Enhancing Chinese Word Segmentation with Character Clustering

Yijia Liu, Wanxiang Che, Ting Liu

Research Center for Social Computing and Information Retrieval
School of Computer Science and Technology
Harbin Institute of Technology, China
{yjliu,car,tliu}@ir.hit.edu.cn

Abstract. In semi-supervised learning framework, clustering has been proved a helpful feature to improve system performance in NER and other NLP tasks. However, there hasn't been any work that employs clustering in word segmentation. In this paper, we proposed a new approach to compute clusters of characters and use these results to assist a character based Chinese word segmentation system. Contextual information is considered when we perform character clustering algorithm to address character ambiguity. Experiments show our character clusters result in performance improvement. Also, we compare our clusters features with widely used mutual information (MI). When two features integrated, further improvement is achieved.

Keywords: Brown clustering, Chinese word segmentation, semi-supervised learning

1 Introduction

Chinese word segmentation is the first step of many NLP and IR tasks. Over the past years, word segmentation system's performance has been improved. However there are still some challenging problems. One of these problems is how to unearth helpful information from large scale unlabeled data and use this information to improve word segmentation system's performance. Former researchers have tried to use auto-segmented result of large scale unlabeled data[1] and statistical magnitudes like mutual information, accessory variety[2] to help the semi-supervised learning system. Performance improvement is achieved in their works.

In other tasks like NER, word clustering has been proved a helpful method to derive information from unlabeled data and improve the semi-supervised learning systems performance[3][4]. However, there hasn't been any work that applies clustering to word segmentation. The main reason is that there is no natural word boundary in Chinese. Traditional routine of clustering words cannot be applied to segmentation task directly. But, as character is the minimum unit of Chinese language, it's promising that we build clusters from character and use this character clustering information to assist word segmentation. In this

paper, we try to employ Brown clustering algorithm to build character-based clusters and embed contextual information into the character cluster. Finally we compile the clustering result into features and use this features to improve the word segmentation task. Experiments shows our character clustering results can help improving word segmentation performance.

The reminder of this paper is organized as follows. Section 2 describes the intuitive motivation and theoretical analysis of our character-based clustering method. Section 3 introduces the semi-supervised model we use to incorporate clustering results. Section 4 presents experimental results and empirical analysis. Section 5 gives some conclusion and future work.

2 Character-based Brown Clustering

2.1 Brown clustering

Data sparsity is always an issue in many NLP tasks. Capturing generality from unlabeled data is a promising way to address this issue.[4] Intuitively, character under the similar context environment tends to have similar function when compositing words. Supposing there is some criterion reflecting this similarity, we can use this criterion to help our word segmentation system. For example, in the following sentence "...期货中的做空行为..."(... the shorting in futures ...), "做空"(the shorting) is a financial term which barely occurs in newswire. While, similar context may occurs in "...管理中的违纪行为...[1]" (... the disciplinary offence...) and this context is a typical newswire. This kind of similarity provides a clue for inferring the segmentation of "做空".

In this paper, we view our clusters as *class-based bigram language model*. The *class-based bigram language model* considers the sentence as a sequence of characters and there is a hidden class behind each character in the sentence. Figure 1 illustrate the *class-based bigram language model* where c_i represent i_{th} character of the sentence and C_i is its cluster.

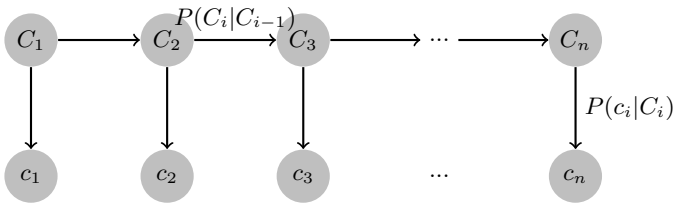

Fig. 1. Brown clustering model[4]

[1] This example occures People Daily corpus

Given a sentence $c_{1...n}$ consisting of n characters, probability of $P(c_{1...n})$ is modeled as follow:

$$\begin{aligned} P(c_{1...n}) &= P(c_1, c_2, ...c_n, C(c_1), C(c_2), ..., C(c_n)) \\ &= \prod_{i=1}^{n} P(c_i|C(c_i))P(C(c_i)|C(c_{i-1})) \end{aligned} \tag{1}$$

To maximum the likelihood of the *class-based bigram language model*, we can derive a hierarchical clustering of words with a bottom-up clustering algorithm, which is known as Brown clustering algorithm[5][4]. The input of Brown clustering is a sequence of item. The output is a binary tree, which can be represented by a string of 01.

2.2 Unigram Character Clustering

In our unigram character clustering model, we follow our model definition as mentioned in section 2.1 and the cluster of a character only depends on the character itself. As result of the unigram character clustering model, each character is allocated with a single cluster. In our experiment, sentence is split into sequence of characters and brown clustering algorithm is employed on the sequence. Table 1 illustrates some result of the unigram character clustering model. For the clustering result shown in table 1, it seems our clustering model works

character	cluster	character	cluster	character	cluster
镓	0101111110	九	00111010111	张	00110010010
铟	0101111110	七	00111010111	王	00110010010
镁	0101111110	八	00111010111	李	00110010010
锂	0101111110	六	00111010111	叶	00110010010

Table 1. clustering result, experiment are conducted on Gigawords setting number of clusters to 500

well by clustering the Chinese digit into one cluster and Chinese metal name into another cluster. However, farther analytics will cast doubt on these results' effect. Syntactical and semantical function of same Chinese character varies under different circumstance. Simply dropping the contextual environment and clustering the character into mono-clustering will introduce a lot of ambiguity. Clustering result in Table 1 also shows this problem. The Chinese character "叶" can be a family name (translated as 'Ye'), while it can also indicating part of the plant(translated as 'leaf'). When used as a family name, "叶" is usually the leading character of a word. But when used as leaf, "叶" can composite word like "树叶"(leaf), "一叶障目"(a Chinese idiom which means having ones view of the important overshadowed by the trivial) and used as middle or end of a word. In following section, our experimental result also prove unigram character clustering doesn't work well.

2.3 Bigram and Trigram Character Clustering

To settle the problems mentioned above, same character under different context circumstance should be categorized into different clusters. We incorporate contextual information by considering character's bigram and trigram. The model of $P(c_{1...n})$ changes into

$$\begin{aligned} P(c_{1...n}) &= P(c_1, c_2, ...c_n, C_1, C_2, ..., C_n) \\ &= \prod_{i=1}^{n} P(c_i c_{i-1}|C(c_i c_{i-1}))P(C(c_i c_{i-1})|C(c_{i-1}c_{i-2})) \end{aligned} \tag{2}$$

in bigram case and

$$P(c_{1...n}) = \prod_{i=1}^{n} P(c_{i+1}c_i c_{i-1}|C(c_{i+1}c_i c_{i-1}))P(C(c_{i+1}c_i c_{i-1})|C(c_i c_{i-1}c_{i-2})) \tag{3}$$

in trigram case.

Table 2 shows our bigram character clustering result. First column of the result shows that our model cluster "叶" under the environment where it's used as leading character and means leaf into same cluster. The second column capture the sentence segmentation like "...特级大师叶江川...", "...设计师赵葆常...", "...教师张百战..." [2]. In this situation, the bigram "师叶" provide a clue for segmentation. The third column cluster the rare word "做空" into a cluster of common words. Analogously, trigram model gives similar results.

character	cluster	character	cluster	character	cluster
叶脉	10001011	师赵	111010011	做空	1001101
叶片	10001011	师徐	111010011	选举	1001101
叶柄	10001011	师朱	111010011	遏制	1001101
叶子	10001011	师叶	111010011	抑制	1001101

Table 2. clustering result, experiment are conducted on Gigawords setting number of clusters to 500

3 Semi-supervised Learning Model

Previous study[1][2][6] has presented a simple yet effective semi-supervised method of incorporating information derived from large scale unlabeled data. Their method introduces new semi-supervised feature into robust machine learning model. In this paper, we follow their work and employ a conditional random fields (CRFs) model to incorporate character clustering results. This model is a character-based sequence labeling model, in which a character is labeled a tag representing the position of its position in word. We follow the work in [1] and select tagset of 6-tag style (B, B2, B3, I, E, S).

[2] All these three example is drawn from Chinese Gigawords(LDC2011T13)

3.1 Baseline Features

We employ a set of simple but widely used feature as baseline feature. The features we use are listed below.

- character unigram: c_s $(i-2 \leq s \leq i+2)$
- character bigram: $c_s c_{s+1}$ $(i-2 \leq s \leq i+1)$, $c_s c_{s+2}$ $(i-2 \leq s \leq i)$
- character trigram: $c_{s-1} c_s c_{s+1}$ $(s = i)$
- repetition of characters: is c_s equals c_{s+1} $(i-1 \leq s \leq i)$, is c_s equals c_{s+2} $(i-2 \leq s \leq i)$
- character type: is c_i an *alphabet*, *digit*, *punctuation* or *others*

3.2 Mutual Information Features

In order to compare character clustering with traditional semi-supervised feature, we follow previous work[2] and feed mutual information to our semi-supervised model. Mutual information of two character is define as,

$$MI(c_i c_{i+1}) = \log \frac{p(c_i c_{i+1})}{p(c_i)p(c_{i+1})} \tag{4}$$

For each character c_i, we compute $MI(c_{i-1}, c_i)$ and $MI(c_i, c_{i+1})$ and round them down to integer. These integer value are integrated into CRF model as a type of features.

3.3 Clustering Features

We compile character clusters result into a kind of feature. When clustering algorithm is performed over large scale unlabeled data, a lexicon indicating ngram is cluster is maintained. For each character c_i in sentence, we extract the clusters of ngram and integrate them into our CRF model as a type of features.

For different clustering models, we extract different features. The clustering features are listed below,

- For our unigram character clustering model, brown clustering results of character in a window of 5 are extracted: $brown(c_s)(i-2 \leq s \leq i+2)$
- For our bigram character clustering model, we extract $brown(c_{i-1}c_i)$ and $brown(c_i c_{i+1})$
- For our trigram character clustering model, we extract $brown(c_{s-1}c_s c_{s+1})(i-1 \leq s \leq i+1)$

Here, $brown(x)$ represents the clusters of ngram x.

Data set	# of sent.			# of words		
	train	test	dev	train	test	dev
CTB5.0	18,086	348	350	493,934	8,008	6,821
CTB6.0	23,417	2,769	2,077	641,329	81,578	59,947

Table 3. Statistic of the corpus

4 Experiments

4.1 Settings

To test the character clustering's effect on Chinese word segmentation, we select CTB5.0 and CTB6.0 as our labeled data. For these two data set, we split the data according to the recommandation in the document. Some statistic of the data is listed in Table 3.

Chinese Gigawords(LDC2011T13) is a remarkable achieve of unlabeled data, because of its huge quantity and broad coverage. Xinhua news(from 2000 to 2010) is chosen from Chinese Gigawords as unlabeled data in our experiment, which has about 500 million characters.

F1-score is used as measurement for our model. Define precision p as percentage of words that are correctly segmented in model output, and recall r as percentage of words that are correctly segmented in gold standard output. F1-score equals $\frac{2pr}{(p+r)}$.

We use a CRFs toolkit CRFSuite[7] to label sequential data. During the training pharse, stochastic gradient descent is set as training algorithm. Two parameters $feature.possible\_trainstions = 1$ and $feature.possible\_states = 1$ is configed to enable negative features.

We use Liang's implementation of Brown clustering algorithm[3] to maintain the character clusters. Algorithm's running time on an Xeon(R) 2.67GHz server is list in Table 4. We didn't maintain clustering results of 500 and 1000 clusters in trigram case, because it would consume too much time.

Model	$c = 100$	$c = 200$	$c = 500$	$c = 1000$
Bigram	3	11	73	245
Trigram	29	126	-	-

Table 4. Brown clustering algorithm's running time(hours) on Chinese Gigawords with different number of clusters. $c = x$ means number of clusters is x.

[3] `https://github.com/percyliang/brown-cluster`

4.2 Results

According to previous study, number of clusters controls the representation capicity of clustering result[8]. We conduct experiment on various settings of cluster number. Figure 2 shows our experimental result on CTB5.0 and CTB6.0. *baseline* means the CRFs model trained with baseline features. Symbol '+' means model trained both with baseline features and the new features. Experiment results shows that in bigram case, model of $c = 500$ and $c = 1000$ respectively achieve best accuracy on CTB5.0 and CTB6.0's development data. In trigram case, model of $c = 200$ perform better than that of $c = 100$ in both data set.

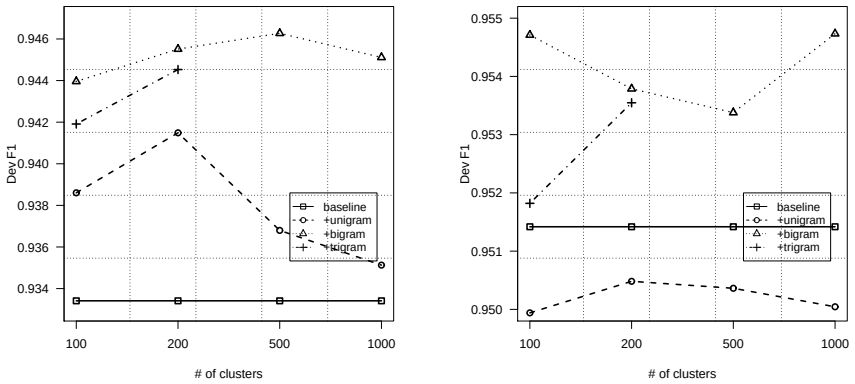

Fig. 2. Result on CTB5.0(left) and CTB6.0(right) development data

These results basically match what we expected in our former theoretical analysis. Unigram character clustering(+unigram) have almost no effect on Chinese word segmentation. However when increase the order of clustering model(+bigram,+trigram), increasement in F1-score is achieved. This result proves character clustering model considering contextual information is effective on Chinese word segmentation. Theoretically, trigram clustering eploits more contextual information and is expected to have better performance than bigram clustering. However, in both the CTB5.0 and CTB6.0, performance of model with trigram(+trigram) clusters is slightly lower than bigram model(+bigram). One reason maybe that trigram clustering introduce much noise due to exponential increased vocabulary size. Another reason for this may result from the limited number of clusters in trigram model. It takes more than 5 days to compute trigram brown clustering result of 200-clusters and it takes more of the 500-clusters cases.

In our bigram character clustering experiments, there is no significant improvment when we increase the number of clusters. But in trigram experiment, performance improvement is achieved when we transfer from 100-clusters to 200-clusters. Generally, we can conclude that fine-grained clusters help the word segmentation more.

After maintaining best c on development data, we conduct experiments on test data with best c configuration. Experiment result is show in Tabel 5. From this table, we can see our method outperform the baseline model. Significance tests between our method and baseline model also demostrate that the improvements of our cluster features(+bigram,+trigram) is significant with $p - value < 10^{-4}$.

Model	CTB5.0			CTB6.0		
	P	R	F	P	R	F
Baseline	0.9652	0.9733	0.9692	0.9478	0.9433	0.9455
+bigram	0.9699	0.9781	0.9740	0.9523	0.9504	0.9514
+trigram	0.9700	0.9760	0.9730	0.9506	0.9481	0.9494
+MI	0.9729	0.9804	0.9766	0.9530	0.9516	0.9523
+MI+bigram	**0.9738**	**0.9808**	**0.9773**	**0.9533**	**0.9539**	**0.9536**

Table 5. Result of different models on CTB5.0 and CTB6.0 test data. bigram and trigram model is configed with best c in former experiments. $c = 500$ is set in the *+MI+bigram* model.

MI is a standard measurement of the association between character. Characters with stronger association is more likely to combine and composite a word. We compare our character clustering model that is configed with best c with model incorporated with MI. Table 5 shows the comparison results. In our experiment, we have seen that system incorporate with mutual information outperforms the character clustering model.

Although perform well, one limitation of MI is that it's computed only with local information between two characters. The character clustering which considering global information of a sentence makes a good complement of this limitation. In our experiments, we have seen further improvement on performance when two methods are combined.

5 Conclusion and Future Work

In this paper, we propose a method of building clusters from Chinese character. Contextual information is considered when we perform character clustering algorithm to address character ambiguity. Experimental result shows our character clustering result can help improve word segmentation performance.

In future, we will try to apply this method to some cross domain corpus. Also, we will try to use the character clusters to help other character-based NLP task like character-based Chinese parsing model.

Acknowledgments

This work was supported by National Natural Science Foundation of China (NSFC) via grant 61133012, the National "863" Major Projects via grant 2011AA01A207, and the National "863" Leading Technology Research Project via grant 2012AA011102.

References

1. Wang, Y., Jun' ichi Kazama, Y., Chen, W., Zhang, Y., Torisawa, K.: Improving chinese word segmentation and pos tagging with semi-supervised methods using large auto-analyzed data. In: Proceedings of the Fifth International Joint Conference on Natural Language Processing (IJCNLP-2011). (2011)
2. Sun, W., Xu, J.: Enhancing chinese word segmentation using unlabeled data. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2011) 970–979
3. Miller, S., Guinness, J., Zamanian, A.: Name tagging with word clusters and discriminative training. In: Proceedings of HLT-NAACL. Volume 4., Citeseer (2004)
4. Liang, P.: Semi-supervised learning for natural language. PhD thesis, Massachusetts Institute of Technology (2005)
5. Brown, P.F., Desouza, P.V., Mercer, R.L., Pietra, V.J.D., Lai, J.C.: Class-based n-gram models of natural language. Computational linguistics **18**(4) (1992) 467–479
6. Chen, W., Kazama, J., Uchimoto, K., Torisawa, K.: Improving dependency parsing with subtrees from auto-parsed data. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2, Association for Computational Linguistics (2009) 570–579
7. Okazaki, N.: Crfsuite: a fast implementation of conditional random fields (crfs) (2007)
8. Turian, J., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics (2010) 384–394