Chinese Named Entity Recognition with Cascaded Hybrid Model

Xiaofeng Yu

Information Systems Laboratory
Department of Systems Engineering & Engineering Management
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
xfyu@se.cuhk.edu.hk

Abstract

We propose a high-performance cascaded hybrid model for Chinese NER. Firstly, we use Boosting, a standard and theoretically well-founded machine learning method to combine a set of weak classifiers together into a base system. Secondly, we introduce various types of heuristic human knowledge into Markov Logic Networks (MLNs), an effective combination of first-order logic and probabilistic graphical models to validate Boosting NER hypotheses. Experimental results show that the cascaded hybrid model significantly outperforms the state-of-the-art Boosting model.

1 Introduction

Named entity recognition (NER) involves the identification and classification of certain proper nouns in text, such as person names (PERs), locations (LOCs), organizations (ORGs), miscellaneous names (MISCs), temporal, numerical and monetary phrases. It is a well-established task in the NLP community and is regarded as crucial technology for many NLP applications, such as information extraction, question answering, information retrieval and machine translation.

Compared to European-language NER, Chinese NER seems to be more difficult (Yu *et al.*, 2006). Recent approaches to Chinese NER are a shift away from manually constructed rules or finite state patterns towards machine learning or statistical methods. However, rule-based NER systems lack robustness and portability, and machine learning approaches might be unsatisfactory to learn linguistic information in Chinese NEs. In fact, Chinese NEs have distinct linguistic characteristics in their composition and human beings usually use prior knowledge to recognize NEs. For example, about 365 of the highest frequently used surnames cover 99% Chinese surnames (Sun *et al.*, 1995). For the LOC "北京市/Beijing City", "北京/Beijing" is the name part and "市/City" is the salient word. For the ORG "北京市政府/Beijing City Government", "北京/Beijing" is the LOC name part, "市/City" is the LOC salient word and "政府/Government" is the ORG salient word. Some ORGs contain one or more PERs, LOCs, MISCs and ORGs. A more complex example is the nested ORG "湖北省武汉市武汉大学计算机学院/School of Computer Science, Wuhan University, Wuhan City, Hubei Province" which contains two ORGs "武汉大学/Wuhan University" and "计算机学院/School of Computer Science" and two LOCs "湖北省/Hubei Province" and "武汉市/Wuhan City". The two ORGs contain ORG salient words "大学/University" and "学院/School", while the two LOCs contain LOC salient words "省/Province" and "市/City" respectively.

Inspired by the above observation, we propose a cascaded hybrid model for Chinese NER [1]. First, we employ Boosting, which has theoretical justification and has been shown to perform well on other NLP problems, to combine weak classifiers into a strong classifier. We then exploit a variety of heuristic human knowledge into MLNs, a powerful combination of logic and probability, to validate Boosting NER hypotheses. We also use three Markov chain Monte Carlo (MCMC) algorithms for probabilistic inference in MLNs. Experimental results show that the cascaded hybrid model yields better NER results than the stand-alone Boosting model by a large margin.

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 combin-

[1] In this paper we only focus on PERs, LOCs, ORGs and MISCs. Since temporal, numerical and monetary phrases can be well identified with rule-based approaches.

Proceedings of NAACL HLT 2007, Companion Volume, pages 197–200,
Rochester, NY, April 2007.

ing 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 re-weights the training examples, forcing the newly generated weak classifier to give higher weights to the examples that are misclassified in the previous iteration.

We use 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). It has 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)). In particular, reasonable Chinese NER results were still obtained using Boosting, even though there was no Chinese-specific tuning and the model was only trained on one-third of the provided corpora in SIGHAN bakeoff-3 (Yu *et al.*, 2006).

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.

3 Markov Logic Networks

A Markov Network (also known as Markov Random Field) is a model for the joint distribution of a set of variables (Pearl, 1988). It is composed of an undirected graph and a set of potential functions. A First-Order Knowledge Base (KB) (Genesereth and Nislsson, 1987) is a set of sentences or formulas in first-order logic. A Markov Logic Network (MLN) (Richardson and Domingos, 2006) is a KB with a weight attached to each formula (or clause). Together with a set of constants representing objects in the domain, it species a ground Markov Network containing one feature for each possible grounding of a first-order formula in the KB, with the corresponding weight. The weights associated with the formulas in an MLN jointly determine the probabilities of those formulas (and vice versa) via a *log-linear model*. An MLN defines a probability distribution over Herbrand interpretations (possible worlds), and can be thought of as a *template* for constructing Markov Networks. The probability distribution over possible worlds x specified by the ground Markov Network $M_{L,C}$ is given by

$$P(X = x) = \frac{1}{Z}\exp(\sum w_i n_i(x)) = \frac{1}{Z}\prod \phi_i\left(x_{\{i\}}\right)^{n_i(x)} \tag{1}$$

where F_i is the formula in first-order logic, w_i is a real number, $n_i(x)$ is the number of true groundings of F_i in x, $x_{\{i\}}$ is the true value of the atoms appearing in F_i, and $\phi_i\left(x_{\{i\}}\right) = e^{w_i}$.

3.1 Learning Weights

Given a relational database, MLN weights can in principle be learned generatively by maximizing the likelihood of this database. The gradient of the log-likelihood with respect to the weights is

$$\frac{\partial}{\partial w_i}\log P_w(X = x) = n_i(x) - \sum P_w(X = x')n_i(x') \tag{2}$$

where the sum is over all possible databases x', and $P_w(X = x')$ is $P(X = x')$ computed using the current weight vector $w = (w_1, ..., w_i, ...)$. Unfortunately, computing these expectations can be very expensive. Instead, we can maximize the *pseudo-likelihood* of the data more efficiently. If x is a possible database and x_l is the lth ground atom's truth value, the *pseudo-log-likelihood* of x given weights w is

$$\log P_w^*(X = x) = \sum_{l=1}^{n} \log P_w(X_{l=x_l} \mid MB_x(X_l)) \tag{3}$$

where $MB_x(X_l)$ is the state of X_l's Markov blanket in the data. Computing Equation 3 and its gradient does not require inference over the model, and is therefore much faster. We optimize the *pseudo-log-likelihood* using the limited-memory BFGS algorithm (Liu and Nocedal, 1989).

3.2 Inference

If F_1 and F_2 are two formulas in first-order logic, C is a finite set of constants including any constants that appear in F_1 or F_2, and L is an MLN, then

$$\begin{aligned} P \quad (F_1 \mid F_2, L, C) &= P(F_1 \mid F_2, M_{L,C}) && (4)\\ &= \frac{P(F_1 \wedge F_2 \mid M_{L,C})}{P(F_2 \mid M_{L,C})} && (5)\\ &= \frac{\sum_{x \in \chi_{F_1} \cap \chi_{F_2}} P(X = x \mid M_{L,C})}{\sum_{x \in \chi_{F_2}} P(X = x \mid M_{L,C})} && (6) \end{aligned}$$

where χ_{F_i} is the set of worlds where F_i holds, and $P(x \mid M_{L,C})$ is given by Equation 1. The question of whether a knowledge base entails a formula F in first-order logic is the question of whether $P(F \mid L_{\mathrm{KB}}, C_{\mathrm{KB},F}) = 1$, where L_{KB} is the MLN obtained by assigning infinite weight to all the formulas in KB, and $C_{\mathrm{KB},F}$ is the set of all constants appearing in KB or F.

The most widely used approximate solution to probabilistic inference in MLNs is Markov chain Monte Carlo (MCMC) (Gilks *et al.*, 1996). In this framework, the Gibbs sampling algorithm is to generate an *instance* from the distribution of each variable in turn, conditional on the current values of the other variables. One way to speed up Gibbs sampling is by Simulated Tempering (Marinari and Parisi, 1992), which performs simulation in a *generalized ensemble*, and can rapidly achieve an equilibrium state. Poon and Domingos (2006) proposed MC-SAT, an inference algorithm that combines ideas from MCMC and satisfiability.

4 Heuristic Human Knowledge

Even though the Boosting model is able to accommodate a large number of features, some NEs, especially LOCs, ORGs and MISCs are difficult to identify due to lack of linguistic knowledge. For example, some ORGs are possibly mistagged as LOCs and/or MISCs. We incorporate heuristic human knowledge via MLNs to validate the Boosting NER hypotheses. We extract 151 location salient words and 783 organization salient words from the LDC Chinese-English bi-directional NE lists compiled from Xinhua News database. We also make a punctuation list which contains 19 items. We make the following assumptions to validate the Boosting results:

- Obviously, if a tagged entity ends with a location salient word, it is a LOC. If a tagged entity ends with an organization salient word, it is an ORG.
- If a tagged entity is close to a subsequent location salient word, probably they should be combined together as a LOC. The closer they are, the more likely that they should be combined.
- Heuristically, if a series of consecutive tagged entities are close to a subsequent organization salient word, they should probably be combined together as an ORG because an ORG may contain multiple PERs, LOCs, MISCs and ORGs.
- Similarly, if there exists a series of consecutive tagged entities and the last one is tagged as an ORG, it is likely that all of them should be combined as an ORG.
- Entity length restriction: all kinds of tagged entities cannot exceed 25 Chinese characters.
- Punctuation restriction: in general, all tagged entities cannot span any punctuation.
- Since all NEs are proper nouns, the tagged entities should end with noun words.

All the above human knowledge can be formulized as first-order logic to construct the structure of MLNs. And all the validated Boosting results are accepted as new NE candidates (or common nouns). We train an MLN to recognize them.

5 Experiments

We randomly selected 15,000 and 3,000 sentences from the People's Daily corpus as training and test sets, respectively. We used the *decision stump*[2] as the weak classifier in Boosting to construct a character-based Chinese NER baseline system.

The features used were as follows:

- 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 [3] chunk tags of the previous 3 characters.

We declared 10 *predicates* and specified 9 first-order formulas according to the heuristic human knowledge in Section 4. For example, we used `person(candidate)` to predicate whether a candidate is a PER. *Formulas* are recursively constructed from atomic formulas using logical connectives and quantifiers. They are constructed using four types of symbols: *constants*, *variables*, *functions*, and *predicates*. *Constant* symbols represent objects in the domain of interest (e.g., "北京/Beijing" and "上海/Shanghai" are LOCs). *Variable* symbols range over the objects in the domain. *Function* symbols represent mappings from tuples of objects to objects. *Predicate* symbols represent relations among objects in the domain or attributes of objects. For example, the formula `endwith(r, p)^locsalientword(p)=>location(r)` means if `r` ends with a location salient word `p`, then it is a LOC.

[2] 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.

[3] 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).

We extracted all the distinct NEs (4,475 PERs, 2,170 LOCs, 2,823 ORGs and 614 MISCs) from the 15,000-sentence training corpus. An MLN training database, which consists of 10 *predicates* and 44,810 ground atoms was built. A *ground atom* is an atomic formula all of whose arguments are ground terms (terms containing no variables). For example, the ground atom `location(北京市)` conveys that "北京市/Beijing City" is a LOC.

During MLN learning, each formula is converted to Conjunctive Normal Form (CNF), and a weight is learned for each of its clauses. The weight of a clause is used as the mean of a Gaussian prior for the learned weight. These weights reflect how often the clauses are actually observed in the training data.

We validated 352 Boosting results to construct the MLN testing database, which contains 1,285 entries and these entries are used as *evidence* for inference. Inference is performed by grounding the minimal subset of the network required for answering the query predicates. We applied 3 MCMC algorithms: Gibbs sampling (GS), MC-SAT and Simulated Tempering (ST) for inference and the comparative NER results are shown in Table 1. The cascaded hybrid model greatly outperforms the Boosting model. We obtained the same results using GS and ST algorithms. And GS (or ST) yields slightly better results than the MC-SAT algorithm.

Table 1: Comparative Chinese NER Results

	Precision	Recall	$F_{\beta=1}$
Boosting			
PER	99.39%	99.06%	99.22
LOC	87.55%	91.81%	89.63
ORG	82.15%	66.61%	73.57
MISC	80.00%	87.84%	83.74
Overall	90.26%	89.42%	89.84
Hybrid (MC-SAT)			
PER	99.39%	99.06%	99.22
LOC	94.83%	91.81%	93.30
ORG	87.82%	85.69%	86.74
MISC	93.53%	85.10%	89.12
Overall	95.01%	92.78%	93.88
Hybrid (GS/ST)			
PER	99.39%	99.06%	99.22
LOC	94.80%	91.91%	93.34
ORG	87.82%	86.28%	87.04
MISC	93.53%	85.10%	89.12
Overall	94.99%	92.93%	93.95

6 Conclusion

In this paper we propose a cascaded hybrid model for Chinese NER. We incorporate human heuristics via MLNs, which produce a set of weighted first-order clauses to validate Boosting NER hypotheses. To the best of our knowledge, this is the first attempt at using MLNs for the NER problem in the NLP community. Experiments on People's Daily corpus illustrate the promise of our approach. Directions for future work include learning the structure of MLNs automatically and using MLNs for information extraction and statistical relational learning (e.g., entity relation identification).

References

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.

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.

Michael R. Genesereth and Nils J. Nislsson. *Logical foundations of artificial intelligence*. Morgan Kaufmann Publishers Inc., San Mateo, CA, 1987.

W.R. Gilks, S. Richardson, and D.J. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman and Hall, London, UK, 1996.

Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

Enzo Marinari and Giorgio Parisi. Simulated Tempering: A new Monte Carlo scheme. *Europhysics Letters*, 19:451–458, 1992.

Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 1988.

Hoifung Poon and Pedro Domingos. Sound and efficient inference with probabilistic and deterministic dependencies. In *21st National Conference on Artificial Intelligence (AAAI-06)*, Boston, Massachusetts, July 2006. The AAAI Press.

Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.

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

Maosong Sun, Changning Huang, Haiyan Gao, and Jie Fang. Identifying Chinese names in unrestricted texts. *Journal of Chinese Information Processing*, 1995.

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.

Xiaofeng Yu, Marine Carpuat, and Dekai Wu. Boosting for Chinese named entity recognition. In *5th SIGHAN Workshop on Chinese Language Processing*, Australia, July 2006.