 characters from the current word;
3. Capitalization: whether the word starts with a capital letter and/or the entire word is capitalized;
4. A small set of conjunctions of POS tags and words within a window of 2 words of the current word;
5. Previous History: the chunk tags (gold standard during training; assigned for evaluation) of the previous two words.

During the course of the experimentation, an analysis of the errors showed that there existed a number of multi-word named entities which the model was unable to identify correctly, seemingly only because the length of the entity posed a problem for the window size that our model was allowed to consider. To tackle this problem, a lexicon consisting of all named-entities which were longer than 3 words and had a consistent type (i.e. has only one named-entity tag) was compiled from the training corpus. In addition, a handful of named-entities which corresponded to names of professional sports clubs and countries were added to the list. At evaluation time, if these named entities appeared in the corpus, they were tagged with the corresponding chunk tags by default.

3.3 Results

Table 1 presents the results obtained on the development and test sets for both Spanish and Dutch for the boosting/decision stumps. As a comparison, Table 2 shows the results of the maximum entropy tagger on the evaluation set for Dutch and Spanish. Both systems were trained on the given training set only; with data analysis and parameter tuning performed on the development set; and evaluation on the test set performed just prior to submission.

A comparison between the two tables shows clearly that the boosting model outperforms the maximum entropy model significantly on all named-entity types. One could make an argument that the boosting model was provided with part-of-speech tags — information that the maximum entropy tagger did not have access to; however, the maximum entropy tagger achieved the best performance among all the off-the-shelf systems that were tried, and can therefore be reasonably considered to be the strongest available competitor to the boosting model.

4 Conclusion

This paper presented a system that applied boosting and decision stumps to the task of named entity recognition. BoosTexter, a publicly-available off-the-shelf toolkit, is used to construct a resulting system that achieves a performance of 76.61 F-Measure for the Spanish test set and a 75.36 F-Measure for the Dutch test set for the CoNLL-2002 shared task competition. The system was not tailored with any language-specific knowledge, and was only provided with information that would be easily obtainable for almost any language. In addition, the results achieved represent a 13.9% and 22.8% error reduction over that of our strongest baseline system — a maximum entropy tagger — for Spanish and Dutch, respectively.

5 Acknowledgments

The work presented in this paper was funded by and is joint work between the Hong Kong University of Science and Technology and Weniwen Technologies.

References

E. Bauer and R. Kohavi. 1999. An empirical comparison of voting classification algorithms: Bagging, boosting and variants. In *Machine Learning, 36*, pages 105–142.

W. Daelemans, J. Zavrel, and S. Berck. 1996. MBT: A memory-based part of speech tagger-generator.

G. Escudero, L. Marquez, and G. Rigau. 2000. Boosting applied to word sense disambiguation. In *European Conference on Machine Learning*, pages 129–141.

Y. Freund and R. E. Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. In *Journal of Computer and System Sciences, 55(1)*, pages 119–139.

D. Opitz and R. Maclin. 1999. Popular ensemble methods: An empirical study. In *Journal of Artificial Intelligence Researce, 11*, pages 169–198.

A. Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proceedings of the Empirical Methods in Natural Language Processing Conference*, Philadelphia, PA, May 17-18. ACL.

F. Sanchez. 1995. Development of a Spanish version of the Xerox tagger.

R. E. Schapire and Y. Singer. 1999. Improved boosting algorithms. using confidence-rated predictions. In *Machine Learning, 37*, pages 297–336.

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

R. E. Schapire. 2002. The boosting approach to machine learning. an overview. In *MSRI Workshop on Nonlinear Estimation and Classification.*

D. Yarowsky and R. Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of ACL-2000*, pages 207–216, Hong Kong.

Spanish Dev.	precision	recall	$F_{\beta=1}$
LOC	65.72%	77.26%	71.02
MISC	45.19%	45.39%	45.29
ORG	75.18%	72.53%	73.83
PER	84.50%	78.97%	81.64
overall	72.05%	72.63%	72.34

Spanish Test	precision	recall	$F_{\beta=1}$
LOC	79.15%	77.40%	78.26
MISC	55.76%	44.12%	49.26
ORG	74.73%	79.21%	76.91
PER	80.20%	89.25%	84.48
overall	75.85%	77.38%	76.61

Dutch Dev.	precision	recall	$F_{\beta=1}$
LOC	76.38%	75.42%	75.90
MISC	71.68%	71.58%	71.63
ORG	78.99%	55.38%	65.11
PER	68.47%	76.69%	72.35
overall	72.95%	69.45%	71.16

Dutch Test	precision	recall	$F_{\beta=1}$
LOC	81.88%	79.12%	80.47
MISC	78.12%	68.58%	73.04
ORG	73.86%	62.83%	67.90
PER	74.88%	84.51%	79.40
overall	76.95%	73.83%	75.36

Table 1: Boosting/Decision Stump results obtained for the development and the test data sets for the two languages used in this shared task.

Spanish Test	precision	recall	$F_{\beta=1}$
LOC	73.86%	73.25%	73.55
MISC	50.00%	42.06%	45.69
ORG	72.81%	76.50%	74.61
PER	74.94%	84.63%	79.49
overall	71.82%	73.90%	72.84

Dutch Test	precision	recall	$F_{\beta=1}$
LOC	74.47%	72.63%	73.54
MISC	73.99%	61.58%	67.22
ORG	67.67%	56.33%	61.48
PER	67.17%	72.83%	69.88
overall	70.60%	65.73%	68.08

Table 2: For comparison: Maximum Entropy results obtained for the evaluation data set for the two languages used in this shared task.