Chinese Word Segmentation with Character Abstraction

Le Tian, Xipeng Qiu*, and Xuanjing Huang

School of Computer Science, Fudan University, China

Abstract. Chinese word segmentation is an important and necessary problem to analyze Chinese texts. In this paper, we focus on the primary challenges in Chinese word segmentation: low accuracy of out-of-vocabulary word. To resolve this difficult problems, we group the "similar" characters to generate more abstract representation. Experimental results show that character abstraction yields a significant relative error reduction of 24.83% in average over the state-of-the-art baseline.

1 Introduction

Although words are the basic language units in Chinese, Chinese sentences consist of the continuous sequence of characters (called Hanzi) without no space between words. Therefore, word segmentation is a necessary initial step to process the Chinese language. Previous research shows that word segmentation models trained on labeled data are reasonably accurate.

Currently, the state-of-art Chinese word segmentation (CWS) methods are mostly based on sequence labeling algorithm with word-based or character-based features[16, 12, 14, 1]. These methods use the discriminative model with millions of overlapping binary features.

Recent works have tended to be "feature engineering" by trying various well-designed features to obtain the best performance. Intuitively, the complex features can give more accurate prediction than simple features, and these methods often perform remarkably well. However, they still suffer from the out-of-vocabulary (OOV) words (namely, unknown words) problem. Although the accuracy of OOV can be improved greatly by the character-based methods [16], it is still significantly lower than the accuracy of in-vocabulary (IV) words.

To deal with this problem, we wish to merge the characters, which are used in paradigmatical similar way, into abstract representation.

According to our statistics, the distribution of the characters is very skew and is subject to Zipf's law. As reported in [18], though modern Chinese character sets normally include about 10,000-20,000 characters, most of them are rarely used in everyday life. Typically, 2,500 most used Chinese characters can cover 97.97% text, while 3,500 characters can cover 99.48% text.

* Corresponding author, Email: xpqiu@fudan.edu.cn

The sparsity has a large potential to compress the space of the characters in an abstraction way, which can also bridge the gap between high and low frequency characters.

Although, there are some works [10] to use character clustering, such as Brown algorithm [2]. However, Brown algorithm classifies all the same characters into a single cluster, but Chinese characters may have many senses, hard clustering algorithm such as the Brown algorithm may not be able to deal with multiple senses gracefully. Moreover, Brown algorithm generally tends to cluster the characters which significantly occur together in text. The characters in same cluster have syntagmatical similarity, not paradigmatical similarity. [10] also reports that the features with these clusters do not improve performance on CWS.

In this paper, we propose a Chinese word segmentation method with abstraction on character levels. In Chinese, some characters are semantically or paradigmatically similar. We abstract characters into their semantic concept space. We propose a semi-supervised k-means clustering method to cluster the similar characters to the same class according their context. The map function is learned from large-scale raw texts, which can capture paradigmatic similarity among characters. Our approach yields a relative error reduction of 24.83% and an improvement of OOV recall of 34.92% in average with character abstraction over the baseline.

The rest of the paper is organized as follows: We first introduce the related works in section 2, then we describe the background of character-based word segmentation in section 3. Section 4 presents our character abstraction method. The experimental results are manifested in section 5. Finally, we conclude our work in section 6.

2 Related Works

Character abstraction is very similar to word cluster in English.

[10] uses character clustering features derived using the Brown algorithm [2] and finds that they do not improve performance on CWS. One problem might be that Chinese characters have many more senses than English words, so a hard clustering algorithm such as the Brown algorithm may not be able to deal with multiple senses gracefully.

[9] use word clustering features from a soft word clustering algorithm which can improve performance of CWS.

More broadly, [15] evaluate Brown clusters, word embeddings [4], and HLBL [11] embeddings of words on other sequence labeling tasks (Named Entity Recognition and chunking) and find that each of the three word representations improves the accuracy of these baselines.

3 Discriminative Character-based Word Segmentation

We use discriminative character-based sequence labeling for word segmentation. Each character is labeled as one of {B, I, E, S} to indicate the segmentation. {B, I, E} represent *Begin*, *Inside*, *End* of a multi-character segmentation respectively, and S represents a *Single* character segmentation.

Sequence labeling is the task of assigning labels $\mathbf{y} = y_1, \ldots, y_n$ to an input sequence $\mathbf{x} = x_1, \ldots, x_n$. Given a sample $\mathbf{x}$, we define the feature $\Phi(\mathbf{x}, \mathbf{y})$. Thus, we can label $\mathbf{x}$ with a score function,

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} S(\mathbf{w}, \Phi(\mathbf{x}, \mathbf{y})), \tag{1}$$

where $\mathbf{w}$ is the parameter of score function $S(\cdot)$. The feature vector $\Phi(\mathbf{x}, \mathbf{y})$ consists of lots of overlapping features, which is the chief benefit of discriminative model. We use online Passive-Aggressive (PA) algorithm [5, 6] to train the model parameters. Following [3], the average strategy is used to avoid the overfitting problem.

4 Character Abstraction: Learning Character Semantic Concepts

To bridge the gap between high and low frequency character, we first map the characters with same semantic concepts into a single cluster. Different with English letters, Chinese characters are associated with full or partial semantic concepts. For example, the characters “鸡 (chickens)”, “鸭 (ducks)” and “鹅 (geese)” have the same concept “fowl”. They are used in the same way to compose words with other characters, such as “头 (head)”, “爪 (feet)” and “肉 (meat)”. Since characters that appear in similar contexts (especially surrounding words) tend to have similar meanings, we can use clustering technologies to find the semantic concepts from large-scale texts.

The Chinese character clustering is similar to English word clustering. Both of them partitions sets of words/characters into subsets of semantically similar words/characters.

Brown cluster algorithm [2] is a popular algorithm of word cluster which derives a hierarchical clustering of words from unlabeled data.

Table 1 shows the top clusters derived with Brown algorithm. Besides “又/却”, “刘/杨” and “八/七”, the other clusters are not what we expected. The characters in each cluster are often collocation relations. These clusters are helpful for other NLP tasks, such as text classification, but may be misleading in CWS.

4.1 Semi-supervised K-means Cluster

To avoid the shortcomings of Brown clustering algorithm, we propose a semi-supervised K-means clustering method to map each character to its corresponding concepts based on its context.

Table 1. Top Clusters derived with Brown algorithm. (Each column represents a cluster.)

编	康	引	矿	阿	截	汉	又	喜	微	刘	八	圣	必	经	国	北	们	2	资	公
辑	健	吸	煤	伊	甚	武	却	欢	博	杨	七	诞	须	济	中	京	我	月	投	司

Different with unsupervised clustering methods, we first use HowNet Knowledge Database[7] as a initial guide for semantic concepts, then we use k-means algorithm to cluster on large-scale unlabeled texts.

HowNet gives the means not only for each word but also for each Chinese character. We extract all single characters and the corresponding semantic concepts from HowNet and categorize them by their semantic concepts. Each category represents a different semantic concept or meaning. There are $7,117$ characters and $3,666$ categories in total. $2,979$ characters belong to more than one category. Among them, "打" has the most meanings and belongs to 32 different categories, such as "dozen", "draw", "beat", "build", "call" and so on.

The detailed statistics are shown in Table 2.

Table 2. Categories of Characters from HowNet

Number of Characters	7,117
Number of Categories	3,666
Average Number of Characters per Category	3.99
Average Number Categories of per Character	2.06

Although each character can belong to more than one category, it can has one meaning in certain context. So we need map different occurrences of a character to different categories based on their different contexts. We use k-means algorithm [8] to automatically learn the map function from large scale texts.

We set the number of clusters to $3,666$, which is same to the number of semantic categories defined in HowNet.

The initial center for each cluster m_i is calculated by

$$m_i = \frac{\sum_x \sum_{x \in CH_i} \mathbf{f}(x)}{\sum_x \sum_{x \in CH_i} \mathbf{1}} \tag{2}$$

where x is every occurrence of character and $\mathbf{f}(x)$ means the context feature of x; CH_i represents the i^{th} category defined in HowNet. All the feature vectors are extracted from unlabeled data.

In order to better consider the context information, we use the previous, succeeding and the union of them as features. For example: the features of the character “鸡” in sequence “吃鸡肉” will be {-1: 吃, 1: 肉, 吃肉}.

Then we re-assign each occurrence of character to the nearest cluster. The distance we used here is Euclidean distance. The cluster center will be updated when it changes. We make a restriction that the assigned cluster for each character must be one of its categories defined in HowNet.

We use large scale unlabeled corpus from web pages collected by Sogou[1] to learn the cluster centers. The corpus contains $1,060,471,497$ characters and has $9,851$ unique characters. For the characters which are not included in HowNet, we classify them into “unknown” category.

When using in CWS, we find the category for each character x by

$$c = \arg\max_i ||f(x) - m_i||^2 \tag{3}$$

Table 3 shows some examples for character abstraction. When we represent the character with its category, we can avoid the problem of data sparsity on some level.

Table 3. Examples for Character Abstraction

Abstraction Representation	Sequences of Characters
C_{Animal}-$C_{BodyPart}$	猪头，狗头，鸡脚
C_{Number}-C_{Unit}	5 毫，五尺，3 寸，9 码，壹分，7 里，八米
$C_{Surname}$	覃 (经理)，温 (总理)，冯 (先生)，$\cdots$

5 Experiments

5.1 Dataset

All our experiments are conducted on the corpora provided by CIPS-SIGHAN-2010[17]. This dataset is well known and widely adopted. The training corpus which contains one month data of the People’s Daily in 1998 was provided by Peking University. There are four domains in the testing data: Literature (L), Computer (C), Medicine (M) and Finance (F). One main reason we use these corpora is that it addresses the ability of word segmentation for out-of-domain text.

We implement our system based on FudanNLP [13], a toolkit for Chinese natural language processing.

We first evaluate the performance with our character abstraction method.

[1] http://www.sogou.com/labs/dl/ca.html

Table 4. Traditional Feature Templates

$C_i, T_0 (i = -2, -1, 0, 1, 2)$
$C_i, C_j, T_0 (i, j = -2, -1, 0, 1, 2 \text{ and } i \neq j)$
C_{-1}, C_0, C_1, T_0
T_{-1}, T_0

Table 5. Performances of Different Methods

	Methods	**R**	**P**	**F1**	R_{OOV}	R_{IV}
L	B	0.905	0.911	0.908	0.540	0.932
	B+C	0.905	0.915	0.910	0.546	0.932
	B+CA	0.913	0.921	0.917	0.618	0.935
C	B	0.864	0.784	0.822	0.388	0.950
	B+C	0.881	0.907	0.894	0.520	0.946
	B+CA	0.914	0.921	0.918	0.704	0.952
M	B	0.899	0.894	0.897	0.594	0.937
	B+C	0.897	0.899	0.898	0.602	0.934
	B+CA	0.905	0.907	0.906	0.657	0.936
F	B	0.909	0.905	0.907	0.506	0.948
	B+C	0.911	0.921	0.916	0.528	0.948
	B+CA	0.930	0.934	0.932	0.674	0.955

1. **B**: The baseline method. We use usually the commonly used features in CWS. The form of features is shown in Table 4, where C represents a Chinese character, and T represents the character-based tag. The subscript i indicates its position related to the current character.
2. **B+C**: Besides the features used in baseline method, we use the character type features directly extracted from HowNet. For the character belonging to multiple types, we use all its types directly. This method can be regarded as the manually character clustering and is more accurate than Brown algorithm.
3. **B+CA**: Besides the features used in baseline method, we use our proposed character abstract features. The reason we use the character features is that there are some culture-specific words, common saying or idioms. These words is used regardless of character type, such as “不管三七二十一 (regardless of the consequence)”, “由此可见 (thus it can be seen)”, etc.

The results are shown in 5, which shows the information of character type can improve the performances. While the improvement is very limited to simply use these information **B+C**, our method (**B+CA**) achieves large improvements on the baseline (**B**) and yields the relative error reductions of 9.78%, 53.93%, 8.74% and 26.88% respectively on four datasets. The average relative error reduction is 24.83%. Meanwhile, the recalls OOV words are also improved by 14.44%, 81.44%, 10.61% and 33.20% respectively. The average improvement of OOV recalls is 34.92%.

Table 6 gives the numbers of nonzero parameters of models trained by different methods. We can see that our method (**B+CA**) uses fewer parameters than the baseline, which indicates the character abstraction can merge the characters used similarly and results to reduction of actually active features.

Table 6. Number of Nonzero Parameters

Methods	**Number**
B	2,022,396
B+C	2,683,524
B+CA	1,501,705

5.2 Analysis

We can see from the above experiments that The character abstraction can still boost the performance with less actually active features. However, we also found a number of inconsistent or irrational annotations in segmentation both in the training and the test data. For example, "建设银行 (Construction Bank)" is segmented while "中国银行 (Bank of China)" is used a word. These inconsistent or irrational annotations may have more impact for abstraction based method than character-based method because they can influence the process of feature abstraction.

6 Conclusion

In this paper, we focus on the challenges in Chinese word segmentation: low accuracy of out-of-vocabulary word. The experiments have shown that: abstract representation can improve the performance over the baseline. In future work, we would also like to investigate the other methods for character abstraction and we believed that good abstract features can boost the performance of CWS.

7 Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. This work was funded by NSFC (No.61003091 and No.61073069).

References

1. Andrew, G.: A hybrid markov/semi-markov conditional random field for sequence segmentation. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing. pp. 465–472. Association for Computational Linguistics (2006)

2. Brown, P., Desouza, P., Mercer, R., Pietra, V., Lai, J.: Class-based n-gram models of natural language. Computational linguistics 18(4), 467–479 (1992)
3. Collins, M.: Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (2002)
4. Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th international conference on Machine learning. pp. 160–167. ACM (2008)
5. Crammer, K., Singer, Y.: Ultraconservative online algorithms for multiclass problems. Journal of Machine Learning Research 3, 951–991 (2003)
6. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. Journal of Machine Learning Research 7, 551–585 (2006)
7. Dong, Z., Dong, Q.: Hownet And the Computation of Meaning. World Scientific Publishing Co., Inc. River Edge, NJ, USA (2006)
8. Duda, R., Hart, P., Stork, D.: Pattern Classification. Wiley, New York, 2nd edn. (2001)
9. Li, W., McCallum, A.: Semi-supervised sequence modeling with syntactic topic models. In: Proceedings of the National Conference on Artificial Intelligence. p. 813 (2005)
10. Liang, P.: Semi-supervised learning for natural language. Ph.D. thesis, Massachusetts Institute of Technology (2005)
11. Mnih, A., Hinton, G.: A scalable hierarchical distributed language model. Advances in neural information processing systems 21, 1081–1088 (2009)
12. Peng, F., Feng, F., McCallum, A.: Chinese segmentation and new word detection using conditional random fields. Proceedings of the 20th international conference on Computational Linguistics (2004)
13. Qiu, X., Zhang, Q., Huang, X.: FudanNLP: A toolkit for chinese natural language processing. In: Proceedings of ACL (2013)
14. Sarawagi, S., Cohen, W.: Semi-markov conditional random fields for information extraction. Advances in Neural Information Processing Systems 17, 1185–1192 (2005)
15. Turian, J., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. Urbana 51, 61801 (2010)
16. Xue, N.: Chinese word segmentation as character tagging. Computational Linguistics and Chinese Language Processing 8(1), 29–48 (2003)
17. Zhao, H., Huang, C., Li, M., Lu, B.: A unified character-based tagging framework for chinese word segmentation. ACM Transactions on Asian Language Information Processing (TALIP) 9(2), 5 (2010)
18. Zhao, H., Liu, Q.: The cips-sighan clp 2010 chinese word segmentation bakeoff. In: Proceedings of the First CPS-SIGHAN Joint Conference on Chinese Language Processing (2010)