Boosting for Chinese Named Entity Recognition

Xiaofeng YU **Marine CARPUAT** **Dekai WU***
Human Language Technology Center
HKUST
Department of Computer Science and Engineering
University of Science and Technology
Clear Water Bay, Hong Kong
{xfyu,marine,dekai}@cs.ust.hk

Abstract

We report an experiment in which a high-performance boosting based NER model originally designed for multiple European languages is instead applied to the Chinese named entity recognition task of the third SIGHAN Chinese language processing bakeoff. Using a simple character-based model along with a set of features that are easily obtained from the Chinese input strings, the system described employs boosting, a promising and theoretically well-founded machine learning method to combine a set of weak classifiers together into a final system. Even though we did no other Chinese-specific tuning, and used only one-third of the MSRA and CityU corpora to train the system, reasonable results are obtained. Our evaluation results show that 75.07 and 80.51 overall F-measures were obtained on MSRA and CityU test sets respectively.

1 Introduction

Named entity recognition (NER), which includes the identification and classification of certain proper nouns, such as person names, organizations, locations, temporal, numerical and monetary phrases, plays an important part in many natural language processing applications, such as machine translation, information retrieval, information extraction and question answering. Much of the NER research was pioneered in the MUC/DUC and Multilingual Entity Task (MET) evaluations, as a result of which significant progress has been made and many NER systems of fairly high accuracy have been constructed. In addition, the shared tasks of CoNLL-2002 and CoNLL-2003 helped spur the development toward more language-independent NER systems, by evaluating four types of entities (people, locations, organizations and names of miscellaneous entities) in English, German, Dutch and Spanish.

However, these are all European languages, and Chinese NER appears to be significantly more challenging in a number of important respects. We believe some of the main reasons to be as follows: (1) Unlike European languages, Chinese lacks capitalization information which plays a very important role in identifying named entities. (2) There is no space between words in Chinese, so ambiguous segmentation interacts with NER decisions. Consequently, segmentation errors will affect the NER performance, and vice versa. (3) Unlike European languages, Chinese allows an open vocabulary for proper names of persons, eliminating another major source of explicit clues used by European language NER models.

This paper presents a system that introduces boosting to Chinese named entity identification and classification. Our primary aim was to conduct a controlled experiment to test how well the boosting based models we designed for European languages would fare on Chinese, *without* major modeling alterations to accommodate Chinese. We evaluated the system using data from the third SIGHAN Chinese language processing bakeoff, the goal of which was to perform NER on three types of named entities: PERSON, LOCATION and ORGANIZATION.[1] Three training corpora from MSRA, CityU and LDC were given. The MSRA and LDC corpora were simplified Chinese texts while the CityU corpus was traditional

*This work was supported in part by DARPA GALE contract HR0011-06-C-0023, and by the Hong Kong Research Grants Council (RGC) research grants RGC6083/99E, RGC6256/00E, and DAG03/04.EG09.

[1]Except in the LDC corpus, which contains four types of entities: PERSON, LOCATION, ORGANIZATION and GEOPOLITICAL.

Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing, pages 150–153,
Sydney, July 2006.

Chinese. In addition, the competition also specified open and closed tests. In the open test, the participants may use any other material including material from other training corpora, proprietary dictionaries, and material from the Web besides the given training corpora. In the closed test, the participants can only use the three training corpora. No other material or knowledge is allowed, including part-of-speech (POS) information, externally generated word-frequency counts, Arabic and Chinese numbers, feature characters for place names, common Chinese surnames, and so on.

The approach we used is based on selecting a number of features, which are used to train several weak classifiers. Using boosting, which has been shown to perform well on other NLP problems and is a theoretically well-founded method, the weak classifiers are then combined to perform a strong classifier.

2 Boosting

The main idea behind the boosting algorithm is that a set of many simple and moderately accurate weak classifiers (also called **weak hypotheses**) can be effectively combined to yield a single strong classifier (also called the **final hypothesis**). The algorithm works by training weak classifiers sequentially whose classification accuracy is slightly better than random guessing and finally combining them into a highly accurate classifier. Each weak classifier searches for the hypothesis in the hypotheses space that can best classify the current set of training examples. Based on the evaluation of each iteration, the algorithm reweights the training examples, forcing the newly generated weak classifier to give higher weights to the examples that are misclassified in the previous iteration. The boosting algorithm was originally created to deal with binary classification in supervised learning. The boosting algorithm is simple to implement, does feature selection resulting in a relatively simple classifier, and has fairly good generalization.

Based on the boosting framework, our system uses the AdaBoost.MH algorithm (Schapire and Singer, 1999) as shown in Figure 1, an n-ary classification variant of the original well-known binary AdaBoost algorithm (Freund and Schapire, 1997). The original AdaBoost algorithm was designed for the binary classification problem but did not fulfill the requirements of the Chinese NER task. AdaBoost.MH has shown its usefulness on standard machine learning tasks through extensive theoretical and empirical studies, where different standard machine learning methods have been used as the weak classifier (e.g., Bauer and Kohavi (1999), Opitz and Maclin (1999), Schapire (2002)). It also performs well on a number of natural language processing problems, including text categorization (e.g., Schapire and Singer (2000), Sebastiani *et al.* (2000)) and word sense disambiguation (e.g., Escudero *et al.* (2000)). In particular, it has also been demonstrated that boosting can be used to build language-independent NER models that perform exceptionally well (Wu *et al.* (2002), Wu *et al.* (2004), Carreras *et al.* (2002)).

Input: A training set $T_r = \{< d_1, C_1 >, \ldots, < d_g, C_g >\}$ where $C_j \subseteq C = \{c_1, ..., c_m\}$ for all $j = 1, \ldots, g$.
Output: A final hypothesis $\Phi(d, c) = \sum_{s=1}^{S} \alpha_s \Phi_s(d, c)$.
Algorithm: Let$D_1(d_j, c_i) = \frac{1}{mg}$ for all $j = 1, \ldots, g$ and for all $i = 1, \ldots, m$. For $s = 1, \ldots, S$ do:

- pass distribution $D_s(d_j, c_i)$to the weak classifier;
- derive the weak hypothesis Φ_s from the weak classifier;
- choose $\alpha_s \in R$;
- set $D_{s+1}(d_j, c_i) = \frac{D_s(d_j,c_i)\exp(-\alpha_s C_j[c_i]\Phi_s(d_j,c_i))}{Z_s}$ where $Z_s = \sum_{i=1}^{m}\sum_{j=1}^{g} D_s(d_j, c_i)\exp(-\alpha_s C_j[c_i]\Phi_s(d_j, c_i))$ is a normalization factor chosen so that $\sum_{i=1}^{m}\sum_{j=1}^{g} D_{s+1}(d_j, c_i) = 1$.

Figure 1: The AdaBoost.MH algorithm.

The weak classifiers used in the boosting algorithm come from a wide range of machine learning methods. We have chosen to use a simple classifier called a **decision stump** in the algorithm. 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 = 香港.

3 Experiment Details

In order to implement the boosting/decision stumps, we used the publicly available software AT&T BoosTexter (Schapire and Singer, 2000), which implements boosting on top of decision stumps. For preprocessing we used an off-the-shelf Chinese lexical analysis system, the open source ICTCLAS (Zhang *et al.*, 2003), to segment and POS tag the training and test corpora.

3.1 Data Preprocessing

The training corpora provided by the SIGHAN bakeoff organizers were in the CoNLL two column format, with one Chinese character per line and hand-annotated named entity chunks in the second column.

In order to provide basic features for training the decision stumps, the training corpora were segmented and POS tagged by ICTCLAS, which labels Chinese words using a set of 39 tags. This module employs a hierarchical hidden Markov model (HHMM) and provides word segmentation, POS tagging and unknown word recognition. It performs reasonably well, with segmentation precision recently evaluated at 97.58%.[2] The recall rate of unknown words using role tagging was over 90%.

We note that about 200 words in each training corpora remained untagged. For these words we simply assigned the most frequently occurring tags in each training corpora.

3.2 Feature Set

The boosting/decision stumps were able to accommodate a large number of features. The primitive features we used were:

- The current character and its POS tag.
- The characters within a window of 2 characters before and after the current character.
- The POS tags within a window of 2 characters before and after the current character.
- The chunk tags (gold standard named entity label during the training) of the previous two characters.

The chunk tag is the **BIO** representation, which was employed in the CoNLL-2002 and CoNLL-2003 evaluations. In this representation, each character is tagged as either the beginning of a named entity (**B** tag), a character inside a named entity (**I** tag), or a character outside a named entity (**O** tag).

When we used conjunction features, we found that they helped the NER performance significantly. The conjunction features used are basically conjunctions of 2 consecutive characters and 2 consecutive POS tags. We also found that a larger context window (3 characters instead of 2 before and after the current character) to be quite helpful to performance.

Apart from the training and test corpora, we considered the gazetteers from LDC which contain about 540K persons, 242K locations and 98K organization names. Named entities in the training corpora which appeared in the gazetteers were identified lexically or by using a maximum forward match algorithm. Once named entities have been identified, each character can then be annotated with an NE chunk tag. The boosting learner

[2] Results from the recent official evaluation in the national 973 project.

Table 1: Dev set results on MSRA and CityU.

	Precision	*Recall*	$F_{\beta=1}$
MSRA			
LOC	82.00%	85.93%	83.92
ORG	76.99%	61.44%	68.34
PER	89.33%	74.47%	81.22
Overall	82.62%	76.45%	79.41
CityU			
LOC	88.62%	81.69%	85.02
ORG	82.50%	66.44%	73.61
PER	84.05%	84.58%	84.31
Overall	86.46%	79.26%	82.71

Table 2: Test set results on MSRA, CityU, LDC.

	Precision	*Recall*	$F_{\beta=1}$
MSRA			
LOC	84.98%	80.94%	82.91
ORG	72.82%	57.78%	64.43
PER	82.89%	59.91%	69.55
Overall	81.95%	69.26%	75.07
CityU			
LOC	88.65%	83.58%	86.04
ORG	83.75%	57.25%	68.01
PER	86.11%	76.42%	80.98
Overall	86.92%	74.98%	80.51
LDC			
LOC	65.84%	76.51%	70.78
ORG	53.69%	39.52%	45.53
PER	80.29%	68.97%	74.20
Overall	67.20%	65.54%	66.36
LDC (w/GPE)			
GPE	0.00%	0.00%	0.00
LOC	1.94%	37.74%	3.70
ORG	53.69%	39.52%	45.53
PER	80.29%	68.97%	74.20
Overall	30.58%	29.82%	30.19

can view the NE chunk tag as an additional feature. Here we used binary gazetteer features. If the character was annotated with an NE chunk tag, its gazetteer feature was set to 1; otherwise it was set to 0. However we found that adding binary gazetteer features does not significantly help the performance when conjunction features were used. In fact, it actually hurt the performance slightly.

The features used in the final experiments were:

- The current character and its POS tag.
- The characters within a window of 3 characters before and after the current character.
- The POS tags within a window of 3 characters before and after the current character.
- A small set of conjunctions of POS tags and characters within a window of 3 characters of the current character.
- The BIO chunk tags of the previous 3 characters.

4 Results

Table 1 presents the results obtained on the MSRA and CityU development test set. Table 2 presents the results obtained on the MSRA, CityU and LDC test sets. These numbers greatly underrepresent what could be expected from the boosting model, since we only used one-third of MSRA and CityU training corpora due to limitations of the boosting software. Another problem for the LDC corpus was training/testing mismatch: we did not train any models at all with the LDC training corpus, which was the only training set annontated with geopolitical entities (GPE). Instead, for the LDC test set, we simply used the system trained on the MSRA corpus. Thus, when we consider the geopolitical entity (GPE), our low overall F-measure on the LDC test set cannot be interpreted meaningfully.[3] Even so, using only one-third of the training data, the results on the MSRA and CityU test sets are reasonable: 75.07 and 80.51 overall F-measures were obtained on the MSRA and CityU test sets, respectively.

5 Conclusion

We have described an experiment applying a boosting based NER model originally designed for multiple European languages instead to the Chinese named entity recognition task. Even though we only used one-third of the MSRA and CityU corpora to train the system, the model produced reasonable results, obtaining 75.07 and 80.51 overall F-measures on MSRA and CityU test sets respectively.

Having established this baseline for comparison against our multilingual European language boosting based NER models, our next step will be to incorporate Chinese-specific attributes into the model to compare with.

[3] Our LDC test result was scored twice by the organizer.

References

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

Xavier Carreras, Lluís Màrquez, and Lluís Padró. Named entity extraction using AdaBoost. In *Computational Natural Language Learning (CoNLL-2002), at COLING-2002*, pages 171–174, Taipei, Sep 2002.

Gerard Escudero, Lluís Màrquez, and German Rigau. Boosting applied to word sense disambiguation. In *11th European Conference on Machine Learning (ECML-00)*, pages 129–141, 2000.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Computer and System Sciences*, 55(1):119–139, 1997.

David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.

Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2-3):135–168, 2000.

Robert E. Schapire. The boosting approach to machine learning: An overview. In *MSRI workshop on Nonlinear Estimation and Classification*, 2002.

Fabrizio Sebastiani, Alessandro Sperduti, and Nicola Valdambrini. An improved boosting algorithm and its application to automated text categorization. In *Proceedings of 9th ACM International Conference on Information and Knowledge Management*, pages 78–85, 2000.

Dekai Wu, Grace Ngai, Marine Carpuat, Jeppe Larsen, and Yongsheng Yang. Boosting for named entity recognition. In *Computational Natural Language Learning (CoNLL-2002), at COLING-2002*, pages 195–198, Taipei, Sep 2002.

Dekai Wu, Grace Ngai, and Marine Carpuat. Why nitpicking works: Evidence for Occam's razor in error correctors. In *20th International Conference on Computational Linguistics (COLING-2004)*, Geneva, 2004.

Hua Ping Zhang, Qun Liu, Xue-Qi Cheng, Hao Zhang, and Hong Kui Yu. Chinese lexical analysis using Hierarchical Hidden Markov Model. In *Proceedings of the second SIGHAN workshop on Chinese language processing*, volume 17, pages 63–70, 2003.