Neural Chinese Word Segmentation with Dictionary Knowledge

Junxin Liu[1], Fangzhao Wu[2], Chuhan Wu[1], Yongfeng Huang[1](✉), and Xing Xie[2]

[1] Department of Electronic Engineering, Tsinghua University, Beijing, China
{ljx16,wuch15}@mails.tsinghua.edu.cn, yfhuang@mail.tsinghua.edu.cn
[2] Microsoft Research Asia, Beijing, China
{fangzwu,Xing.Xie}@microsoft.com

Abstract. Chinese word segmentation (CWS) is an important task for Chinese NLP. Recently, many neural network based methods have been proposed for CWS. However, these methods require a large number of labeled sentences for model training, and usually cannot utilize the useful information in Chinese dictionary. In this paper, we propose two methods to exploit the dictionary information for CWS. The first one is based on pseudo labeled data generation, and the second one is based on multi-task learning. The experimental results on two benchmark datasets validate that our approach can effectively improve the performance of Chinese word segmentation, especially when training data is insufficient.

Keywords: Chinese word segmentation · Dictionary · Neural network

1 Introduction

Different from English texts, in Chinese texts there is no explicit delimiters such as whitespace to separate words. Thus, Chinese word segmentation (CWS) is an important task for Chinese natural language processing [3,17], and an essential step for many downstream tasks such as POS tagging [20], named entity recognition [9], dependency parsing [2,15] and so on.

Since a Chinese sentence is usually a sequence of Chinese characters, Chinese word segmentation is usually modeled as a sequence labeling problem [13,17]. Many sequence modeling methods such as hidden Markov model (HMM) [5] and conditional random field (CRF) [6] have been applied to the CWS task. A core problem in these sequence modeling based CWS methods is building the feature vector for each character in sentences. In traditional CWS methods these character features are constructed via manual feature engineering [10,19]. These handcrafted features need a large amount of domain knowledge to design, and the size of these features is usually very large [3].

In recent years, many neural network based methods have been proposed for CWS [3,16,17,20]. For example, Peng et al. [11] proposed to use Long Short-Term Memory Neural Network (LSTM) to learn the character representations for CWS and use CRF to jointly decode the labels. However, these neural network

M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11109, pp. 80–91, 2018.
https://doi.org/10.1007/978-3-319-99501-4_7

based methods usually rely on a large number of labeled sentences. For words which are scarce or absent in training data, these methods are very difficult to correctly segment the sentences that contain these words [17]. Since these words are in large quantity, it is very expensive and even unpractical to improve the coverage of these words via annotating more sentences. Luckily, many of these words are well defined in existing Chinese dictionaries. Thus, Chinese dictionaries have the potential to improve the performance of neural network based CWS methods and reduce the dependence on labeled data [17].

In this paper we propose to incorporate the dictionary information into neural network based CWS approach in an end-to-end manner without any feature engineering. More specifically, we propose two methods to incorporate the dictionary information for CWS. The first one is based on pseudo labeled data generation, where we build pseudo labeled sentences by randomly sampling words from Chinese dictionaries. The second one is based on multi-task learning. In this method we introduce another task named Chinese word classification (i.e., classifying a sequence of Chinese characters based on whether they can form a Chinese word), and jointly train this task with CWS by sharing the parameters of neural networks. We conducted extensive experiments on two benchmark datasets. The experimental results validate that our methods can effectively improve the performance of CWS, especially when training data is insufficient.

2 Related Work

In recent years, many neural network based methods have been proposed for Chinese word segmentation [3,16,17,20]. Most of these methods model CWS as a sequence labeling task [3,17]. The core difference between these methods mainly lies in how they learn the contextual feature representation for each character in sentence. For example, Zheng et al. [20] proposed to use multilayer perceptrons to learn feature representations of characters from a fixed window. Chen et al. [3] used LSTM to capture global contextual information. They also explicitly captured the local context by combining the embedding of current character with the embeddings of neighbouring characters as the input of LSTM. In [11], LSTM is used to learn character representations and CRF is used to jointly decode the labels. These methods rely on a large number of labeled sentences to train CWS models and cannot exploit the useful information in Chinese dictionaries [17]. Since there are massive Chinese words which are scarce or absent in the labeled sentences, these neural CWS methods usually have difficulty in correctly segmenting sentences containing these words [17].

Recently, incorporating the dictionary information into neural Chinese word segmentation has attracted increasing attentions [14,17]. For example, Yang et al. [14] proposed to incorporate external information such as punctuation, automatic segmentation and POS data into neural CWS via pretraining. However, the useful information in Chinese dictionaries is not considered in their method. Zhang et al. [17] proposed to incorporate the dictionary information into an LSTM based neural CWS method via feature engineering. They used several handcrafted templates to build an additional feature vector for each character using the dictionary and the neighbouring characters. These additional feature vectors are fed to another LSTM network to learn additional character representations. However, designing these handcrafted feature templates needs a lot of domain knowledge. In addition, more model parameters are introduced in their method, making it more difficult to train neural CWS model especially when training data is insufficient. Different from [17], our method to incorporate dictionary information into neural CWS can be trained in an end-to-end manner and does not need manual feature engineering. Experimental results show that our approach can achieve better performance than the method in [17].

3 Our Approach

In this section we first present the basic neural architecture for Chinese word segmentation used in our approach. Then, we introduce our methods of incorporating dictionary information for neural CWS.

3.1 Basic Neural Architecture

Following many previous works [3,17], in this paper we model Chinese word segmentation as a character-level sequence labeling problem. For each character in a sentence, our model will assign one of the tags in a predefined tag set to it, indicating its position in a word. We use the *BMES* tagging scheme, where B, M and E mean the beginning, middle and end position in the word, and S represents single character word.

The basic neural architecture for CWS used in our approach is CNN-CRF. This neural architecture contains three main layers. The first layer is the character embedding layer. In this layer, the input sentence is converted to a sequence of vectors. Denote the input sentence as $\mathbf{x} = [c_1, c_2, ..., c_M]$, where M is the sentence length and c_i is the i-th character in this sentence. After the embedding layer, the input sentence will become $\mathbf{x} = [\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_M]$, where $\mathbf{c}_i \in \mathcal{R}^D$ is the embedding of character c_i and D is the embedding dimension.

The second layer is the CNN layer. Previous studies show that local context information is important for Chinese word segmentation [1,14]. In addition, many researchers have shown that CNN is effective in capturing local context information [7,12,18]. Motivated by these observations, we use CNN to learn the contextual representations of characters for CWS. Denote $\mathbf{w} \in \mathcal{R}^{KD}$ as the parameter of a filter with kernel size K, then the hidden representation of the i-th character generated by this filter is formulated as follows:

$$h_i = f(\mathbf{w}^T \times \mathbf{c}_{i-\lceil \frac{k-1}{2} \rceil : i+\lfloor \frac{k-1}{2} \rfloor} + b), \tag{1}$$

where $\mathbf{c}_{i-\lceil \frac{k-1}{2} \rceil : i+\lfloor \frac{k-1}{2} \rfloor}$ is the concatenation of the embeddings of neighbouring characters, f is the ReLU function, and $\mathbf{w}$ and b are the parameters of the filter.

Multiple filters with different kernel sizes are used. The final hidden representation of the i-th character is the concatenation of the output of all filters at this position, which is denoted as $\mathbf{h}_i \in \mathcal{R}^F$ (F is the number of filters).

The third layer is the CRF layer. In Chinese word segmentation there are usually strong dependencies among neighbouring tags [3]. For example, the tag M cannot follow tag S or E. Following many previous works on CWS [11,17], we use CRF to capture the dependencies among neighbouring tags. Denote the input sentence as $\mathbf{x} = [c_1, c_2, ..., c_M]$, and the predicted tag sequence as $\mathbf{y} = [y_1, y_2, ..., y_M]$, then the score of this prediction is formulated as:

$$g(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{M} (S_{i,y_i} + A_{y_{i-1},y_i}), \tag{2}$$

where S_{i,y_i} is the score of assigning tag y_i to the i-th character, and A_{y_{i-1},y_i} is the score of jumping from tag y_{i-1} to tag y_i. In our approach, S_i is defined as:

$$S_i = \mathbf{W}^T \mathbf{h}_i + \mathbf{b}, \tag{3}$$

where $\mathbf{h}_i$ is the hidden representation of the i-th character learned by the CNN layer, and $\mathbf{W} \in \mathcal{R}^{F \times T}$ and $\mathbf{b} \in \mathcal{R}^T$ (T is the size of the tag set) are the parameters for character score prediction. In CRF, the probability of sentence $\mathbf{x}$ having tag sequence $\mathbf{y}$ is defined as:

$$p(\mathbf{y}|\mathbf{x}) = \frac{\exp(g(\mathbf{x}, \mathbf{y}))}{\sum_{\mathbf{y}' \in \mathcal{Y}(\mathbf{x})} \exp(g(\mathbf{x}, \mathbf{y}'))}, \tag{4}$$

where $\mathcal{Y}(\mathbf{x})$ is the set of all possible tag sequences of sentence $\mathbf{x}$.

Then the loss function can be formulated as:

$$\mathcal{L} = -\sum_{i=1}^{N} \log(p(\mathbf{y}_i|\mathbf{x}_i)), \tag{5}$$

where N is the number of labeled sentences for training, and $\mathbf{y}_i$ is the ground-truth tag sequence of the i-th sentence.

For prediction, given a sentence $\mathbf{x}$ to be segmented, the predicted tag sequence $\mathbf{y}^\star$ is the one with the highest likelihood:

$$\mathbf{y}^\star = \arg\max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} p(\mathbf{y}|\mathbf{x}). \tag{6}$$

We use Viterbi algorithm to solve the decoding problem in Eq. (6).

3.2 Incorporating Dictionary Information for Neural CWS

Existing neural CWS methods usually rely on a large number of labeled sentences for model training. Researchers have found that the neural models trained on labeled sentences usually have difficulties in segmenting sentences which contain OOV or rarely appearing words [17]. For example, a Chinese sentence is

"人工智能最近很火" (Recently AI is hot). Its ground-truth segmentation is "人工智能/最近/很火". However, if "人工智能" (AI) does not appear in the labeled data or only appears for a few times, then there is a large probability that this sentence will be segmented into "人工/智能/最近/很火", since "人工" and "智能" are both popular words which may frequently appear in the labeled data. Luckily, many of these rare words are included in Chinese dictionary. If the neural model is aware of that "人工智能" is a Chinese word, then it can better segment the aforementioned sentence. Thus, dictionary information has the potential to improve the performance of neural CWS methods.

In this paper we propose two methods for incorporating dictionary information into training neural CWS models. Next we will introduce them in detail.

Pseudo Labeled Data Generation. Our first method for incorporating dictionary information into neural CWS model training is based on pseudo labeled data generation. More specifically, given a Chinese dictionary which contains a list of Chinese words, we randomly sample U words and use them to form a pseudo sentence. For example, assuming that three words "很火", "最近" and "人工智能" are sampled, then a pseudo sentence "很火最近人工智能" can be built. Since the boundaries of these words are already known, the tag sequence of the generated pseudo sentence can be automatically inferred. For instance, the tag sequence of aforementioned pseudo sentence is "B/E/B/E/B/M/M/E" under the *BMES* tagging scheme. Then we repeat this process until N_p pseudo labeled sentences are generated. These pseudo labeled sentences are added to labeled data set to enhance the training of neural CWS model.

Since the pseudo labeled sentences may have different informativeness from the manually labeled sentences, we assign different weights to the loss on these two kinds of training data, and the final loss function is formulated as:

$$\mathcal{L} = -\sum_{i=1}^{N} \log(p(\mathbf{y}_i|\mathbf{x}_i)) - \lambda_1 \sum_{i=1}^{N_p} \log(p(\mathbf{y}_i^s|\mathbf{x}_i^s)), \tag{7}$$

where $\mathbf{x}_i^s$ and $\mathbf{y}_i^s$ represent the i-th pseudo labeled sentence and its tag sequence, and λ_1 is a non-negative coefficient.

Multi-task Learning. Our second method for incorporating dictionary information into neural CWS model training is based on multi-task learning. In this method, we design an additional task, i.e., word classification, which means classifying a sequence of Chinese characters based on whether it can be a Chinese word. For example, the character sequence "人工智能" will be classified to be true, while the character sequence "人重智新" will be classified to be false. The positive samples are obtained from a Chinese dictionary. The negative samples are obtained via randomly sampling a word from the dictionary, and then each character in this word will be randomly replaced by a random selected character with a probability p. This step is repeated multiple times until a predefined

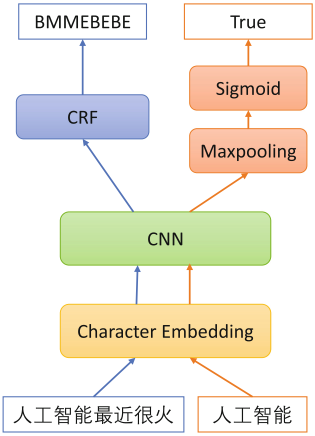

Fig. 1. Our proposed framework for jointly training CWS and word classification models. The left part is for CWS and the right part is for word classification.

number of negative samples are obtained. We use a neural method for the word classification task, whose architecture is similar with the CNN-CRF architecture for CWS, except that the CRF layer is replaced by a max-pooling layer and a sigmoid layer for binary classification. The loss function of the word classification task is formulated as:

$$\mathcal{L} = \sum_{i=1}^{N_w} \log(1 + e^{-y_i s_i}), \tag{8}$$

where N_w is the number of training samples for word classification, s_i is the predicted score of the i-th sample, and y_i is the word classification label which can be 1 or -1 (1 represents true and -1 represents false).

Motivated by multi-task learning, we propose a unified framework to jointly train the Chinese word segmentation model and the word classification model, which is illustrated in Fig. 1. In our framework, the CWS model and the word classification model share the same embedding layer and CNN layer. In this way, these two layers can better capture the word information in Chinese dictionary via jointly training with the word classification task, and the performance of CWS can be improved. In model training we assign different weights to the loss of these two tasks, and the final loss function is:

$$\mathcal{L} = -(1 - \lambda_2) \sum_{i=1}^{N} \log(p(\mathbf{y}_i|\mathbf{x}_i)) + \lambda_2 \sum_{i=1}^{N_w} \log(1 + e^{-y_i s_i}), \tag{9}$$

where λ_2 is a coefficient ranging from 0 to 1.

4 Experiment

4.1 Dataset

In our experiments we used two benchmark datasets released by the third international Chinese language processing bakeoff[1] [8]. The detailed statistics of these two datasets are summarized in Table 1. We used the last 10% data of the training set as development set.

Table 1. The statistics of datasets.

Dataset		#Sentence	#Word	#Character	OOV Rate
MSRA	Train	46.3K	1.27M	2.17M	-
	Test	4.4K	0.10M	0.17M	3.4%
UPUC	Train	18.8K	0.51M	0.83M	-
	Test	5.1K	0.15M	0.26M	8.8%

4.2 Experimental Settings

The character embeddings used in our experiments were pretrained on the Sogou news corpus[2] using the word2vec[3] tool. The dimension of character embedding is 200. We used 400 filters in the CNN layer and the kernel sizes of these filters range from 2 to 5. Rmsprop [4] was used as the algorithm for neural model training. The learning rate was set to 0.001 and the batch size was 64. Dropout was applied to the embedding layer and the CNN layer. The dropout rate was set to 0.3. We use early stopping strategy. When the loss on the development set doesn't reduce after 3 consecutive epochs, the training is stopped. We repeated each experiment for 5 times and reported the average results.

4.3 Performance Evaluation

In this section we compare our approach with several baseline methods. These baseline methods include: (1) Chen et al. [3], a LSTM based CWS method which also considers local contexts; (2) LSTM-CRF, a popular neural CWS method based on the LSTM-CRF architecture [11,17]; (3) CNN-CRF, a neural CWS method based on the CNN-CRF architecture, which is the basic model for our approach; (4) Zhang et al. [17], a neural CWS method which can incorporate dictionary information via feature templates. In order to evaluate the performance of different methods under different amounts of labeled data, we randomly sampled different ratios of labeled data for training. The experimental results are summarized in Tables 2 and 3. According to Tables 2 and 3, we have two observations.

[1] http://sighan.cs.uchicago.edu/bakeoff2006/download.html.

[2] http://www.sogou.com/labs/resource/ca.php.

[3] https://code.google.com/archive/p/word2vec/.

Table 2. The performance of different methods on the *MSRA* dataset. *P*, *R* and *F* represent precision, recall and Fscore respectively. *Ours_Pseudo* represents our approach based on pseudo labeled data generation, and *Ours_Multi* represents our approach based on multi-task learning.

	1%			10%			100%		
	P	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
Chen et al. [3]	75.50	75.80	75.64	87.71	86.22	86.96	94.24	93.35	93.80
LSTM-CRF	75.88	74.86	75.36	85.52	84.81	85.16	94.26	93.29	93.78
CNN-CRF	75.59	74.43	75.00	89.72	89.14	89.43	95.03	94.53	94.78
Zhang et al. [17]	75.75	75.95	75.85	89.52	89.01	89.27	95.71	95.41	95.56
Ours_Pseudo	80.58	77.97	79.25	90.49	89.59	90.04	95.36	94.71	95.03
Ours_Multi	78.47	77.31	77.88	89.91	89.27	89.59	95.10	94.50	94.80

Table 3. The performance of different methods on the *UPUC* dataset.

	5%			25%			100%		
	P	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
Chen et al. [3]	82.31	82.60	82.44	88.00	89.90	88.94	90.79	92.92	91.84
LSTM-CRF	81.08	80.88	80.98	86.76	88.40	87.57	91.39	92.58	91.98
CNN-CRF	82.44	84.50	83.46	89.95	91.57	90.75	92.22	93.84	93.02
Zhang et al. [17]	83.38	84.98	84.17	89.93	91.41	90.66	92.60	93.89	93.24
Ours_Pseudo	87.37	86.56	86.97	90.97	92.04	91.50	92.77	94.09	93.43
Ours_Multi	84.59	86.22	85.40	90.43	91.68	91.05	92.35	93.93	93.13

First, both of our approaches perform better than various neural CWS methods which do not consider dictionary information, and the performance advantage becomes larger when training data is insufficient. This result validates that by incorporating the dictionary information our approaches can effectively improve the performance of neural CWS. This is because there are many words which do not appear or rarely appear in the training data, and the neural CWS models which are trained purely on labeled data usually have difficulty in segmenting sentences containing these words. Many of these words are usually included in Chinese dictionaries, and exploiting the useful information in dictionaries can help the neural CWS model better recognize these words.

Second, although the method proposed in [17] can also incorporate the dictionary information for CWS, our approaches usually can outperform it, especially when training data is insufficient. This result shows that our approaches are more appropriate for incorporating dictionary information for CWS than the method proposed in [17]. This is maybe because in [17] the feature templates for incorporating dictionary information are manually designed, which may not be optimal.

In addition, in [17] an additional LSTM network is used to learn character representations from these dictionary based features. Thus, more model parameters are incorporated, making it more difficult to train the CWS model especially when training data is insufficient. Our approaches do not rely on feature engineering and the additional model parameters introduced in our approaches are limited. Thus, our approach can achieve better performance than [17].

4.4 Influence of Dictionary

In this section we conducted several experiments to explore the influence of the type and the size of Chinese dictionary on the performance of our approach.

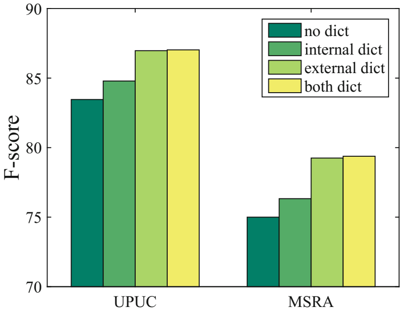

Fig. 2. The influence of dictionary type.

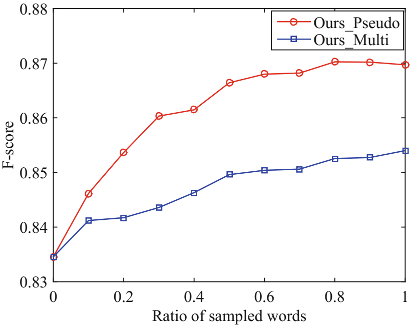

Fig. 3. The influence of dictionary size.

First, we explore the influence of dictionary type. In previous section, the Chinese dictionary used in our approach is the Sogou Chinese Dictionary, which can be regarded as an external dictionary. We also built an internal dictionary using the words appearing in the training data. The results of our approach without any dictionary, with only internal dictionary, with only external dictionary, and with both dictionaries are summarized in Fig. 2. We randomly sampled 5% training data of *UPUC* dataset and 1% training data of *MSRA* dataset for model training.

According to Fig. 2, with external dictionary our approach can improve the performance of CWS. In addition, our approach can also improve the performance with only internal dictionary. This result is promising, since the internal dictionary is built on the words appearing in training data and no external resource is involved. In addition, incorporating both internal and external dictionaries can further improve the performance of our approach, which indicates that these two dictionaries contain complementary information.

Next, we explore the influence of the dictionary size on the performance of our approach. We randomly sampled different numbers of words from the Sogou dictionary, and the experimental results on *UPUC* dataset are summarized in Fig. 3.

From Fig. 3, with the size of dictionary grows the performance improves. This result is intuitive since when a dictionary contains more words it can have a better coverage of the Chinese words, and our approach can benefit from this by incorporating the useful information in these words into training neural CWS model.

4.5 The Influence of Parameters

There are two most important parameters in our approaches. The first one is λ_1, which controls the relative importance of pseudo labeled samples. The second one is λ_2, which controls the relative importance of word segmentation task. The influence of these parameters on the performance of our approaches is illustrated in Figs. 4 and 5.

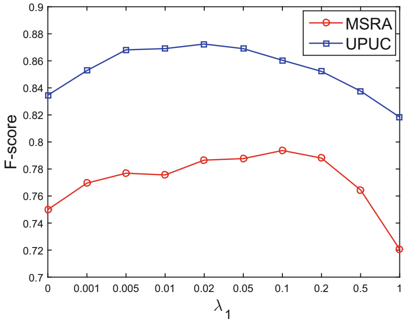

Fig. 4. The influence of λ_1.

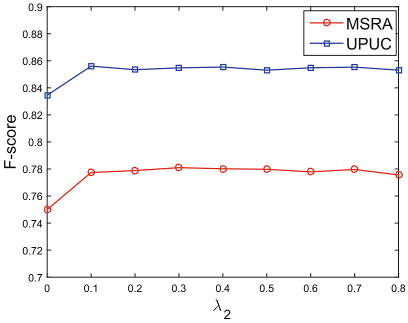

Fig. 5. The influence of λ_2.

From Figs. 4 and 5 we can see that when λ_1 and λ_2 are too small, the performance of our approach is not optimal, and improves as λ_1 and λ_2 increase. This is because when these parameters are too small, the useful information in the dictionary is not fully exploited. However, when λ_1 and λ_2 become too large, the performance of our approach decreases. This is because in these cases the pseudo labeled samples and the word classification task are over-emphasized. Accordingly, the manually labeled samples and the CWS task are not fully respected. Thus, a moderate value is most appropriate for λ_1 and λ_2.

4.6 Case Study

In this section we conducted several case studies to explore why our approach can improve the performance of Chinese word segmentation via incorporating the dictionary information. Several segmentation results of our approach without dictionary (i.e., the CNN-CRF method), with internal dictionary and with external dictionary are shown in Table 4. For illustration purpose, we only show the results of our approach based on pseudo labeled data generation.

Table 4. Several Chinese word segmentation examples.

	Example 1	Example 2
Original	5 名男子和被害人有恩怨	警方一口气带回了 5 0 多人
CNN-CRF	5 /名/男子/和/被/害/人/有/恩怨	警方/一/口/气/带回/了/ 5 0 多/人
+Internal dictionary	5 /名/男子/和/被/害/人/有/恩怨	警方/一口气/带回/了/ 5 0 多/人
+External dictionary	5 /名/男子/和/被害人/有/恩怨	警方/一口气/带回/了/ 5 0 多/人

According to Table 4, after incorporating the dictionary information, our approach can correctly segment many sentences where the basic CNN-CRF method has difficulties. For instance, in the first example, the true segmentation of "被害人" is "被害人". However, CNN-CRF incorrectly segments it into "被/害/人", because "被害人" is an OOV word which does not appear in training data. Our approach with external dictionary can correctly segment this sentence because the word "被害人" is in the external dictionary and our approach can fully exploit this useful information. In the second example, CNN-CRF incorrectly segments "一口气" into "一/口/气", because "一口气" is an rare word which only appears 2 times in the training data which is difficult for neural CWS model to segment it. Since this word is in both internal and external dictionaries, our approach with either dictionary can correctly segment this sentence. Thus, these results clearly show that incorporating dictionary information into training neural CWS methods is beneficial.

5 Conclusion

In this paper we present two approaches for incorporating the dictionary information into neural Chinese word segmentation. The first one is based on pseudo labeled data generation, where pseudo labeled sentences are generated by combining words randomly sampled from dictionary. The second one is based on multi-task learning, where we design a word classification task and using the dictionary to build labeled samples. We jointly train the Chinese word segmentation and the word classification task via sharing the same network parameters. Experimental results on two benchmark datasets show that our approach can effectively improve the performance of Chinese word segmentation, especially when training data is insufficient.

References

1. Cai, D., Zhao, H., Zhang, Z., Xin, Y., Wu, Y., Huang, F.: Fast and accurate neural word segmentation for Chinese. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, vol. 2, pp. 608–615 (2017)
2. Chen, W., Zhang, Y., Zhang, M.: Feature embedding for dependency parsing. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 816–826 (2014)

3. Chen, X., Qiu, X., Zhu, C., Liu, P., Huang, X.: Long short-term memory neural networks for Chinese word segmentation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1197–1206 (2015)
4. Dauphin, Y., de Vries, H., Bengio, Y.: Equilibrated adaptive learning rates for non-convex optimization. In: Advances in Neural Information Processing Systems, pp. 1504–1512 (2015)
5. Eddy, S.R.: Hidden markov models. Curr. Opin. Struct. Biol. **6**(3), 361–365 (1996)
6. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: probabilistic models for segmenting and labeling sequence data (2001)
7. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
8. Levow, G.A.: The third international Chinese language processing bakeoff: word segmentation and named entity recognition. In: Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing, pp. 108–117 (2006)
9. Luo, W., Yang, F.: An empirical study of automatic Chinese word segmentation for spoken language understanding and named entity recognition. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 238–248 (2016)
10. Peng, F., Feng, F., McCallum, A.: Chinese segmentation and new word detection using conditional random fields. In: Proceedings of the 20th International Conference on Computational Linguistics, p. 562. Association for Computational Linguistics (2004)
11. Peng, N., Dredze, M.: Multi-task domain adaptation for sequence tagging. In: Proceedings of the 2nd Workshop on Representation Learning for NLP, pp. 91–100 (2017)
12. dos Santos, C., Gatti, M.: Deep convolutional neural networks for sentiment analysis of short texts. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 69–78 (2014)
13. Xue, N.: Chinese word segmentation as character tagging. Int. J. Comput. Linguisti. Chin. Lang. Process. **8**(1), 29–48 (2003). Special Issue on Word Formation and Chinese Language Processing
14. Yang, J., Zhang, Y., Dong, F.: Neural word segmentation with rich pretraining. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 839–849 (2017)
15. Zhang, M., Zhang, Y., Che, W., Liu, T.: Chinese parsing exploiting characters. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. Long Papers, vol. 1, pp. 125–134 (2013)
16. Zhang, M., Zhang, Y., Fu, G.: Transition-based neural word segmentation. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 421–431 (2016)
17. Zhang, Q., Liu, X., Fu, J.: Neural networks incorporating dictionaries for Chinese word segmentation (2018)
18. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Advances in Neural Information Processing Systems, pp. 649–657 (2015)
19. Zhao, H., Huang, C.N., Li, M., Lu, B.L.: Effective tag set selection in Chinese word segmentation via conditional random field modeling. In: Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation, pp. 87–94 (2006)
20. Zheng, X., Chen, H., Xu, T.: Deep learning for Chinese word segmentation and pos tagging. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 647–657 (2013)