Adapting Conventional Chinese Word Segmenter for Segmenting Micro-blog Text: Combining Rule-based and Statistic-based Approaches

Ning Xi, Bin Li, Guangchao Tang, Shujian Huang, Yinggong Zhao,
Hao Zhou, Xinyu Dai, Jiajun Chen
State Key Laboratory for Novel Software Technology,
Department of Computer Science and Technology,
Nanjing University, Nanjing, 210023, China
{xin,lib,tanggc,huangsj,zhaoyg,zhouh,dxy,chenjj}@nlp.nju.edu.cn

Abstract

We describe two adaptation strategies which are used in our word segmentation system in participating the Micro-blog word segmentation bake-off: Domain invariant information is extracted from the in-domain unlabelled corpus, and is incorporated as supplementary features to conventional word segmenter based on Conditional Random Field (CRF), we call it *statistic-based* adaptation. Some heuristic rules are further used to post-process the word segmentation result in order to better handle the characters in emoticons, name entities and special punctuation patterns which extensively exist in micro-blog text, and we call it *rule-based* adaptation. Experimentally, using both adaptation strategies, our system achieved 92.46 points of F-score, compared with 88.73 points of F-score of the unadapted CRF word segmenter on the pre-released development data. Our system achieved 92.51 points of F-score on the final test data.

1 Introduction

Recent years have witnessed the great development of Chinese word segmentation (CWS) techniques. Among various approaches, character labelling via *Conditional Random Field* (CRF) modelling has become a prevailing technique (Lafferty et al., 2001; Xue, 2003; Zhao et al., 2006), due to its good performance in OOV words recognition and low development cost. Given a large-scale corpus with human annotation, the only issue the developer need to focus on is to design an expressive set of feature templates which captures the various characteristics of word segmentation to achieve better performance.

The demand for Chinese micro-blog data mining has been unprecedentedly increased, owing to the growing number of the Chinese micro-blog users in the past few years. In these tasks, Chinese word segmentation plays an important role in correctly understanding the micro-blog text. Chinese word segment on the micro-blog text is a challenging task. On one hand, it is difficult to obtain large-scale labelled corpora of micro-blog domain for CRF-based learning, and the only labelled corpus we have is *People's Daily corpus* (PDC) which comes from the News domain; on the other, compared with the News text, the micro-blog text contains a large number of new words, name entities, URLs, emoticons (such as ":)"), punctuation patterns (such as "...."), as well as structured symbols representing conversation ("@"), repost("//@"), and topic ("#...#") etc. The word distribution and usage of micro-blog text are also much more free than the News text, making things more difficult.

In this paper, we adapt the conventional Chinese word segmenter which is trained on out-of-domain (News domain) labelled corpus using CRF to segment in-domain micro-blog text, without using any information from the labelled in-domain data. We use two adaptation strategies: the first is *statistic-based adaptation*. We incorporate domain invariant information extracted from the in-domain unlabelled corpus as supplementary features to the conventional CRF segmenter, in order to enhance its ability of recognizing domain-specific words. The unlabelled corpus can be conveniently crawled from the web; the other is *rule-based adaptation*. We proposed some heuristic rules to further post-process the word segmentation result in order to enhance to better handle the

characters in emoticons, name entities and special punctuation patterns which extensively exist in micro-blog text. Experimentally, using both adaptation strategies, our system achieved 92.46 points of F-score, compared with 88.73 points of F-score of the unadapted CRF word segmenter on the pre-released development data. Our system achieved 92.51 points of F-score on the final test data.

2 System Description

In this section, we describe our adapted CRF-based word segmenter.

2.1 Basic Model

Chinese word segmentation (CWS) was first formulated as a character tagging problem by Xue (2003). This approach treats the unsegmented Chinese sentence as a character sequence. It assigns a label to each Chinese character in the sentence, indicating whether a character locates at the beginning of (label "B") of a word, inside ("M") a word, at the end ("E") of a word, or itself forms a single character word ("S"). An example of the labelled sequence is shown in Table 1, which corresponds to the word segmentation "开/出/一朵朵/红莲".

Sequence	开	出	一	朵	朵	红	莲
Label	S	S	B	M	E	B	E

Table 1: An example of labelled sequence

Conditional Random Field (CRF) (Lafferty et al., 2001) is a statistical sequence labelling model. It assigns the probability of a particular label sequence as follows:

$$P(y_1^T|w_1^T) = \frac{exp(\sum_t \sum_k \lambda_k f_k(y_{t-1}, y_t, w_1^T, t))}{Z(w_1^T)} \quad (1)$$

where $w_1^T = w_1 w_2 ... w_T$ is the Chinese character sequence, y_1^T is the corresponding label sequence, t is the index of the character, y_{t-1} and y_t denote the label of the $t-1$th and the t-th character respectively, f_k is a feature function and k ranges from 1 to the number of features, λ_k is the associated feature weight, and $Z(w_1^T)$ is the normalization factor. λ_ks are trained on *People's Daily corpus* (PDC) which is a out-of-domain labelled corpus. In our implementation, CRF++ package[1] was used.

Without any constraint, the CRF model will label Chinese characters as well as non-Chinese characters in the sentence being segmented, including English letters and numeric characters. These non-Chinese characters are strong indicators of word boundaries. Therefore, we use the following heuristics to pre-group these characters: 1) all consecutive English characters. They often form English words or abbreviations (such as "HTC" in sentence "领取HTC手机"), 2) all consecutive numeric characters. They often form numeric words (such as the "205" in sentence "进入205房间"). Splitting these two kinds of consecutive characters will yield meaningless words. Treating these two kinds of words as single units in implementing CRF will not only speed up the decoding process but also improve the segmentation performance on these kinds of words. Moreover, the characters in a URL are pre-grouped using a simple regular expression, and punctuations representing structure symbols (such as conversation ("@"), repost("//@"), topic ("#...#")) are treated as a single unit.

2.2 Feature Template

The primary art in CRF-based CWS is to design an expressive set of features that captures the various characteristics of CWS. In the next, we will elaborate three kinds of features we adopted in our system, including character-based features (section 2.2.1), word-based features (section 2.2.2) and metric-based features (section 2.2.3).

2.2.1 Character-based Features

The character-based features are extensively used by almost all the CRF word segmenters (Xue, 2003; Zhao et al., 2006). Word segmenters incorporating character features have a good generalization ability in recognizing OOV words. To conveniently illustrate the features we used, we denote the current character token c_i, and its context characters $...c_{i-1}c_ic_{i+1}...$. Moreover, we define $p_i = 1$ if c_i is a punctuation character and $p_i = 0$ otherwise, $n_i = 1$ if c_i is numeric character and $n_i = 0$ otherwise, $a_i = 1$ if c_i is English letter and $a_i = 0$ otherwise. The character-based features template associated with each character type are listed in Table 2.

[1] http://crfpp.googlecode.com/svn/trunk/doc/index.html

Type	Template
surface form	$c_{-1}, c_0, c_1, c_{-1}c_0, c_0c_1, c_{-1}c_1$
number	$n_{-1}, n_0, n_1, n_{-1}n_0, n_0n_1, n_{-1}n_1$
punctuation	p_{-1}, p_0, p_1
English letter	$a_{-1}, a_0, a_1, a_{-1}a_0, a_0a_1, a_{-1}a_1$

Table 2: Character-based feature template.

2.2.2 Word-based Features

Combining word-based features and character-based features has been suggested by (Sun 2010; Sun and Xu, 2011), based on the observation that word-based features capture a relatively larger context than character-based features. We define $c_{[i:j]}$ as a string that starts at the i-th character and ends at the j-th character, and then define $D_{[i:j]} = 1$ if $c_{[i:j]}$ matches a word in a pre-defined dictionary, and 0 otherwise. The word-based feature templates are listed in Table 3.

Template
$D_{[i-5:i]}, D_{[i-4:i]}, D_{[i-3:i]}, D_{[i-2:i]}, D_{[i-1:i]}$
$D_{[i:i+1]}, D_{[i:i+2]}, D_{[i:i+3]}, D_{[i:i+4]}, D_{[i:i+5]}$

Table 3: Word-based feature template.

In order to incorporate word-based features, two dictionaries are constructed. The first dictionary consists of words which were directly extracted from the PDC, and the second dictionary consists of the words in the first dictionary as well as the n-grams with length up to 3 which are extracted from the unsegmented micro-blog corpus and have higher confidence scores than a pre-defined threshold. In our system, we choose *Mutual information* (MI) to measures the association between two consecutive characters. The higher the MI, the more likely these two characters are contained in the same word. We adopted the method of Li and Chen (2006) (Eq. 2) to compute the mutual information of strings with length up to four. In practice, we use 7.0 as the threshold.

$$\begin{aligned} MI(a,b) &= \frac{P(ab)}{P(a)P(b)} \\ MI(a,b,c) &= \frac{P(ab)P(bc)P(ac)}{P(a)P(b)P(c)P(abc)} \end{aligned} \quad (2)$$

2.2.3 Metric-based Feature

We use two metrics to compute the confidence of how likely a string in the unsegmented micro-blog text be a word, they are *Accessor Variety* and *Punctuation Variety*. These metrics can be computed conveniently on large-scale in-domain unlabelled corpus using suffix array (Kit and Wilks, 1999). The values of these metrics can be used as supplementary features to the baseline CRF-based word segmenter. These features are domain-invariant (Gao et al., 2010), therefore, the associated feature weights can be trained on out-of-domain labelled corpus. We call the approach *statistic-based adaptation*.

Accessor Variety (AV) is firstly proposed by Feng et al. (2004) in the task of identifying meaningful Chinese words from an unlabelled corpus. The basic idea of this approach is when a string appears under different linguistic contexts, it may carry a meaning. The more contexts a string appears in, the more likely it is a independent word. Given a string s, we define the *left accessor variety* of s as the number of distinct characters that precede s in the corpus, denoted by $L_{AV}(s)$. The higher value $L_{AV}(s)$ is, the more likely that s can be separated at its start position. Similarly, *right accessor variety* of s is defined as the number of distinct characters that follow s in the corpus, denoted by $R_{AV}(s)$. The higher value $R_{AV}(s)$ is, the more likely that s can be separated at its end position.

Punctuation Variety (PV) is a metric similar to AV, which is used by Sun and Xu (2011). The basic idea is when a string appears many times preceding or following punctuations, there tends to be word-breaks succeeding or preceding that string. We define the *left punctuation variety* of a string s as the number of times a punctuation precedes s in a corpus, denoted by $L_{PV}(s)$, and define the *right punctuation variety* of a string s as the number of times a punctuation follows s in a corpus, denoted by $R_{PV}(s)$.

As the values of AV and PV are integers, when incorporating them as features in CRF, simple discretization method is adopted to deal with data sparseness. For example, the value of PV are binned into two intervals. If it is greater than 30, the feature "$PV > 30$" is set to 1 while the feature "PV(0-30)" is set to 0; if the value is less than 30, the feature "$PV > 30$" is set to 0 while the feature "PV(0-30)" is set to 1; The value of AV are also binned into three intervals: "< 30", "30-50", and "> 50", and is incorporated similarly as PV.

Template
$L_{AV}(c_{[i:i+1]}), L_{AV}(c_{[i+1:i+2]})$
$L_{AV}(c_{[i:i+2]}), L_{AV}(c_{[i+1:i+3]})$
$L_{AV}(c_{[i:i+3]}), L_{AV}(c_{[i+1:i+4]})$
$R_{AV}(c_{[i-1:i]}), R_{AV}(c_{[i-2:i-1]})$
$R_{AV}(c_{[i-2:i]}), R_{AV}(c_{[i-3:i-1]})$
$R_{AV}(c_{[i-3:i]}), R_{AV}(c_{[i-4:i-1]})$
$L_{PV}(c_{[i:i+1]}), L_{PV}(c_{[i:i+2]}), L_{PV}(c_{[i:i+3]}),$ $R_{PV}(c_{[i-1:i]}), R_{PV}(c_{[i-2:i]}), R_{PV}(c_{[i-3:i]})$

Table 4: Feature template of accessor variety and punctuation variety.

2.3 Rule-based Adaptation

We proposed some heuristic rules to further post-process the results given by the word segmenter as described above, in order to better handle the following patterns which are hard to recognize otherwise.

Emoticon In the original output of CRF segmenter, characters representing an emoticon are usually separated by spaces. For example, the emoticon ":-D" is usually segmented as ": - D" which does not preserve the meaning of "smile". To reduce the segmentation errors like this, we collected a list of emoticons from the web. For each emoticon in the list, we create a regular expression which removes any intervening space in this emoticon.

Full Stops In the micro-blog text, consecutive stops such as "..." or consecutive Chinese stops such as "。。。。" are often used to express the meaning of being surprised or embarrassed. We create a rule to group these stops. According to the official pre-released development data (see section 3.1), every three consecutive stops from left to right in the output of CRF segmenter are grouped as a token, the remaining one or two stops are also grouped when necessary.

Name Entities As our system does not have separate modules to recognize name entities, we leverage ICTCLAS[2] to recognize them. We use the ICTCLAS to segment and POS-tag the micro-blog text. If a word is POS-tagged as *nr, ns, nt, nz, nl, or ng* by ICTCLAS, we adjusted our word segmentation to accept this word too.

Setting	P	R	F
CRF	89.18	88.29	88.73
+RB	91.34	91.72	91.53
+RB+WF0	90.67	93.94	92.28
+RB+WF1	91.80	92.26	92.03
+RB+MF	91.99	91.18	91.58
+RB+WF0+MF	91.15	93.82	92.46
+RB+WF1+MF	91.91	92.21	92.06

Table 5: Results of our systems on development data, measured in **P**: precision, **R**: recall, and **F**: F-score. **RB**: rule-base adaptation. **WF0**: word-based feature using dictionary extracted from data (a). **WF1**: word-based feature using dictionary extract from both data (a) and data (b). **MF**: metric-based feature.

3 Experiments

3.1 Data

The following four pieces of data were used in our experiment:

(a) out-of-domain labelled corpus. *People's Daily Corpus* of the first half year in 1998, which is segmented under PKU specification[3]. It was used as CRF training corpus;

(b) in-domain unlabelled corpus. It is a large micro-blog corpus containing 1.9M sentences crawled from the web. It was used to compute word-based features or metric-based features for CRF training;

(c) official pre-released development data. It contains 600 segmented sentences in micro-blog domain under PKU specification. In our experiments, it is only used as **development data** to choose the best setting;

(d) official released test data. It is used for final evaluation.

Full-width characters in all the above data are converted to the corresponding half-width characters. Traditional Chinese characters are also converted to their simplified version.

[2] a well-known Chinese word segmenter/POS-tagger downloaded from www.ictclas.org/

[3] PKU specification is adopted in this track

	P	R	F	CS	CS(%)
Baseline+RB+WF0	0.924	0.9262	0.9251	1628	32.56
Best System	0.946	0.9496	0.9478	2244	44.88

Table 6: Comparison of our system and the best system in the Bake-off on the final test data. **CS**: the number of correct sentences. **CS(%)**: percent of the number of correct sentences.

3.2 Results on development data

We first conducted experiments on the development data to investigate the effectiveness of various features. Table 5 shows the results of seven settings in terms of precision, recall and F-score. **Baseline** represents the setting of the conventional CRF, where only character-based features were incorporated, and no adaptation strategy was used. As we can see, having incorporated rule-based adaptation into the baseline, as shown in **Baseline+RB**, the F-score was significantly improved from 88.73 to 91.53, which achieved a 24.8% reduction of error rate. This improvement shows that rule-based adaptation is an very simple and effective approach in adapting a conventional word segmenter to work on micro-blog domain.

We next investigated incorporating word-based features into **Baseline+RB**. As noted in section 2.2.2, we tried two dictionaries respectively, the first dictionary was extracted from only data (a), denoted by **Baseline+RB+WF0**; and the other dictionary was extracted from both data (a) and data (b), denoted by **Baseline+RB+WF1**. We see that using the first dictionary yielded an improvement of 0.75 points of F-scores, compared to **Baseline+RB**. However, using the second dictionary yielded an improvement of 0.5 F-score only. These results suggest that incorporating word-based features do improve the word segmentation results, however, its effectiveness could rely heavily on the quality of the dictionary. The first dictionary consists of words extracted from from data (a), which is annotated by humans, thus it is of high quality. However, the words extracted from data (b) are not guaranteed to be genuine words because they are included into the second dictionary as long as their confidence scores were higher than the threshold. The noisy words in the second dictionary seem to be blame for the worse results in **Baseline+RB+WF1**.

We then evaluated the impact of incorporating metric-based features. Moving from **Baseline+RB** to **Baseline+RB+MF**, the F-score increased from 91.51 to 91.58. It seems that the metric-based features are not very useful. However, comparing **Baseline+RB+WF0** and **Baseline+RB+WF0+MF**, the improvement increased from 92.28 to 92.46, and **Baseline+RB+WF0+MF** achieved the best performance among all settings, indicating the effectiveness of using metric-based features. Again, **Baseline+RB+WF0+MF** outperformed **Baseline+RB+WF1+MF**, which confirms the conclusion we draw in the last paragraph. Overall, both rule-based adaptation and statistic-based adaptation work well in micro-blog word segmentation.

Finally, we present the results of our system and the best system on the test data in Table 6. Although our results underperformed the best system with a margin of 2.27 points of F-score, we did not use any information extracted from in-domain labelled corpus, i.e. development corpus.

4 Conclusions and Future Works

We describe our Chinese word segmentation systems that we developed for participating the Chinese Micro-blog Word Segmentation Bakeoff. We adapt the conventional Chinese word segmenter which is trained on segmented News domain corpus by Conditional Random Field (CRF) to work on text from the micro-blog domain. Both statistic-based and rule-based adaptation strategies are demonstrated useful in micro-blog word segmentation.

In the future, we will firstly try to investigate how to incorporate more effective domain invariant features to improve the results. We will also try to develop better domain-specific name entity recognition tools to further enhance the performance.

Acknowledgements

We thank anonymous reviewers for their constructive comments. This work is supported by the National Natural Science Foundation of China (No. 61003112 and No. 61170181), the Research Fund for the Doctoral Program of Higher Education

of China (Grant No. 20110091110003), China Post Doctoral Fund under contract 2012M510178, and Jiangsu Post Doctoral Fund under contract 1101065C.

References

Weiwei Sun. 2010. Word-based and Character-based Word Segmentation Models: Comparison and Combination. *Proceedings of COLING 2010*, 1211–1219.

Weiwei Sun, Jia Xu. 2011. Enhancing Chinese Word Segmentation Using Unlabelled Data. *Proceedings of the 2011 Conference on Empirical Methods in natural language Processing*, 970–979.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Comditional Random Fields: Probabilistic models for segmenting and labelling sequence data. *Proceedings of ICML 2001*, 282–289.

Haodi Feng, Kang Chen, Xiaotie Deng, Weimin Zheng. 2004. Accessor Variety Criteria for Chinese Word Extraction. *Computational Linguistics*, 30:75–93.

Wenjun Gao, Xipeng Qiu, and Xuanjing Huang. 2010. Adaptive Chinese Word Segmentation with Online Passive-Aggressive Algorithm. *CIPS-SIGHAN Joint Conference on Chinese Language Processing*.

Zhongguo Li and maosong Sun. 2009. Punctuation as Implicit Annotations for Chinese Word Segmentation. *Computational Linguistics*, 35:505–512.

Chunyu Kit and Yorick Wilks. 1998. The Virtual Corpus Approach to Deriving N-gram Statistics from large Scale Corpora. *Proceedings of the 1998 International Conference on Chinese Information Processing*, :223–229.

Nianwen Xue. 2003. Chinese Word Segmentation as Character Tagging. *Proceedings of the International Journal of Computational Linguistics and Chinese Language Processing*.

Bin Li and Xiaohe Chen. 2003. A Human-Computer Interaction Word Segmentation Method Adapting to Chinese Unknown Texts. *Journal of Chinese Information Processing*.

Hai Zhao, Chang-Ning Huang, and Mu Li 2006. An improved Chinese Word Segmentation System with Conditional Random Field.. *Proceeding of SIGHAN-5*, 162–165.