ISCAS: A Cascaded Approach for CIPS-SIGHAN Micro-Blog Word Segmentation Bakeoff 2012 Track

Bei Shi, Xianpei Han, Le Sun
Institute of Software, Chinese Academy of Sciences
HaiDian District, Beijing, China
{shibei, xianpei, sunle}@nfs.iscas.ac.cn

Abstract

The state-of-the-art Chinese word segmentation systems have achieved high performance on well-formed long document. However, the segmentation for microblog is difficult due to the noise problem and the OOV problem. In this paper, we present a Chinese Micro-Blog Segmentation system for the CIP-SIGHAN Word Segmentation Bakeoff 2012 track. The proposed system adopts a cascaded approach which contains three steps, correspondingly the preprocessing, the word segmentation and the post-processing. In the preprocessing step, the noise which contains the special characters is processed and removed. The remaining sentences are segmented in the second step. Finally, we use the dictionary to detect the OOVs which are not correctly segmented. The results show the competitive performance of our approach.

1 Introduction

In recent years, Chinese word segmentation (CWS) has a large of progress on statistical methods (Peng et al., 2004). For instance, character-based tagging method (N Xue et al., 2003) achieves great success in the second International Chinese word segmentation Bakeoff in 2005 (Low et al., 2005). And the state-of-the-art CRF-based systems have achieved great performance using the closed train set and test set. However, the segmentation performance on the web document or on the open set is still low (Huang Changing et al., 2007). Specifically, generated by different kinds of users in the daily life, the micro blogs are noisy and full of OOV (Gustavo et al, 2010). For example, for the brevity and the significance of labels, there are lots of emotion labels, URLs, abbreviations and special characters in the micro-blogs. Otherwise, due to the social property of the micro blogs, there are lots of OOVs (including names of users, stars, locations and organizations), which make it a challenge task for the segmentation of micro blogs.

In this paper, we propose a cascaded approach of Micro-Blog segmentation. Firstly, we use regex expressions to recognize the URLs, English words and Numbers. Some special characters and punctuations are used to split the sentence into pieces. Secondly, the generated components of the sentences are partitioned into smaller pieces which comprise the preliminary result using a segmentation system. Finally, we leverage quantities of dictionaries of OOVs and idioms from the network to merge the words in order to handle the words which are segmented incorrectly. Our system's final F1 score on the test set is 92.73%.

In the rest of this paper, the models and the method used in our tasks are presented in section 2. The experiments and the results are described in section 3. We will discuss the method in section 4. Finally, we give the conclusions and make prospect in the future work.

Proceedings of the Second CIPS-SIGHAN Joint Conference on Chinese Language Processing, pages 58–62,
Tianjin, China, 20-21 DEC. 2012

2 A Cascaded Approach

In this section, we describe our system in detail. The system consists of three steps: preprocessing, HMM-based segmentation (Liu Qun et al., 2004) and post-processing.

2.1 Preprocessing

As mentioned above, the contents of micro blogs is full of noise including special format words and special characters. In order to remove the noise, we preprocess the micro blog content through two steps which are demonstrated below.

Firstly, we recognize and extract the fixed format content types such as date, fraction, and decimals using the regex expressions which are shown in Table 1.

Table 1: The regex expressions for fixed format content extraction

Regex Expression	**Component**
http://[a-zA-Z0-9\/\.]*	URL
www\.[a-zA-Z0-9\/\.]*	URL
[。]+	the sequence of ‘。’
[￥]{0,}\d+\.\d+	the representation of China Yuan
\d+:\d+[:\d+]	Time
\d+%	Percentage
[\d+\.]%	Percentage
[A-Za-z0-9\-\—_０１２３４５６７８９]+	English words and numbers

Secondly, we split the remaining pieces of sentences using some special characters and punctuations which are shown in Table 2.

Table 2: The split characters

Split characters			
Space	*	/	\
[	]	《	》
(	)	=	+
|	{	}	“
#	:	!	@
?	~	☆	◆
【	】	→	◢

From Table 2, we can see that there are lots of rare characters. It is noteworthy that both the full-width and half-width characters should be used as split characters due to the users’ random input in micro blog. In this paper, most of the split characters are extracted from the format reference of the segmentations provided by the organizers.

It may be that a word will be split in-correctly by the split characters. For example, the emotion label ‘^_^’ will be split into ‘^’, ‘_’ and ‘^’. We will resolve this problem in the step of post processing step.

2.2 Segmentation

Given the split sentences, we segments them into words using two different systems: 1) The first is ChineseNLPTools, a HMM segmentation system trained with Ren Min newspaper corpora; 2) The second is a hierarchical hidden Markov model (HHMM) based system, ICTCLAS (Hua-Ping Zhang et al., 2003), which integrates Chinese word segmentation, Part-Of-Speech tagging, dis-ambulation and unknown words recognition within a uniform framework.

We observed that ICTCLAS is better on recall than ChineseNLPTools in experiments. However, The ChineseNLPTools achieves better performance on named entity reorganization and precision. In order to get better performance, we combine the results of both two tools: we first segment the text using ChineseNLPTools, then the words whose length is greater than four will be segmented again by ICTCLAS and the corresponding results will be replaced.

Because the first name and the last name of people are separated in the format reference, it is important for us to recognize the people name. We use a precision based vote strategy to determine whether a word is named entity using the results of ChineseNLPTools and ICTCLAS.

2.3 Post Processing

In the results produced through the above steps, some words (especially the OOVs) are incorrectly segmented. For example, “盛德利” will be split into “盛” and “德利”。Therefore, we introduce a post processing step which can merge the words into the correct OOVs. Besides, the reduplicated words and the negative words are handled in this step.

We observed that we can better detect the OOVs using more word dictionaries. In this paper, we use the title of Baidu Baike[1], the title of

[1] http://baike.baidu.com

Wikipedia[2], and the list of Chinese and Foreign stars as word dictionaries. We also use the hot topic words in the Feng Yun Bang[3] of Sina Micro Blog. We also use the dictionary of the frequent words of the network which is published by Sogou Labs[4]. The emotion labels will also be extracted in this step.

To merge the different segmentation candidates, we adopt the shortest-path strategy which prefers the long word. In case of the noise in the dictionary, we also filter words whose length is greater than 3 because long words in the dictionary matches will decrease the recall with a fine-grained criterion of segmentation.

After the match of strings, the reduplicated words are merged and handled by rules. Besides, the person names which are voted by the two tokenizers will be split into the first name and the last name in accord with the official format reference.

3 Empirical Results

3.1 Experiment Setup

In the CIPS-SIGHAN track, the train data set consists of 503 sentences. And we mainly do experiments on train data set for evaluating the performance of our tokenizer because of the test data set has not been published.

There are three evaluation metrics used in this bake-off task: Precision (P), Recall (R) and F1, where F1 is calculated as 2RP/(R+P).

3.2 Experimental Results

In this section, we evaluate our methods and discuss the result of each step.

3.2.1 Preprocessing

As mentioned above, we preprocess the sentences to filter out the noise text. We demonstrate the segmentation results with and without preprocessing step in Table 3, where ‘With_Pre’ denotes the tokenizer with preprocessing and vice versa.

Table 3: Results with and without preprocessing

	Precision	Recall	F1
With_Pre	**0.9367**	**0.9315**	**0.9341**
Without_Pre	0.8898	0.8811	0.8855

From Table 3, we can see that the F score has increased by about 5 percentage points. It means that the step of preprocessing is useful to the word segmentation for micro blogs. And the split characters have a very significance for the noise reduction

We believe this is because the existed tokenizers have worse performance on the words in the format of date, time and so on. Due to the diversity of micro blogs, it is of great difficulty to extract them only through segmentation.

3.2.2 Segmentation

After preprocessing, we compare three tokenizers. The CRF one which uses CRF++ is trained on the corpora of SIGHAN 2005 bakeoff. The ICTCLAS is generated by passing the longer words produced by our model to ICTCLAS. The last one stands for the tokenizer without ICTCLAS.

Table 4: Results of the three tokenizers

	Precision	Recall	F1
CRF	0.8899	0.8679	0.8787
With_ICTCLAS	0.9367	**0.9315**	**0.9341**
Without_ICTCLAS	**0.9375**	0.9233	0.9303

Table 4 shows the results of the three tokenizers. We can see the method of CRF is the worst due to domain variety between the training news document and the test micro blogs. Besides, the tokenizer with ICTCLAS has better performance on Recall and F1. It means ICTCLAS makes a contribution on the recall.

3.2.3 Post-processing

We obtain the preliminary segmentation results through the HHMM model-based segmentation, and then we detect the OOVs using different dictionaries. In order to eliminate as more errors as possible, we demonstrate how adding resources will increase the segmentation performance.

[2] http://zh.wikipedia.org
[3] http://data.weibo.com/top
[4] http://www.sogou.com/labs/dl/w.html

Figure 1: The performance using different resources

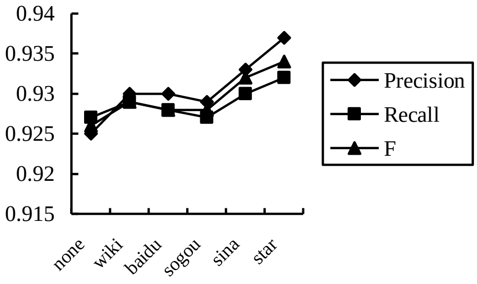

Figure 1 illustrates the quality of the dictionaries. ‘None’ stands for the system without the OOV dictionary. ‘Wiki’ stands for the import of the title of wiki. ‘Baidu’ denotes the title of Baidu Baike. ‘Sogou’ stands for the frequent words list of web published by Sogou Labs. ‘Sina’ means the frequent words that appear in micro blog frequently and ‘star’ means the list of the stars captured on the Internet. The curves decrease at the point of ‘baidu’ and ‘sogou’. It indicates the quality of ‘baidu’ and ‘sogou’ is poorer than others due to its consistency with the original micro blog segmentation. For example, the word ‘打卤面’ are merged in the dictionary while ‘打卤’ and ‘面’ are split in the corpus. The growth trend of the whole curve shows that the use of resources can improve the overall segmentation performance.

After processing the reduplicated words, negative words and quantifiers, the final segmentation performance increases 1% in F1.

3.3 Evaluation and Analysis in Test Set

In this task of micro blogs, our final results are showed in Table 5. “CS” denotes the number of the correct sentences; “PCS” denotes the percentage of the correct sentences. The first row is our result and the second row is the best result in this task.

Table 5: Final Result of the Test Set

Precision	Recall	F	CS	PCS
0.9258	0.9288	0.9273	1684	33.68%
0.946	0.9496	0.9478	2244	44.88%

Table 5 indicates that our result (0.9273) of the test set is worse than our result on the train set (0.9341). We believe this is because the resources are not sufficient for the test set.

4 Discussion

In this task, our result is slightly lower than the best performance. The reasons are as follows. First, spelling mistakes and the abbreviations of words which are common in micro blogs make the segmentation more difficult. What is more, the social property of micro blogs also increases the appearance of person names, location names, etc. Second, the quality of the dictionaries we crawl from the Internet is not as high as we expected (For example, Baidu Baike and Sogou). Third, we use the dictionaries determinedly by the shortest path, rather than probabilistically. This will make some mistake since it didn’t consider the context of the OOVs. Besides, a large number of OOVs do not exist in our dictionary because of there are not up-to-date. Finally, the criterion of segmentation of our own tokenizer is not in accordance with the official criterion. In a word, the users’ imagination and the properties of micro blogs cause difficulties on the segmentation.

5 Conclusions and Future work

In this paper, we have briefly described a cascaded approach for the Chinese word segmentation for micro blogs. A HMM model is implemented and combined with ICTCLAS. In order to solve the noise and the OOV in micro blog, we employ some special strategies. The results on training data set and test data set show that our approach is competitive. However, our method still has much improvement room to resolve the problem of OOVs in micro blog.

Acknowledgements

The work is supported by the National Natural Science Foundation of China under Grants no 90920010 and 61100152.

References

Xue, Nianwen, 2003, Chinese Word Segmentation as Character Tagging, Computational Linguistics and Chinese Language Processing. Vol.8, No.1, pp29-48

Peng, F., F. Feng, and A. McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. Proceedings of the 20th international conference on Computational Linguistics

Low, Jin Kiat et al., 2005, A Maximum Entropy Approach to Chinese Word Segmentation. Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, Jeju Island, Korea,. pp161-164

Huang Changning, HaoHai. 2007. Ten Years of Chinese word segmentation. Vol. 21, No. 3. JOURNAL OF CHINESE INFORMATION PROCESSING

Gustavo Laboreiro and Luis Sarmento. 2010. Tokenizing Micro-Blogging Messages using a Text Classification Approach. AND'10, October 26, 2010, Toronto, Ontario, Canada.

Liu Qun, Zhang Huaping, Yu Hongkui. 2004. Chinese lexical analysis using cascaded hidden Markov model. Journal of Computer Research and Development, 2004, 41(8):1421-1429

Hua-Ping Zhang, Hong-Kui Yu, De-Yi Xiong and Qun Liu. 2003. HHMM-based Chinese Lexical Analyzer ICTCLAS. SIGHAN'03 Proceedings of the second SIGHAN workshop on Chinese Language Processing - Volume 17, Pages 184-187