Improving Chinese Word Segmentation on Micro-blog Using Rich Punctuations

Longkai Zhang Li Li Zhengyan He Houfeng Wang* Ni Sun
Key Laboratory of Computational Linguistics (Peking University) Ministry of Education, China
zhlongk@qq.com, li.l@pku.edu.cn, hezhengyan.hit@gmail.com,
wanghf@pku.edu.cn,sunny.forwork@gmail.com

Abstract

Micro-blog is a new kind of medium which is short and informal. While no segmented corpus of micro-blogs is available to train Chinese word segmentation model, existing Chinese word segmentation tools cannot perform equally well as in ordinary news texts. In this paper we present an effective yet simple approach to Chinese word segmentation of micro-blog. In our approach, we incorporate punctuation information of unlabeled micro-blog data by introducing characters behind or ahead of punctuations, for they indicate the beginning or end of words. Meanwhile a self-training framework to incorporate confident instances is also used, which prove to be helpful. Experiments on micro-blog data show that our approach improves performance, especially in OOV-recall.

1 INTRODUCTION

Micro-blog (also known as tweets in English) is a new kind of broadcast medium in the form of blogging. A micro-blog differs from a traditional blog in that it is typically smaller in size. Furthermore, texts in micro-blogs tend to be informal and new words occur more frequently. These new features of micro-blogs make the Chinese Word Segmentation (CWS) models trained on the source domain, such as news corpus, fail to perform equally well when transferred to texts from micro-blogs. For example, the most widely used Chinese segmenter "ICTCLAS" yields 0.95 f-score in news corpus, only gets 0.82 f-score on micro-blog data. The poor segmentation results will hurt subsequent analysis on micro-blog text.

*Corresponding author

Manually labeling the texts of micro-blog is time consuming. Luckily, punctuations provide useful information because they are used as indicators of the end of previous sentence and the beginning of the next one, which also indicate the start and the end of a word. These "natural boundaries" appear so frequently in micro-blog texts that we can easily make good use of them. TABLE 1 shows some statistics of the news corpus vs. the micro-blogs. Besides, English letters and digits are also more than those in news corpus. They all are natural delimiters of Chinese characters and we treat them just the same as punctuations.

We propose a method to enlarge the training corpus by using punctuation information. We build a semi-supervised learning (SSL) framework which can iteratively incorporate newly labeled instances from unlabeled micro-blog data during the training process. We test our method on micro-blog texts and experiments show good results.

This paper is organized as follows. In section 1 we introduce the problem. Section 2 gives detailed description of our approach. We show the experiment and analyze the results in section 3. Section 4 gives the related works and in section 5 we conclude the whole work.

2 Our method

2.1 Punctuations

Chinese word segmentation problem might be treated as a character labeling problem which gives each character a label indicating its position in one word. To be simple, one can use label 'B' to indicate a character is the beginning of a word, and use 'N' to indicate a character is not the beginning of a word. We also use the 2-tag in our work. Other tag sets like the 'BIES' tag set are not suiteable because the puctuation information cannot decide whether a character after punctuation should be labeled as 'B' or 'S'(word with Single

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 177–182,
Sofia, Bulgaria, August 4-9 2013.

	Chinese	English	Number	Punctuation
News	85.7%	0.6%	0.7%	13.0%
micro-blog	66.3%	11.8%	2.6%	19.3%

Table 1: Percentage of Chinese, English, number, punctuation in the news corpus vs. the micro-blogs.

character).

Punctuations can serve as implicit labels for the characters before and after them. The character right after punctuations must be the first character of a word, meanwhile the character right before punctuations must be the last character of a word. An example is given in TABLE 2.

2.2 Algorithm

Our algorithm "ADD-N" is shown in TABLE 3. The initially selected character instances are those right after punctuations. By definition they are all labeled with 'B'. In this case, the number of training instances with label 'B' is increased while the number with label 'N' remains unchanged. Because of this, the model trained on this unbalanced corpus tends to be biased. This problem can become even worse when there is inexhaustible supply of texts from the target domain. We assume that labeled corpus of the source domain can be treated as a balanced reflection of different labels. Therefore we choose to estimate the balanced point by counting characters labeling 'B' and 'N' and calculate the ratio which we denote as η. We assume the enlarged corpus is also balanced if and only if the ratio of 'B' to 'N' is just the same to η of the source domain.

Our algorithm uses data from source domain to make the labels balanced. When enlarging corpus using characters behind punctuations from texts in target domain, only characters labeling 'B' are added. We randomly reuse some characters labeling 'N' from labeled data until ratio η is reached. We do not use characters ahead of punctuations, because the single-character words ahead of punctuations take the label of 'B' instead of 'N'. In summary our algorithm tackles the problem by duplicating labeled data in source domain. We denote our algorithm as "ADD-N".

We also use baseline feature templates include the features described in previous works (Sun and Xu, 2011; Sun et al., 2012). Our algorithm is not necessarily limited to a specific tagger. For simplicity and reliability, we use a simple Maximum-Entropy tagger.

3 Experiment

3.1 Data set

We evaluate our method using the data from weibo.com, which is the biggest micro-blog service in China. We use the API provided by weibo.com[1] to crawl 500,000 micro-blog texts of weibo.com, which contains 24,243,772 characters. To keep the experiment tractable, we first randomly choose 50,000 of all the texts as unlabeled data, which contain 2,420,037 characters. We manually segment 2038 randomly selected micro-blogs.We follow the segmentation standard as the PKU corpus.

In micro-blog texts, the user names and URLs have fixed format. User names start with '@', followed by Chinese characters, English letters, numbers and '_', and terminated when meeting punctuations or blanks. URLs also match fixed patterns, which are shortened using "`http://t.cn/`" plus six random English letters or numbers. Thus user names and URLs can be pre-processed separately. We follow this principle in following experiments.

We use the benchmark datasets provided by the second International Chinese Word Segmentation Bakeoff[2] as the labeled data. We choose the PKU data in our experiment because our baseline methods use the same segmentation standard.

We compare our method with three baseline methods. The first two are both famous Chinese word segmentation tools: ICTCLAS[3] and Stanford Chinese word segmenter[4], which are widely used in NLP related to word segmentation. Stanford Chinese word segmenter is a CRF-based segmentation tool and its segmentation standard is chosen as the PKU standard, which is the same to ours. ICTCLAS, on the other hand, is a HMM-based Chinese word segmenter. Another baseline is Li and Sun (2009), which also uses punctuation in their semi-supervised framework. F-score

[1] `http://open.weibo.com/wiki`
[2] `http://www.sighan.org/bakeoff2005/`
[3] `http://ictclas.org/`
[4] `http://nlp.stanford.edu/projects/chinese-nlp.shtml\#cws`

评	论	是	风	格	，	评	论	是	能	力	。
B	-	-	-	-	-	B	-	-	-	-	-
B	N	B	B	N	B	B	N	B	B	N	B

Table 2: The first line represents the original text. The second line indicates whether each character is the Beginning of sentence. The third line is the tag sequence using "BN" tag set.

ADD-N algorithm

Input: labeled data $\{(x_i, y_i)_{i-1}^{l}\}$, unlabeled data $\{x_j\}_{j=l+1}^{l+u}$.

1. Initially, let $L = \{(x_i, y_i)_{i-1}^{l}\}$ and $U = \{x_j\}_{j=l+1}^{l+u}$.
2. Label instances behind punctuations in U as 'B' and add them into L.
3. Calculate 'B', 'N' ratio η in labeled data.
4. Randomly duplicate characters whose labels are 'N' in L to make 'B'/'N'$= \eta$
5. Repeat:
   5.1 Train a classifier f from L using supervised learning.
   5.2 Apply f to tag the unlabeled instances in U.
   5.3 Add confident instances from U to L.

Table 3: ADD-N algorithm.

is used as the accuracy measure. The recall of out-of-vocabulary is also taken into consideration, which measures the ability of the model to correctly segment out of vocabulary words.

3.2 Main results

Method	P	R	F	OOV-R
Stanford	0.861	0.853	0.857	0.639
ICTCLAS	0.812	0.861	0.836	0.602
Li-Sun	0.707	0.820	0.760	0.734
Maxent	0.868	0.844	0.856	0.760
No-punc	0.865	0.829	0.846	0.760
No-balance	0.869	0.877	0.873	0.757
Our method	0.875	0.875	0.875	0.773

Table 4: Segmentation performance with different methods on the development data.

TABLE 4 summarizes the segmentation results. In TABLE 4, Li-Sun is the method in Li and Sun (2009). Maxent only uses the PKU data for training, with neither punctuation information nor self-training framework incorporated. The next 4 methods all require a 100 iteration of self-training. No-punc is the method that only uses self-training while no punctuation information is added. No-balance is similar to ADD N. The only difference between No-balance and ADD-N is that the former does not balance label 'B' and label 'N'.

The comparison of Maxent and No-punctuation shows that naively adding confident unlabeled instances does not guarantee to improve performance. The writing style and word formation of the source domain is different from target domain. When segmenting texts of the target domain using models trained on source domain, the performance will be hurt with more false segmented instances added into the training set.

The comparison of Maxent, No-balance and ADD-N shows that considering punctuation as well as self-training does improve performance. Both the f-score and OOV-recall increase. By comparing No-balance and ADD-N alone we can find that we achieve relatively high f-score if we ignore tag balance issue, while slightly hurt the OOV-Recall. However, considering it will improve OOV-Recall by about $+1.6\%$ and the f-score $+0.2\%$.

We also experimented on different size of unlabeled data to evaluate the performance when adding unlabeled target domain data. TABLE 5 shows different f-scores and OOV-Recalls on different unlabeled data set.

We note that when the number of texts changes from 0 to 50,000, the f-score and OOV both are improved. However, when unlabeled data changes to 200,000, the performance is a bit decreased, while still better than not using unlabeled data. This result comes from the fact that the method 'ADD-N' only uses characters behind punctua-

Size	P	R	F	OOV-R
0	0.864	0.846	0.855	0.754
10000	0.872	0.869	0.871	0.765
50000	0.875	0.875	0.875	0.773
100000	0.874	0.879	0.876	0.772
200000	0.865	0.865	0.865	0.759

Table 5: Segmentation performance with different size of unlabeled data

tions from target domain. Taking more texts into consideration means selecting more characters labeling 'N' from source domain to simulate those in target domain. If too many 'N's are introduced, the training data will be biased against the true distribution of target domain.

3.3 Characters ahead of punctuations

In the "BN" tagging method mentioned above, we incorporate characters after punctuations from texts in micro-blog to enlarge training set.We also try an opposite approach, "EN" tag, which uses 'E' to represent "End of word", and 'N' to represent "Not the end of word". In this contrasting method, we only use characters just ahead of punctuations. We find that the two methods show similar results. Experiment results with ADD-N are shown in TABLE 6 .

Unlabeled Data size	"BN" tag		"EN" tag	
	F	OOV-R	F	OOV-R
50000	0.875	0.773	0.870	0.763

Table 6: Comparison of BN and EN.

4 Related Work

Recent studies show that character sequence labeling is an effective formulation of Chinese word segmentation (Low et al., 2005; Zhao et al., 2006a,b; Chen et al., 2006; Xue, 2003). These supervised methods show good results, however, are unable to incorporate information from new domain, where OOV problem is a big challenge for the research community. On the other hand unsupervised word segmentation Peng and Schuurmans (2001); Goldwater et al. (2006); Jin and Tanaka-Ishii (2006); Feng et al. (2004); Maosong et al. (1998) takes advantage of the huge amount of raw text to solve Chinese word segmentation problems. However, they usually are less accurate and more complicated than supervised ones.

Meanwhile semi-supervised methods have been applied into NLP applications. Bickel et al. (2007) learns a scaling factor from data of source domain and use the distribution to resemble target domain distribution. Wu et al. (2009) uses a Domain adaptive bootstrapping (DAB) framework, which shows good results on Named Entity Recognition. Similar semi-supervised applications include Shen et al. (2004); Daumé III and Marcu (2006); Jiang and Zhai (2007); Weinberger et al. (2006). Besides, Sun and Xu (2011) uses a sequence labeling framework, while unsupervised statistics are used as discrete features in their model, which prove to be effective in Chinese word segmentation.

There are previous works using punctuations as implicit annotations. Riley (1989) uses it in sentence boundary detection. Li and Sun (2009) proposed a compromising solution to by using a classifier to select the most confident characters. We do not follow this approach because the initial errors will dramatically harm the performance. Instead, we only add the characters after punctuations which are sure to be the beginning of words (which means labeling 'B') into our training set. Sun and Xu (2011) uses punctuation information as discrete feature in a sequence labeling framework, which shows improvement compared to the pure sequence labeling approach. Our method is different from theirs. We use characters after punctuations directly.

5 Conclusion

In this paper we have presented an effective yet simple approach to Chinese word segmentation on micro-blog texts. In our approach, punctuation information of unlabeled micro-blog data is used, as well as a self-training framework to incorporate confident instances. Experiments show that our approach improves performance, especially in OOV-recall. Both the punctuation information and the self-training phase contribute to this improvement.

Acknowledgments

This research was partly supported by National High Technology Research and Development Program of China (863 Program) (No. 2012AA011101), National Natural Science Foundation of China (No.91024009) and Major National Social Science Fund of China(No. 12&ZD227).

References

Bickel, S., Brückner, M., and Scheffer, T. (2007). Discriminative learning for differing training and test distributions. In *Proceedings of the 24th international conference on Machine learning*, pages 81–88. ACM.

Chen, W., Zhang, Y., and Isahara, H. (2006). Chinese named entity recognition with conditional random fields. In *5th SIGHAN Workshop on Chinese Language Processing, Australia*.

Daumé III, H. and Marcu, D. (2006). Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26(1):101–126.

Feng, H., Chen, K., Deng, X., and Zheng, W. (2004). Accessor variety criteria for chinese word extraction. *Computational Linguistics*, 30(1):75–93.

Goldwater, S., Griffiths, T., and Johnson, M. (2006). Contextual dependencies in unsupervised word segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 673–680. Association for Computational Linguistics.

Jiang, J. and Zhai, C. (2007). Instance weighting for domain adaptation in nlp. In *Annual Meeting-Association For Computational Linguistics*, volume 45, page 264.

Jin, Z. and Tanaka-Ishii, K. (2006). Unsupervised segmentation of chinese text by use of branching entropy. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 428–435. Association for Computational Linguistics.

Li, Z. and Sun, M. (2009). Punctuation as implicit annotations for chinese word segmentation. *Computational Linguistics*, 35(4):505–512.

Low, J., Ng, H., and Guo, W. (2005). A maximum entropy approach to chinese word segmentation. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, volume 164. Jeju Island, Korea.

Maosong, S., Dayang, S., and Tsou, B. (1998). Chinese word segmentation without using lexicon and hand-crafted training data. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pages 1265–1271. Association for Computational Linguistics.

Pan, S. and Yang, Q. (2010). A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359.

Peng, F. and Schuurmans, D. (2001). Self-supervised chinese word segmentation. *Advances in Intelligent Data Analysis*, pages 238–247.

Riley, M. (1989). Some applications of tree-based modelling to speech and language. In *Proceedings of the workshop on Speech and Natural Language*, pages 339–352. Association for Computational Linguistics.

Shen, D., Zhang, J., Su, J., Zhou, G., and Tan, C. (2004). Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 589. Association for Computational Linguistics.

Sun, W. and Xu, J. (2011). Enhancing chinese word segmentation using unlabeled data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 970–979. Association for Computational Linguistics.

Sun, X., Wang, H., and Li, W. (2012). Fast online training with frequency-adaptive learning rates for chinese word segmentation and new word detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 253–262, Jeju Island, Korea. Association for Computational Linguistics.

Weinberger, K., Blitzer, J., and Saul, L. (2006). Distance metric learning for large margin nearest neighbor classification. In *In NIPS*. Citeseer.

Wu, D., Lee, W., Ye, N., and Chieu, H. (2009). Domain adaptive bootstrapping for named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1523–1532. Association for Computational Linguistics.

Xue, N. (2003). Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, 8(1):29–48.

Zhao, H., Huang, C., and Li, M. (2006a). An improved chinese word segmentation system with

conditional random field. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, volume 117. Sydney: July.

Zhao, H., Huang, C., Li, M., and Lu, B. (2006b). Effective tag set selection in chinese word segmentation via conditional random field modeling. In *Proceedings of PACLIC*, volume 20, pages 87–94.