Chinese Named Entity Recognition with Conditional Random Fields

Wenliang Chen and **Yujie Zhang** and **Hitoshi Isahara**
Computational Linguistics Group
National Institute of Information and Communications Technology
3-5 Hikari-dai, Seika-cho, Soraku-gun, Kyoto, Japan, 619-0289
{chenwl, yujie, isahara}@nict.go.jp

Abstract

We present a Chinese Named Entity Recognition (NER) system submitted to the close track of Sighan Bakeoff2006. We define some additional features via doing statistics in training corpus. Our system incorporates basic features and additional features based on Conditional Random Fields (CRFs). In order to correct inconsistently results, we perform the post-processing procedure according to n-best results given by the CRFs model. Our final system achieved a F-score of 85.14 at MSRA, 89.03 at CityU, and 76.27 at LDC.

1 Introduction

Named Entity Recognition task in the 2006 Sighan Bakeoff includes three corpora: Microsoft Research (MSRA), City University of Hong Kong (CityU), and Linguistic Data Consortium (LDC). There are four types of Named Entities in the corpora: Person Name, Organization Name, Location Name, and Geopolitical Entity (only included in LDC corpus).

We attend the close track of all three corpora. In the close track, we can not use any external resources. Thus except basic features, we define some additional features by applying statistics in training corpus to replace external resources. Firstly, we perform word segmentation using a simple left-to-right maximum matching algorithm, in which we use a word dictionary generated by doing n-gram statistics. Then we define the features based on word boundaries. Secondly, we generate several lists according to the relative position to Named Entity (NE). We define another type of features based on these lists. Using these features, we build a Conditional Random Fields(CRFs)-based Named Entity Recognition (NER) System. We use the system to generate n-best results for every sentence, and then perform a post-processing.

2 Conditional Random Fields

2.1 The model

Conditional Random Fields(CRFs), a statistical sequence modeling framework, was first introduced by Lafferty et al(Lafferty et al., 2001). The model has been used for chunking(Sha and Pereira, 2003). We only describe the model briefly since full details are presented in the paper(Lafferty et al., 2001).

In this paper, we regard Chinese NER as a sequence labeling problem. For our sequence labeling problem, we create a linear-chain CRFs based on an undirected graph $G = (V, E)$, where V is the set of random variables $Y = \{Y_i | 1 \leq i \leq n\}$, for each of n tokens in an input sentence and $E = \{(Y_{i-1}, Y_i) | 1 \leq i \leq n\}$ is the set of $n-1$ edges forming a linear chain. For each sentence x, we define two non-negative factors:
$exp(\sum_{k=1}^{K} \lambda_k f_k(y_{i-1}, y_i, x))$ for each edge
$exp(\sum_{k=1}^{K'} \lambda_k^{'} f_k^{'}(y_i, x))$ for each node
where f_k is a binary feature function, and K and K' are the number of features defined for edges and nodes respectively. Following Lafferty et al(Lafferty et al., 2001), the conditional probability of a sequence of tags y given a sequence of tokens x is:

$$P(y|x) = \frac{1}{Z(x)} exp(\sum_{i,k} \lambda_k f_k(y_{i-1}, y_i, x) + \sum_{i,k} \lambda_k^{'} f_k^{'}(y_i, x)) \quad (1)$$

where $Z(x)$ is the normalization constant. Given the training data D, a set of sentences (characters

Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing, pages 118–121,
Sydney, July 2006.

Tag	Meaning
0 (zero)	Not part of a named entity
PER	A person name
ORG	An organization name
LOC	A location name
GPE	A geopolitical entity

Table 1: Named Entities in the Data

with their corresponding tags), the parameters of the model are trained to maximize the conditional log-likelihood. When testing, given a sentence x in the test data, the tagging sequence y is given by $Argmax_{y'}P(y'|x)$.

CRFs allow us to utilize a large number of observation features as well as different state sequence based features and other features we want to add.

2.2 CRFs for Chinese NER

Our CRFs-based system has a first-order Markov dependency between NER tags.

In our experiments, we do not use feature selection and all features are used in training and testing. We use the following feature functions:

$$f(y_{i-1}, y_i, x, i) = p(x, i)q(y_{i-1}, y_i) \quad (2)$$

where $p(x, i)$ is a predicate on the input sequence x and current position i and $q(y_{i-1}, y_i)$ is a predicate on pairs of labels. For instance, $p(x, i)$ might be "the char at position i is 和(and)".

In our system, we used CRF++ (V0.42)[1] to implement the CRFs model.

3 Chinese Named Entity Recognition

The training data format is similar to that of the CoNLL NER task 2002, adapted for Chinese. The data is presented in two-column format, where the first column consists of the character and the second is a tag.

Table 1 shows the types of Named Entities in the data. Every character is to be tagged with a NE type label extended with B (Beginning character of a NE) and I (Non-beginning character of a NE), or 0 (Not part of a NE).

To obtain a good-quality estimation of the conditional probability of the event tag, the observations should be based on features that represent the difference of the two events. In our system, we define three types of features for the CRFs model.

[1]CRF++ is available at http://chasen.org/ taku/software/CRF++/

3.1 Basic Features

The basic features of our system list as follows:

- . $C_n(n = -2, -1, 0, 1, 2)$
- . $C_nC_{n+1}(n = -1, 0)$

Where C refers to a Chinese character while C_0 denotes the current character and $C_n(C_{-n})$ denotes the character n positions to the right (left) of the current character.

For example, given a character sequence "张福贵先生", when considering the character C_0 denotes "贵", C_{-1} denotes "福", $C_{-1}C_0$ denotes "富贵", and so on.

3.2 Word Boundary Features

The sentences in training data are based on characters. However, there are many features related to the words. For instance, the word "先生" can be a important feature for Person Name. We perform word segmentation using the left-to-right maximum matching algorithm, in which we use a word dictionary generated by doing n-gram statistics in training corpus. Then we use the word boundary tags as the features for the model.

Firstly, we construct a word dictionary by extracting N-grams from training corpus as follows:

1. Extract arbitrary N-grams ($2 \leq n \leq 10$, $Frequency \geq 10$) from training corpus. We get a list W_1.
2. Use a tool to perform statistical substring reduction in W_1[described in (Lv et al., 2004)][2]. We get a list W_2.
3. Construct a character list (CH)[3], in which the characters are top 20 frequency in training corpus.
4. Remove the strings from W_2, which contain the characters in the list CH. We get final N-grams list W_3.

Secondly, we use W_3 as a dictionary for left-to-right maximum matching word segmentation. We assign word boundary tags to sentences. Each character can be assigned one of 4 possible boundary tags: "B" for a character that begins a word and is followed by another character, "M" for a

[2]Tools are available at http://homepages.inf.ed.ac.uk/s0450736/Software

[3]To collect some characters such as punctuation, "的", "了" and so on.

character that occurs in the middle of a word, "E" for a character that ends a word, and "S" for a character that occurs as a single-character word.

The word boundary features of our system list as follows:

- . $WT_n(n = -1, 0, 1)$

Where WT refers to the word boundary tag while WT_0 denotes the tag of current character and $WT_n(WT_{-n})$ denotes the tag n positions to the right (left) of the current character.

3.3 Char Features

If we can use external resources, we often use the lists of surname, suffix of named entity and prefix of named entity for Chinese NER. In our system, we generate these lists automatically from training corpus by the procedure as follows:

- PSur: uni-gram characters, first characters of Person Name. (surname)
- PC: uni-gram characters in Person Name.
- PPre: bi-gram characters before Person Name. (prefix of Person Name)
- PSuf: bi-gram characters after Person Name. (suffix of Person Name)
- LC: uni-gram characters in Location Name or Geopolitical entity.
- LSuf: uni-gram characters, the last characters of Location Name or Geopolitical Entity. (suffix of Location Name or Geopolitical Entity)
- OC: uni-gram characters in Organization Name.
- OSuf: uni-gram characters, the last characters of Organization Name. (suffix of Organization Name)
- OBSuf: bi-gram characters, the last two characters of Organization Name. (suffix of Organization Name)

We remove the items in uni-gram lists if their frequencies are less than 5 and in bi-gram lists if their frequencies are less than 2. Based on these lists, we assign the tags to every character. For instance, if a character is included in PSur list, then we assign a tag "PSur_1", otherwise assign a tag "PSur_0". Then we define the char features as follows:

- . $PSur_0PC_0$;
- . $PSur_nPC_nPSur_{n+1}PC_{n+1}(n = -1, 0)$;
- . $PPre_0$;
- . $PSuf_0$;
- . LC_0OC_0;

S is the list of sentences, $S = \{s_1, s_2, ..., s_n\}$.
T is m-best results of S, $T = \{t_1, t_2, ..., t_n\}$, which t_i is a set of m-best results of s_i.
p_{ij} is the score of t_{ij}, that is the jth result in t_i.

Collect NE list:
Loop i in $[1, n]$
if($p_{i0} \geq 0.5$){
Exacting all NEs from t_{i0} to add into NEList.}

Replacing:
Loop i in $[1, n]$
if($p_{i0} \geq 0.5$){
FinalResult(s_i) = t_{i0}.}
else{
TmpResult = t_{i0}.
Loop j in $[m, 1]$
if(the NEs in t_{ij} is included in NEList){
Replace the matching string in TmpResult with new NE tags.}
FinalResult(s_i) = TmpResult.
}

Table 2: The algorithm of Post-processing

- . $LC_nOC_nLC_{n+1}OC_{n+1}(n = -1, 0)$;
- . $LSuf_0OSuf_0$;
- . $LSuf_nOSuf_nLSuf_{n+1}OSuf_{n+1}(n = -1, 0)$;

4 Post-Processing

There are inconsistently results, which are tagged by the CRFs model. Thus we perform a post-processing step to correct these errors.

The post-processing tries to assign the correct tags according to n-best results for every sentence. Our system outputs top 20 labeled sequences for each sentence with the confident scores. The post-processing algorithm is shown at Table 2. Firstly, we collect NE list from high confident results. Secondly, we re-assign the tags for low confident results using the NE list.

5 Evaluation Results

5.1 Results on Sighan bakeoff 2006

We evaluated our system in the close track, on all three corpora, namely Microsoft Research (MSRA), City University of Hong Kong (CityU), and Linguistic Data Consortium (LDC). Our official Bakeoff results are shown at Table 3, where the columns P, R, and FB1 show precision, recall and F measure($\beta = 1$). We used all three types of features in our final system.

In order to evaluate the contribution of features, we conducted the experiments of each type of features using the test sets with gold-standard dataset. Table 4 shows the experimental results,

MSRA	P	R	FB1
LOC	92.81	88.53	90.62
ORG	81.93	81.07	81.50
PER	85.41	74.15	79.38
Overall	88.14	82.34	85.14
CityU	P	R	FB1
LOC	92.21	92.00	92.11
ORG	87.83	74.23	80.46
PER	92.77	89.05	90.87
Overall	91.43	86.76	89.03
LDC	P	R	FB1
GPE	83.78	80.36	82.04
LOC	51.11	21.70	30.46
ORG	71.79	60.82	65.85
PER	82.40	75.58	78.84
Overall	80.26	72.65	76.27

Table 3: Our official Bakeoff results

	MSRA	CityU	LDC
F1	84.73	88.26	76.18
+F2		88.67	76.30
+F3		88.74	
Post	85.23	89.03	76.66

Table 4: Results of different combinations

where F1 refers to use basic features, F2 refers to use the word boundary features, F3 refers to use the char features, and Post refers to perform the post-processing.

The results indicated that word boundary features helped on LDC and CityU, char features only helped on CityU and the post-processing always helped to improve the performance.

6 Conclusion

This paper presented our Named Entity Recognition system for the close track of Bakeoff2006. Our approach was based on Conditional Random Fields model. Except basic features, we defined the additional features by doing statistics in training corpus. In addition, we performed a post-processing according to n-best results generated by the CRFs model. The evaluation results showed that our system achieved state-of-the-art performance on all three corpora in the close track.

References

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML01)*.

Xueqiang Lv, Le Zhang, and Junfeng Hu. 2004. Statistical substring reduction in linear time. In *Proceedings of IJCNLP-04*, HaiNan island, P.R.China.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of HLT-NAACL03*.