

Improving Neural Chinese Word Segmentation Using Unlabeled Data

Yanna Zhang¹, Jinan Xu², Guoyi Miao³, Yufeng Chen⁴ and Yujie Zhang⁵

^{1, 2, 3, 4, 5}School of Computer and Information Technology, Beijing Jiaotong University Shangyuancun No.3, Beijing, P.R.China

{¹yannazhang, ²jaxu, ³gymiao, ⁴chenyf, ⁵yjzhang}@bjtu.edu.cn

Abstract. Supervised word segmentation heavily relies on large-scale and high quality labeled data. However, building such a corpus is difficult, especially with respect to domain specific data. In this paper, we propose a novel semi-supervised Chinese word segmentation (CWS) method. Specifically, we seek to select more useful sample sentences from the large-scale unlabeled sentences to extend the training data, by means of a sampling strategy that uses character-based semantic similarity. The presented similarity algorithm is used to calculate the similarity between unlabeled sentences and the training data, which can help select helpful sample sentences from unlabeled data. In addition, we integrate an attention mechanism into our word segmentation model to focus on available contextual information. Experiments on PKU, MSR and Weibo benchmark data sets show that our method outperforms the previous neural network models and state-of-the-art methods.

1. Introduction

Chinese word segmentation (CWS) is a preliminary and important task for many Chinese natural language processing (NLP) tasks, such as machine translations (MT), question answering systems (QA) and named entity recognition (NER). Neural word segmentation has shown promising progress in recent years [1] [2] [3]. However, neural word segmentation, mainly trained by supervised learning, relies on a large-scale labeled data which is usually expensive and tends to be of a limited amount. Lack of labeled data makes it very difficult to capture useful semantic information from contexts to improve the performance of word segmentation model.

For solving the mentioned problem above, researchers have tried to use large-scale unlabeled data to improve the supervised word segmentation. One intuitive idea is to adopt semi-supervised self-training method. It exploits a small amount of labeled data to train an initial word segmentation model, which is used to annotate unlabeled data. The annotation results are added to training data set to update parameters of word segmentation model in the next iteration. However, there still remains some challenges, for example, some existing semi-supervised word segmentation methods cannot pick out samples containing rich features. The above issues have greatly limited the effect of word segmentation.

In this paper, we propose a novel semi-supervised word segmentation method by leveraging semantic similarity. Our work is mainly about of two aspects to select samples: The uncertainty of annotation and the semantic similarity between unlabeled sentences and training data. Moreover, we adopt an attention-based natural word segmentation to dynamically focus on the character-based n-



gram contextual information. Experiments show that our approach can select samples containing more useful features and achieve significant improvements on Weibo, PKU and MSR datasets.

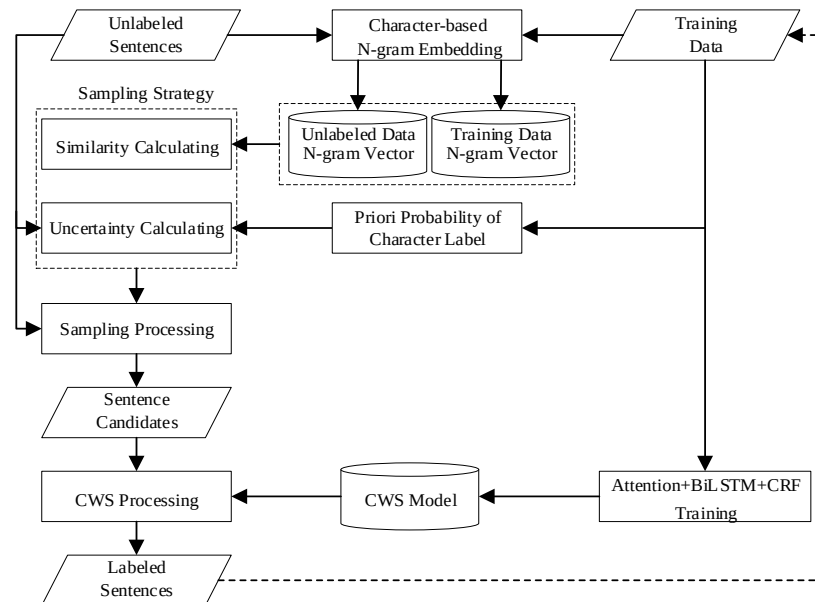


Figure 1. Framework of Semi-supervised Learning for CWS.

The contributions of this paper could be summarized as follows.

- The semantic similarity is first introduced to semi-supervised learning for enhancing CWS, which is effective to measure the diversity of samples.
- For improving the performance of CWS, we integrate an attention mechanism into our CWS model, which is used to dynamically focus on the contextual information and capture useful features (Section 3.2.).
- We propose a new semantic similarity calculation method with the help of character-based n-gram embedding, which incorporates rich contextual information (Section 3.3.).

2. Related Work

Semi-supervised learning is widely used in word segmentation tasks. Zheng et al. [4] and Sun et al. [5] experimented with deriving some statistics-based features such as mutual information, accessor variety and punctuation variety from unlabeled data, and add them to supervised discriminative training. Yang et al. [6] adopted partial-label learning with conditional random fields to make use of the valuable knowledge contained by unlabeled data. Melamud et al. [7] presented a neural model for learning a generic context embedding function from large corpora and applied it to several NLP tasks. Furthermore, Peters et al. [8] directly used large-scale unlabeled corpus to pre-train language models. By using language models, character-level knowledge can be integrated into context vectors and the potential meaning of character knowledge can be well encoded.

One of semi-supervised learning methods in NLP is bootstrapping, which generate more training instances by automatically labeling large-scale data. This is also the main idea of our method. Yang et al. [9], Zeng et al. [10] and Zhang et al. [11] experimented with co-training for semi-supervised Chinese word segmentation. Huang et al. [12] adopted self-training to increasing the quality of the automatically parsed data. Liu et al. [13] combined self-training and character clustering to solve the problem of domain adaptability. Inspired by the previous work, we propose a new and novel sampling strategy for self-training.

3. Methodology

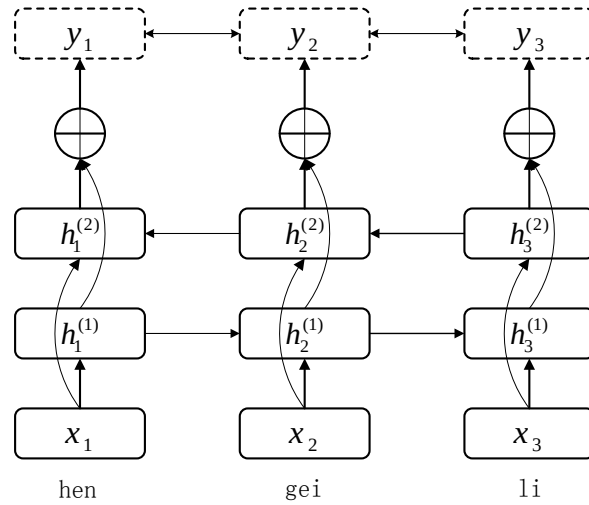


Figure 2. BiLSTM+CRF architecture for CWS. Character embeddings are given as input; a bidirectional LSTM produces context-dependent representations; the information is passed through a hidden layer and an output layer. The outputs are confidence scores for CRFs.

3.1. Semi-supervised Learning for CWS

Semi-supervised self-training learning can be regarded as a learning method which continuously increases training data. Our method consists of two main stages. In the first stage, we train an initial segmentation model (Attention+BiLSTM+CRF) with a small amount of labeled data, which is used to annotate unlabeled samples (See section 3.2.). In the second stage, we obtain samples from large-scale unlabeled sentences according to sampling strategy (See section 3.3). These samples after annotated are added to training data for the next iteration. We can obtain the optimal word segmentation model until the iterations converge. Figure 1 illustrates the framework of our semi-supervised learning for CWS.

3.2. The Initial CWS model

3.2.1. BiLSTM+CRF Architecture for CWS. CWS task is usually solved by character-level sequence labeling algorithm. Specifically, each character in a sentence is labeled as one of $L = \{B, M, E, S\}$, indicating the begin, middle, end of a word, or a word with a single character. For a given sentence $X = (x_1, x_2, \dots, x_n)$ containing n characters, the aim of CWS task is to predict label sequence $y^* = (y_1, y_2, \dots, y_n)$. The BiLSTM+CRF architecture (our baseline) for CWS is illustrated in Figure 2.

Similar to other methods using neural networks, the first step of the BiLSTM+CRF based CWS method is to represent characters in distributed vectors. Formally, we lookup embedding vector from embedding matrix for each character x_i as $e_i \in R^{d_e}$, where d_e is a hyper-parameter indicating the size of character embedding.

In order to incorporate information from both sides of the sequence, we use bidirectional Long Short-Term Memory (BiLSTM) as feature layers. The update of each BiLSTM unit can be described as follows:

$$h_i = \vec{h}_i \oplus \overleftarrow{h}_i \quad (1)$$

where $\vec{h}_i$ and $\overleftarrow{h}_i$ are the hidden states at position i of the forward and backward LSTMs respectively; $\oplus$ is concatenation operation.

Following Lample et al. [14], we employ conditional random fields (CRF) layer to inference labels, which is beneficial to consider the dependencies of adjacent labels. For example, a B (begin) label should be only followed by a M (middle) label or E (end) label. Given that y is a label sequence $y_1, y_2, \dots, y_n$, then the CRF score for this sequence can be calculated as:

$$s(X, y) = \sum_{i=0}^n A_{y_i, y_{i+1}} + \sum_{i=1}^n P_{i, y_i} \quad (2)$$

where A is a matrix of transition scores such that $A_{i,j}$ represents the score of a transition from the tag i to tag j . y_0 and y_n are the start and end tags of a sentence, that we add to the set of possible tags. P is the fractional matrix of the BiLSTM network's output. $P_{i,j}$ corresponds to the j -th label of the i -th word in a sentence.

The output from the model is the tagging sequence with the largest score $s(y)$. A softmax over all possible tag sequences yields a probability for the sequence y :

$$p(y|X) = \frac{e^{s(X,y)}}{\sum_{\tilde{y} \in Y_X} e^{s(X,\tilde{y})}} \quad (3)$$

Finally, we directly maximize the log-probability of the correct tag sequence:

$$\log(p(y|X)) = s(X, y) - \log\left(\sum_{\tilde{y} \in Y_X} e^{s(X,\tilde{y})}\right) = s(X, y) - \log \text{add}_{\tilde{y} \in Y_X}(X, \tilde{y}) \quad (4)$$

While decoding, the output sequence we predict will have the highest score given by the following:

$$y^* = \operatorname{argmax}_{\tilde{y} \in Y_X} s(X, \tilde{y}) \quad (5)$$

We use the Viterbi algorithm to find the optimal labeled sequence during training.

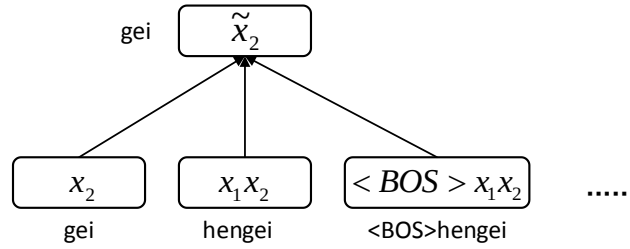


Figure 3. The attention mechanism. If the character at the current moment is “gei”, the attention mechanism gives attention to the n -gram contextual embedding of this character.

3.2.2. Attention+BiLSTM+CRF Architecture for CWS. It is necessary to use an empirically good segmentation model in the process of semi-supervised word segmentation. Therefore, we improved the performance of baseline system by integrating an attention mechanism. The attention mechanism is illustrated in Figure 3. During training, the attention mechanism selectively pays attention to the character-based n -gram contextual embeddings. The input of the BiLSTM+CRF architecture computed as a weighted sum of the character-based n -gram contextual embeddings:

$$\tilde{x} = \sum_{j=1}^N \alpha_{ij} x_j \quad (6)$$

The weight of each annotation α_{ij} is computed by:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^T \exp(e_{ik})} \quad (7)$$

where

$$e_{ij} = a(x_j) \quad (8)$$

here, x_j represents the character-based j -gram contextual embeddings, function $a(\cdot)$ is tanh function. α_{ij} is trainable parameter which denotes weight, $\tilde{x}$ denotes the output of the attention mechanism.

Because of the sparsity of the n-gram contextual embeddings of characters, here we set N to 3. Specifically, we only focus on the unigram, bi-gram and tri-gram contextual embeddings of characters. In order to get these contextual embeddings, we first deal with the sentence. After processing, this sentence “woshizhongguoren.” (I am Chinese.) will be “wo shi zhong guo ren .”, “<BOS> wo woshi shizhong zhongguo guoren ren. .<EOS>” and “<BOS0><BOS>wo <BOS>woshi woshizhong shizhongguo zhongguoren guoren. ren.<EOS> .<EOS><EOS0>” respectively. Then we train these vectors readily using word2vec [15] toolkit on the resulting n-gram sequences.

3.3. Sampling Strategy

3.3.1. Similarity between unlabeled sentences and training data. To extract useful samples from the large-scale unlabeled corpus, we propose to calculate the character-based semantic similarity between unlabeled sentences and training set sentences. In general, the more a sample sentence differs from the training set sentences, the more semantic and structural features it contains. Thus we seek to select sample sentences that are very different from training set sentences and they can then be easily deduced from the similarity between unlabeled sentences and training set sentences. Formally, the character-based semantic similarity is defined as follows:

$$\text{Sim}(s, \tilde{s}) = \sum_{T=1}^3 w_T |\vec{s}, \vec{\tilde{s}}| \quad (9)$$

where $\vec{s}$ denotes unlabeled sentence vector and $\vec{\tilde{s}}$ indicates training set sentence vector. $|\vec{s}, \vec{\tilde{s}}|$ denotes the distance between two vectors, which can be calculated via cosine distance and Euclidean distance. w_T is the weight of the distance calculated by three sentence vectors. We define sentence vector $\vec{s}$ as:

$$\vec{s} = \frac{1}{n} \sum_{i=1}^n e_i^T \quad (10)$$

where e_i^T represents the n-gram contextual embedding of the i-th character. Here are three kinds of sentence vectors due to the different values of T, which is set to 1, 2, and 3 respectively. We obtain these vector representation according to section 3.2.2. n denotes the length of a sentence. $\vec{\tilde{s}}$ is similar to $\vec{s}$.

3.3.2. The Uncertainty of Annotation. Higher the uncertainty of a sample sentence’s annotation, more useful features the sample sentence contains. In our work, therefore, we choose the samples with higher uncertainty in annotation task from unlabeled data.

Additionally, the word segmentation can be represented as a two-class problem according to one factor, that is, whether the character is the right boundary of a word in a sentence. Specifically, the labels B, M, E, S can be divided into two categories: B and M can be grouped together, denoted as N, indicating that the character isn’t the right boundary of a word; Also, E and S can be grouped and denoted as Y, indicating that the character is the right border of a word and need to be divided.

We use information entropy to measure the uncertainty of sequence annotation and it is formally denoted as:

$$E(c) = - \sum_{i=N,Y} p_i \log_2 p_i \quad (11)$$

where, c is a character from unlabeled sentences, $p_N = p_B + p_M$, $p_Y = p_E + p_S$, p_B represents the posterior probability that c is marked as B. p_M , p_E and p_S are similar to p_B .

3.3.3. Sampling Strategy. Due to the lack of labeled data, CWS models cannot learn enough features. In general, if a sample sentence is more different with sentences of the training set, the sample sentence contains more semantic features. Therefore, selecting useful sample sentences relies on two aspects: The uncertainty of annotation and the semantic similarity between unlabeled sentences and training set sentences. In general, the scoring model for selecting sample sentences is finally defined as:

$$D(s) = \lambda \sum_{i=1}^n E(c_i) + (1 - \lambda) \min Sim(s, \tilde{s}) \quad (12)$$

where, s represents a sentence in unlabeled data, n indicates the character number of the sentence, c_i represents the character in the unlabeled sentence, $\tilde{s}$ is a training set sentence, λ is a weight parameter. The higher the score, the more valuable the sample is.

4. Experiments

4.1. Datasets

We evaluate our proposed methods on three prevalent CWS datasets, including NLPCC 2016 Weibo dataset [16] and PKU, MSR from SIGHAN2005 [17]. Table 1 gives the details of three datasets. We use 10% data of shuffled training set as development set for PKU and MSR datasets. All datasets are preprocessed by replacing numbers and continuous English characters with special flags. The performance of our methods is evaluated by F-score.

In our experiments, we use about 2.2 million Weibo unlabeled data built from the Internet. Moreover, we utilize about the same 2.3 million Sogou Lab's news corpus for PKU and MSR datasets.

Table 1. Statistical information of three labeled datasets.

Datasets		Sentence	Words	Chars	Word Types	Char Types
Weibo	train	20,135	421,166	688,743	43,331	4,502
	test	8,592	187,877	315,865	27,804	3,911
PKU	train	19,056	1,109,947	1,826,448	55,302	4,685
	test	1,944	104,372	172,733	1,915	2,932
MSR	train	86,924	2,368,391	4,050,469	88,114	5,150
	test	3,985	106,873	184,355	3,953	2,836

4.2. Parameter Settings

The Hyper-parameters used in the experiments are shown in Table 2. We use different training batch sizes for three datasets. Specifically, we set batch sizes of PKU and MSR datasets to 512 and 256 respectively, and 128 for Weibo dataset.

Table 2. Hyper-parameters settings.

Hyper-parameter	value
Unigram dimension	100
Bi-gram dimension	100
Tri-gram dimension	100
LSTM hidden size	100
LSTM input size	100
Learning rate	0.01
Dropout	0.5

4.3. Experimental Results

4.3.1. The Results of Initial Word segmentation model. The initial word segmentation model for the semi-supervised method in this paper is an attention-based natural segmentation model (Attention+BiLSTM+CRF). Due to the sparsity of data, our experiments only focus on the unigram, bi-gram and tri-gram vector representation of characters. The results of our experiments are shown in Table 3. We can see that the experimental result in first group is lowest, which demonstrates the effectiveness of character contexts. To prove the effectiveness of the attention mechanism, we add

two groups of comparative experiments by concatenating contextual embeddings. As shown in Table 3, the average performance is boosted by 0.30% on F-score from baseline to attention-based model, and 0.10% on F-score from concatenation-based model to attention-based model when set N as 3, which indicates that the attention mechanism is effective to integrate contextual information. In addition, we observe that our attention-based segmentation model can achieve an improvement on small dataset (e.g. Weibo and PKU) and have less effect on dataset with enough training data (MSR). In order to ensure the precision of word segmentation model, we finally choose Attention+BiLSTM+CRF (N=3) as our initial word segmentation model.

Table 3. Segmentation results of the initial word segmentation model on three datasets, evaluated by F-score.

Models		Weibo	PKU	MSR
Baseline(BiLSTM+CRF) ^a	P	93.45	96.10	97.06
	R	93.74	95.90	96.98
	F	93.59	96.00	97.02
Concatenation+BiLSTM+CRF(N=2) ^{b2}	P	93.75	96.29	97.26
	R	93.85	96.15	97.14
	F	93.80	96.22	97.20
Concatenation+BiLSTM+CRF(N=3) ^{b3}	P	93.85	96.42	97.36
	R	93.97	95.22	97.30
	F	93.91	96.32	97.33
Attention+BiLSTM+CRF(N=2) ^{c2}	P	93.85	96.40	97.28
	R	94.05	96.26	97.16
	F	93.95	96.33	97.22
Attention+BiLSTM+CRF(N=3) ^{c3}	P	94.02	96.65	97.35
	R	94.14	96.25	97.31
	F	94.08	96.45	97.33

^a: Our baseline neural word segmentation, which uses the character embeddings as input.

^{b2}: The concatenation of the unigram and bi-gram embeddings are used as the input.

^{b3}: The concatenation of the unigram, bi-gram and tri-gram embeddings is used as the input.

^{c2}: The Attention-based segmentation model that focus on the unigram and bi-gram embeddings.

^{c3}: The Attention-based segmentation model that focus on the unigram, bi-gram and tri-gram embeddings.

4.3.2. The Results of Semi-supervised Learning Segmentation. We first investigated the impact of λ in formula (12) over segmentation performance. The parameter λ is searched ranging from 0.1 to 1 with a step size of 0.1. We put λ into the formula (12) and select the most valuable samples according the sampling strategy. In each iteration, 1000 samples were selected. The number of iterations was 5. According to Figure 4, it can be seen that the segmentation effect is best when λ is set to 0.5 in two cases. Two kinds of methods for calculating distance have little difference in Figure 4. The cosine distance gives a little improvement compared to Euclidean distance.

To verify the effectiveness of our sampling strategy, we conduct a comparative experiment which choose different number of samples randomly. We apply the sampling strategy proposed in Section 3.3 to the unlabeled data. The annotation work of unlabeled data is based on the attention-based segmentation model. Experiment results are shown in Figure 5. It can be seen that with the increase of the number of datasets, the performance of our methods becomes better, stable and convergent. It shows that our proposed method can learn continuously knowledge from selected instances. Our method achieves better performance compared to the method with selecting randomly samples, demonstrating that sampling strategy is helpful to improve the segmentation accuracy.

Furthermore, we add a comparative experiment, which uses the word-based semantic similarity. This method calculates similarity based on word-level. For unlabeled sentences, we use initial word segmentation model to annotate them. The experimental results are shown in Table 4. Compared the

word-based semantic similarity representations, the character-based method outperforms the former on three datasets. The results show that character-based models have the potential of capturing morpheme patterns, thereby improving generalization ability of word segmentation models.

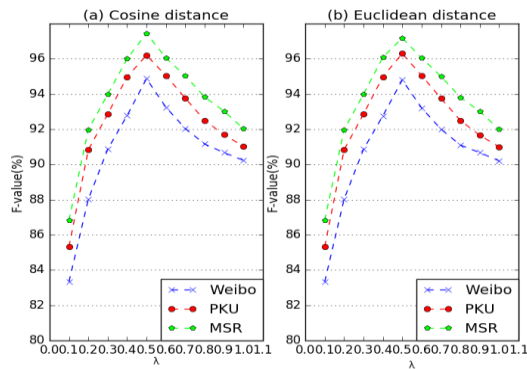


Figure 4. Performances with the different λ on Weibo, PKU and MSR datasets.

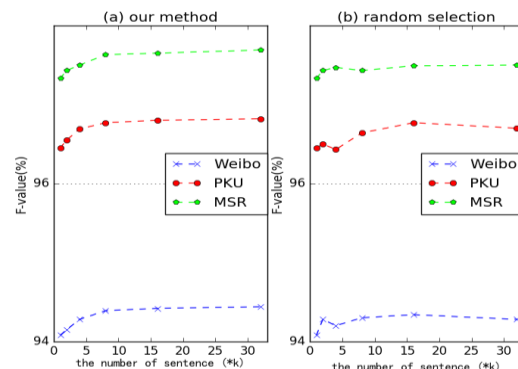


Figure 5. F1-value on three data sets when adding different numbers of samples.

Table 4. Comparison with word-based and character-based semantic similarity.

Methods		Weibo	PKU	MSR
Character-based	P	94.38	97.90	97.75
	R	94.50	97.74	97.63
	F	94.44	96.82	97.69
Word-based	P	94.27	96.82	97.64
	R	94.43	96.72	97.56
	F	94.35	96.77	97.60

Table 5. Comparisons with other methods on three datasets.

Models	Weibo	PKU	MSR
Chen et al. [1]	94.24	94.32	96.04
Yang et al. [2]	95.50	96.30	97.50
Zheng et al. [5]	-	96.67	97.50
Our work (words-based)	94.35	96.77	97.60
Our work(character-based)	94.44	96.82	97.69

Our final results compare favourably to the best neural models, including that using adversarial learning [1], and that leveraging abundant pretraining [2]. As shown in Table 5, our method gives the best accuracies on all corpora except for Weibo, where it underperforms the method of Yang et al. [2]. Compared with the state-of-the-art semi-supervised method [5] in the previous, our semi-supervised method is more stable, and the performance becomes better with the increase of unlabeled data. Experimental results verify that our method is simple but robust, which is of practical importance.

5. Conclusion and Future Work

In this paper, we propose a novel semi-supervised CWS method based on semantic similarity. Specifically, we propose a sampling strategy, which help us select more valuable sample sentences from massive unlabeled sentences to extend the training data. At the same time, we introduce character-based semantic similarity to the sampling strategy. Furthermore we introduce an attention-based neural segmentation, which dynamically focuses on the useful contextual information. The

experimental results on the Weibo dataset fully show that our method achieve better word segmentation by frequent iteration. More importantly, it also shows that our sampling strategy effectively choose samples with richer features. In addition, experimental results on PKU and MSR datasets show that our method has cross domain robustness.

In the future research, we hope to use the method proposed in this paper to automatically acquire training corpora. Moreover, we wish to optimize the semi-supervised learning sampling strategy so that it can have better domain adaptability.

References

- [1] Chen X, Shi Z, Qiu X, et al. Adversarial multi-criteria learning for chinese word segmentation[C]// Meeting of the Association for Computational Linguistics. 2017:1193-1203.
- [2] Yang J, Zhang Y, Dong F. Neural word segmentation with rich pretraining[J]. 2017:839-849.
- [3] Rei M, Crichton G K O, Pyysalo S. Attending to characters in neural sequence labeling models[J]. 2016.
- [4] Zheng B, Che W, Guo J, et al. Enhancing LSTM-based word segmentation using unlabeled Data[J]. 2017.
- [5] Sun W, Xu J. Enhancing chinese word segmentation using unlabeled data[C]// Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2011:970-979.
- [6] Yang F, Vozila P. Semi-supervised chinese word segmentation using partial-label learning with conditional random fields[C]// Conference on Empirical Methods in Natural Language Processing. 2014:90-98.
- [7] Melamud O, Goldberger J, Dagan I. Context2vec: Learning generic context embedding with bidirectional LSTM[C]// Signll Conference on Computational Natural Language Learning. 2016:51-61.
- [8] Peters M E, Ammar W, Bhagavatula C, et al. Semi-supervised sequence tagging with bidirectional language models[J]. 2017:1756-1765.
- [9] Yang F, Vozila P. An empirical study of semi-supervised chinese word segmentation using co-training[C]//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. 2013: 1191-1200.
- [10] Zeng X, Wong D F, Chao L S, et al. Co-regularizing character-based and word-based models for semi-supervised Chinese word segmentation[C]// Meeting of the Association for Computational Linguistics. 2013:171-176.
- [11] Zhang L, Wang H, Sun X, et al. Exploring representations from unlabeled data with co-training for Chinese word segmentation[C]//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. 2013: 311-321.
- [12] Huang Z, Harper M, Petrov S. Self-training with products of latent variable grammars[C]// Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2010:12-22.
- [13] Liu Y, Zhang Y. Unsupervised domain adaptation for joint segmentation and POS-tagging[C]// COLING. 2012.
- [14] Lample G, Ballesteros M, Subramanian S, et al. Neural architectures for named entity recognition[C]// Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016:260-270.
- [15] Mikolov T, Corrado G, Chen K, et al. Efficient estimation of word representations in vector space[C]// International Conference on Learning Representations. 2013:1-12.
- [16] Qiu X, Qian P, Shi Z. Overview of the NLPCC-ICCPOL 2016 shared task: chinese word segmentation for micro-blog texts[J]. 2016.
- [17] Sproat R, Emerson T. The first international chinese word segmentation bakeoff[C]// Sighan Workshop on Chinese Language Processing. Association for Computational Linguistics, 2003:133-143.