The Character-based CRF Segmenter of MSRA&NEU for the 4th Bakeoff

Zhenxing Wang[1,2], Changning Huang[2] and Jingbo Zhu[1]
1 Institute of Computer Software and Theory, Northeastern University,
Shenyang, China, 110004
2 Microsoft Research Asia, 49, Zhichun Road,
Haidian District, Beijing, China, 100080
zxwang@ics.neu.edu.cn
v-cnh@microsoft.com
zhujingbo@mail.neu.edu.cn

Abstract

This paper describes the Chinese Word Segmenter for the fourth International Chinese Language Processing Bakeoff. Base on Conditional Random Field (CRF) model, a basic segmenter is designed as a problem of character-based tagging. To further improve the performance of our segmenter, we employ a word-based approach to increase the in-vocabulary (IV) word recall and a post-processing to increase the out-of-vocabulary (OOV) word recall. We participate in the word segmentation closed test on all five corpora and our system achieved four second best and one the fifth in all the five corpora.

1 Introduction

Since Chinese Word Segmentation was firstly treated as a character-based tagging task in (Xue and Converse, 2002), this method has been widely accepted and further developed by researchers (Peng et al., 2004), (Tseng et al., 2005), (Low et al., 2005), (Zhao et al., 2006). Thus, as a powerful sequence tagging model, CRF became the dominant method in the Bakeoff 2006 (Levow, 2006).

In this paper, we improve basic segmenter under the CRF work frame in two aspects, namely IV and OOV identification respectively. We use the result from word-based segmentation to revise the CRF output so that we gain a higher IV word recall. For the OOV part a post-processing rule is proposed to find those OOV words which are wrongly segmented into several fractions. Our system performs well in the Fourth Bakeoff, achieving four second best and on the fifth in all the five corpora. In the following of this paper, we describe our method in more detail.

The rest of this paper is organized as follows. In Section 2, we first give a brief review to the basic CRF tagging approach and then we propose our methods to improve IV and OOV performance respectively. In Section 3 we give the experiment results on the fourth Bakeoff corpora to show that our method is effective to improve the performance of the segmenter. In Section 4, we conclude our work.

2 Our Word Segmentation System

In this section, we describe our system in more detail. Our system includes three modules: a basic CRF tagger, a word-base segmenter to improve the IV recall and a post-processing rule to improve the OOV recall. In the following of this section, we introduce these three modules respectively.

2.1 Basic CRF tagger

Sequence tagging approach treat Word Segmentation task as a labeling problem. Every character in input sentences will be given a label which indicates whether this character is a word boundary. Our basic CRF[1] tagger is almost the same as the system described in (Zhao et al., 2006) except we add a feature to incorporate word information, which is learned from training corpus.

[1] CRF tagger in this paper is implemented by CRF++ which is downloaded from http://crfpp.sourceforge.net/

Type	Feature	Function
Unigram	C_{-1}, C_0, C_1	Previous, current and next character
Bigram	C_{-1} C_0, C_0 C_1	Two adjacent character
Jump	C_{-1} C_1	Previous character and next character
Word Flag	F_0 F_1	Whether adjacent characters form an IV word

Table 1 Feature templates used for CRF in our system

Under the CRF tagging scheme, each character in one sentence will be given a label by CRF model to indicate which position this character occupies in a word. In our system, CRF tag set is proposed to distinguish different positions in the multi-character words when the word length is less than 6, namely 6-tag set {B, B2, B3, M, E, O}. Here, Tag B and E stand for the first and the last position in a multi-character word, respectively. S stands up a single-character word. B2 and B3 stand for the second and the third position in a multi-character word, whose length is larger than two-character or three-character. M stands for the fourth or more rear position in a multi-character word, whose length is larger than four-character.

We add a new feature, which also used in maximum entropy model for word segmentation task by (Low et al., 2005), to the feature templates for CRF model while keep the other features same as (Zhao et al., 2006). The feature templates are defined in table 1. In the feature template, only the Word Flag feature needs an explanation. The binary function F_0 = 1 if and only if C_{-1} C_0 form a IV word, else F_0 = 0 and F_1 = 1 if and only if C_0 C_1 form a IV word, else F_1 = 0.

2.2 Word based segmenter and revise rules

For the word-based word segmentation, we collect dictionary from training corpus first. Instead of Maximum Match, trigram language model [2] trained on training corpus is employed for disambiguation. During the disambiguation procedure, a beam search decoder is used to seek the most possible segmentation. For detail, the decoder reads characters from the input sentence one at a time, and generates candidate segmentations incrementally. At each stage, the next incoming character is combined with an existing candidate in two different ways to generate new candidates: it is either appended to the last word in the candidate, or taken as the start of a new word. This method guarantees exhaustive generation of possible segmentations for any input sentence. However, the exponential time and space of the length of the input sentence are needed for such a search and it is always intractable in practice. Thus, we use the trigram language model to select top B (B is a constant predefined before search and in our experiment 3 is used) best candidates with highest probability at each stage so that the search algorithm can work in practice. Finally, when the whole sentence has been read, the best candidate with the highest probability will be selected as the segmentation result.

After we get word-based segmentation result, we use it to revise the CRF tagging result similar to (Zhang et al., 2006). Since word-based segmentation result also corresponds to a tag sequence according to the 6-tag set, we now have two tags for each character, word-based tag (WT) and CRF tag (CT). Which tag will be kept as the final result depends on Marginal Probability (MP) of the CT.

Here, we give a short explanation about what is the MP of the CT. Suppose there is a sentence $C = c_0 c_1 ... c_M$, where c_i is the character this sentence containing. CRF model gives this sentence a optimal tag sequence $T = t_0 t_1 ... t_M$, where t_i is the tag for c_i . If $t_i = t$ and $t \in \{B, B_2, B_3, M, E, S\}$, the MP of t_i is defined as:

$$\text{MP}(t_i = t) = \frac{\sum_{T, t_i = t} P(T \mid C)}{\sum_T P(T \mid C)}$$

Here, $P(T \mid C)$ is the conditional probability given by CRF model. For more detail about how to calculate this conditional probability, please refer to (Lafferty et al., 2001).

Assume that the tag assigned to the current character is CT by CRF and WT by word-based segmenter respectively. The rules under which we revise CRF result with word-based result is that if MP(CT) of a character is less than a predefined threshold and WT is not "S", the WT of this character will be kept as the final result, else the CT of the character will be kept as the final result.

[2] Language model used in this paper is SLRIM downloaded from http://www.speech.sri.com/projects/srilm/

The restriction that WT should not be “S” is reasonable because word-based segmentation is incapable to recognize the OOV word and always segments OOV word into single characters. Besides CRF model is better at dealing with OOV word than our word-based segmentation. When WT is “S” it is possible that current word is an OOV word and segmented into single character wrongly by the word-based segmenter, so the CT of the character should be kept under such situation. For more detail about this analysis please refer to (Wang et al., 2008).

2.3 Post-processing rule

The rules we described in last subsection is helpful to improve the IV word recall and now we introduce our post-processing rule to improve the OOV recall.

Our post-processing rule is designed to deal with one typical type of OOV errors, namely an OOV word wrongly segmented into several parts. In practice many OOV errors belong to such type.

The rule is quite simple. When we read a sentence from the result we get by the last step, we also kept the last N sentences in memory, in our system we set N equals to 20. We do this because adjacent sentences are always relevant and some named entity likely occurs repeatedly in these sentences. Then, we scan these sentences to find all n-grams (n from 2 to 7) and count their occurrence. If certain n-gram appears more than a threshold and this n-gram never appears in training corpus, the n-gram will be selected as a word candidate. Then, we filter these word candidates according to the context entropy (Luo and Song, 2004). Assume w is a word candidate appears n times in the current sentence and last N sentences and $\alpha = \{a_0, a_1, ..., a_l\}$ is the set of left side characters of w. Left Context Entropy (LCE) can be defined as:

$$LCE(w) = \frac{1}{n} \sum_{a_i \in \alpha} C(a_i, w) \log \frac{n}{C(a_i, w)}$$

Here, $C(a_i, w)$ is the count of concurrence of a_i and w. For the Right Context Entropy, the definition is the same except change left into right. Now, we define Context Entropy (CE) of a word candidate w as $\min(LCE(w), RCE(w))$. The word candidates with CE larger than a predefined threshold will be bind as a whole word in test corpus no matter what tag sequence the segmenter giving it. If a shorter n-gram is contained in a longer n-gram and both of them satisfy the above condition, the shorter n-gram will be overlooked and the longer n-gram is bind as a whole word.

3 Evaluation of Our System

On the corpora of the Fourth Bakeoff, we evaluate our system. We carry out our evaluation on the closed tracks. It means that we do not use any additional knowledge beyond the training corpus. The thresholds set for MP and CE on each corpus are tuned on left-out data of training corpus by cross validation. To analyze our methods on IV and OOV words, we use a detailed evaluation metric than Bakeoff 2006 (Levow, 2006) which includes Foov and Fiv. Our results are shown in Table 2. In Table 2, the row “Basic Model” means the results produced by our basic CRF tagger, the row “+IV” means the results produced by the combination of CRF tagger and word-based segmenter and the row “+IV+OOV” means the result we get by executing post-processing rule on the combination results. The F measure of the basic CRF tagger alone in the Table 2 is within the top three in the closed tests except Cityu. Performance on Cityu corpus is not so good because the inconsistencies existing in Cityu training and test corpora. In the training corpus the quotation marks are 「」while in test corpus quotation marks are “ ” , which never apper in the training corpus. As a reult, a lot of errors were caused by quotation marks. For example, the following four character “事業” were combined as a one word in our result and fragment “越位” was tagged as two words “越 and 位” . Because CRF tagger never met “ and ” in training corpus so the tagger gave the most common tags, namely B and E to the quotation marks, which cause segmentation errors not only on quotation marks themselves but also on the characters adjacent to them. We remove these inconsistencies munually and got the F measure 0.5 percentage higer than the rusult in table 2. This result is within the top three in the closed tests. On all the five corpora, our “+IV” module can increase the Fiv and our “+OOV” module can increase Foov respectively. However, these improvements are not significant.

Corpus	Method	R	P	F	R_{OOV}	P_{OOV}	F_{OOV}	R_{IV}	P_{IV}	F_{IV}
CKIP	Basic Model	0.946	0.923	0.940	0.651	0.719	0.683	0.969	0.948	0.958
	+ IV	0.949	0.935	0.942	0.647	0.741	0.691	0.973	0.948	0.960
	+ IV + OOV	0.950	0.936	0.943	0.656	0.748	0.699	0.973	0.949	0.961
CityU	Basic Model	0.944	0.934	0.939	0.654	0.721	0.686	0.970	0.951	0.960
	+ IV	0.946	0.936	0.941	0.655	0.738	0.694	0.972	0.951	0.962
	+ IV + OOV	0.949	0.937	0.943	0.678	0.759	0.716	0.973	0.951	0.962
CTB	Basic Model	0.953	0.951	0.952	0.703	0.727	0.715	0.967	0.964	0.965
	+ IV	0.954	0.952	0.953	0.697	0.747	0.721	0.969	0.963	0.966
	+ IV + OOV	0.954	0.953	0.953	0.703	0.749	0.725	0.969	0.964	0.966
NCC	Basic Model	0.940	0.928	0.934	0.438	0.580	0.499	0.965	0.940	0.952
	+ IV	0.944	0.930	0.936	0.434	0.603	0.504	0.969	0.941	0.955
	+ IV + OOV	0.945	0.932	0.939	0.450	0.620	0.522	0.970	0.943	0.956
SXU	Basic Model	0.960	0.953	0.956	0.636	0.674	0.654	0.977	0.967	0.972
	+ IV	0.962	0.955	0.958	0.637	0.696	0.665	0.980	0.967	0.973
	+ IV + OOV	0.962	0.955	0.959	0.645	0.702	0.673	0.979	0.968	0.974

Table 2 performance each step of our system achieves

4 Conclusions and Future Work

In this paper, we propose a three-stage strategy in Chinese Word Segmentation. Based on the results produced by basic CRF, our word-based segmentation module and post-processing module are designed to improve IV and OOV performance respectively. The results above show that our system achieves the state-of-the-art performance. Since only the CRF tagger is good enough as we shown in our experiment, in the future work we will pay effort on the semi-supervised learning for CRF model in order to mining more useful information from training and test corpus for CRF tagger.

References

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. *In Proceedings of ICML-2001*, pages 591–598.

Gina-Anne Levow. 2006. The Third International Chinese Language Processing Bakeoff: Word Segmentation and Named Entity Recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing* , pages 108-117, Sydney: July.

Jin Kiat Low, Hwee Tou Ng, and Wenyuan Guo. 2005. A Maximum Entropy Approach to Chinese Word Segmentation. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, pages 161-164, Jeju Island, Korea.

Zhiyong Luo, Rou Song, 2004. "An integrated method for Chinese unknown word extraction", In *Proceedings of Third SIGHAN Workshop on Chinese Language Processing*, pages 148-154. Barcelona, Spain.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In COLING 2004, pages 562–568. Geneva, Switzerland.

Huihsin Tseng, Pichuan Chang et al. 2005. A Conditional Random Field Word Segmenter for SIGHAN Bakeoff 2005. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, pages 168-171, Jeju Island, Korea.

Zhenxing Wang, Changning Huang and Jingbo Zhu. 2008. Which Performs Better on In-Vocabulary Word Segmentation: Based on Word or Character? In *Proceeding of the Sixth Sighan Workshop on Chinese Language Processing*. To be published.

Neinwen Xue and Susan P. Converse. 2002. Combining Classifiers for Chinese Word Segmentation. In *Proceedings of the First SIGHAN Workshop on Chinese Language Processing*, pages 63-70, Taipei, Taiwan.

Ruiqiang Zhang, Genichiro Kikui and Eiichiro Sumita. 2006. Subword-based Tagging by Conditional Random Fields for Chinese Word Segmentation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion volume*, pages 193-196. New York, USA.

Hai Zhao, Changning Huang et al. 2006. Effective Tag Set Selection in Chinese Word Segmentation via Conditional Random Field Modeling. In *Proceedings of PACLIC-20*. pages 87-94. Wuhan, China, Novemeber.