Chinese Word Segmentation and Recognition Based on Separable Convolution Bidirectional Long Short-Term Memory and Feature Point

Fan Sun¹, Zijiao Chen¹, and Jingrui Pei²,3*

¹ School of Chinese Studies, Dalian University of Foreign Languages Dalian, 116044, China
² Chinese College of Minority Languages and Literature, Minzu University of China Beijing, 100000, China
³ Software College, Shenyang Normal University Shenyang, 110034, China

*Corresponding author. E-mail: 360321978@qq.com

Received: Aug. 01, 2020; Accepted: Nov. 16, 2020

Chinese automatic word segmentation is the premise of Chinese information processing, which is widely used in Chinese full-text retrieval, Chinese automatic full-text translation, Chinese text-to-speech conversion (TTS) and other fields. A dictionary plays an important role in Chinese word segmentation. The advantages and disadvantages of the word segmentation mechanism directly affect the speed and efficiency of Chinese word segmentation. Therefore, we propose a deep learning method for Chinese word segmentation. First, a separable convolution bidirectional long and short-term memory condition random field word segmentation model with feature points containing dictionary features is constructed. The model parameters are obtained by training on the existing word segmentation corpus. Then, the software engineering field text is used as the small-scale word segmentation training corpus. The word segmentation model of general corpus is fine-tuned. The experimental results show that the transfer learning reduces the iteration times of the domain segmentation model. Meanwhile, compared with other Chinese word segmentation models, the proposed model can reduce the corpus labeling time in training process and realize the cross-domain transfer learning of word segmentation model.

Keywords: Chinese word segmentation; deep learning; dictionary feature; bidirectional long and short-term memory; condition random field; separable convolution; Feature point

http://dx.doi.org/10.6180/jase.202104_24(2).0014

1. Introduction

Chinese word segmentation is a process of dividing Chinese into word units according to information processing requirements [1–3]. Chinese word segmentation is the foundation of other Chinese information processing applications. The results directly affect the accuracy of machine translation, information retrieval, information extraction and other information processing technologies. In recent years, Chinese word segmentation technology has been developed rapidly. However, due to the complexity of natural language, Chinese word segmentation is still one of the difficulties in information processing [4–7].

Although the word segmentation method based on statistics has a great improvement in terms of word segmentation efficiency, when the testing corpus and training corpus are inconsistent, the word segmentation effect will be greatly reduced, which means that it is necessary to provide word segmentation training corpus in corresponding fields when conducting word segmentation for texts in different professional fields. And in order to ensure the accuracy of word segmentation, the scale of training corpus is generally larger. For example, the PKU training corpus provided by SIGHAN CWS BACKOFF 2005 includes 19056 sentences. Labeling word samples in each professional field require a large amount of manpower. To improve the domain adaptability of word segmentation methods, Maslov [8] incorporated domain dictionary information into CRF model with a feature form. Yao [9] added relevant features of domain dictionary into deep neural network. Zhang [10] used domain dictionary to further reverse the maxi-
mum matching for word segmentation results. However, only adding the domain dictionary without considering the domain training corpus could lead to the absence of word segmentation features. Rooij [11] selected the most unlabeled small-scale corpus in the texts of professional fields for manual annotation. Chen [12] proposed a semi-supervised CRF model to introduce the conditional entropy of unlabeled data into the training objective function to reduce labeled samples.

Transfer learning can be defined that the learned knowledge in one task environment is adopted to improve the generalization of models in another task environment [13, 14]. It is used to solve problems such as difficulty in obtaining training data in new field and different distribution of training data and test data. Transfer learning methods are divided into: instance-based transfer, feature-based transfer, parameter-based transfer and knowledge-based transfer. Ying [15] proposed the transfer learning of deep circular neural network for sequence labeling. In order to improve the ability of word segmentation in the professional field, Samala [16] jointly trained the word segmentation model in the two fields by controlling the distance of LSTM output distribution of the two models in the general field and the professional field.

This paper adopts a relatively small number of training samples in the software engineering field to fine-tune the deep circular neural network that has been trained on word segmentation materials in the general field. Finally, it tests the corpus in the software engineering field. The test results show that the F value of the proposed word segmentation method is better improved in the field of software engineering. The field adaptability of the proposed professional field segmentation model in this paper is mainly reflected in the following aspects: reducing the number of training samples in the professional field and reducing the inconsistencies impact of word segmentation characteristics in the general field and professional field.

The structures of the paper are as follows. Segment model based on LSTM is introduced in section 2. We detailed explain the proposed network model in domain word segmentation training corpus in section 3. Section 4 gives the experiments and analysis. There is a conclusion in section 5.

2. Proposed word segment model

2.1. Word segment sequence annotation modeling

Chinese word segmentation problem can be regarded as sequence annotation problem. Any Chinese sentence can be expressed as $x = (x_1, x_2, \cdots, x_n)$. Where $x_i, i \in \{1, 2, \cdots, n\}$ is a character such as Chinese, numeral, letter, punctuation mark or other symbols. For $x_i$, there are four positions in words: begin-of-word (B), middle-of-word (M), end-of-word (E), and single-character-word (S). A particle unit is expressed as BE, BME (M is greater than or equal to 1), or S. The word segment sequence annotation result of sentence $x$ can be expressed as $y = (y_1, y_2, \cdots, y_n)$, where $y_i \in \{B, M, E, S\}$ is the corresponding word position of $x_i$.

There are two steps to solve the word segmentation problem by using the sequence labeling model: training and prediction. During the training, N labeled training samples $(x_1^{train}, y_1^{train}), (x_2^{train}, y_2^{train}), \cdots, (x_N^{train}, y_N^{train})$ are used to estimate the parameters of the selected word segmentation model. In the prediction, for the given input sequence $x^{test}$, it finds the output sequence $y^{test}$ of the conditional probability $p(y^{test} | x^{test})$. Finally, the segmentation results are recovered from the labeled sequence $y^{test}$.

2.2. Word segmentation sequence annotation with dictionary based on separable convolution BI-LSTM-CRF

The BI-LSTM-CRF [17] word segmentation model consists of three parts: character embedding layer, BI-LSTM layer and CRF layer. The specific structure is shown in Fig. 1.

![Fig. 1. Proposed structure.](image)

2.2.1. Character embedding layer with feature point

The character embedding layer converts the character $x_i$ in the sentence into the vector form $e_i$ and it is inputted to BI-LSTM layer. First, the Word2Vec algorithm is used to train data in the Chinese wikipedia corpus to obtain the d-dimensional character vector and form the character matrix with $d \times N$. Where $N$ represents the number of valid characters in the training corpus. Second, for each character $x_i$ in a sentence, it sets a window with a length of $l = 5$ (the value of $l$ is adjustable) to extract the context characters sequence $(x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2})$ of $x_i$. For each character in the window, it finds the corresponding vector from the character matrix. Finally, for the current character $x_i$, a character embedding vector is constructed. The vector
\(e_i \in \mathbb{R}^{(2t-1) \times d}\) is composed of the character vector in the window and the bigram feature vector, where the bigram feature vector is the average of the consecutive character vectors in the window. Adding context vector and bigram feature vector to \(e_i\) can improve the word segment performance. Then we extract related feature point for the layer as shown in Fig. 2.

2.2.2. Separable convolution BI-LSTM layer

LSTM network belongs to recurrent neural network (RNN) [18]. Due to the gradient dissipation during RNN training, LSTM is improved. LSTM network is composed of LSTM cells. A LSTM cell is composed of input gate, forgetting gate, output gate and cell state. The input gate controls the cell state by adding new information, the forgetting gate controls the discarded information of the cell state at the previous moment. The output gate controls the output of the cell state. \(e_t \in \mathbb{R}^d\) is the input vector at time \(t\), \(h_{t-1} \in \mathbb{R}^K\) is the output at time \(t-1\) of the LSTM cell, \(c_{t-1} \in \mathbb{R}^k\) is the cellular state at time \(t-1\). The workflow of LSTM at time \(t\) can be expressed as shown in equations (1 ~ 6).

\[
\begin{align*}
i_t &= \sigma(W_i h_{t-1} + U_i e_t + b_i) \\
f_t &= \sigma(W_f h_{t-1} + U_f e_t + b_f) \\
c_t' &= \tanh(W_c h_{t-1} + U_c e_t + b_c) \\
c_t &= f_t' \odot c_{t-1} + i_t \odot c_t' \\
h_t &= o_t \odot \tanh(c_t')
\end{align*}
\]

Where \(\sigma\) function and the tanh function compute each element in the vector. \(\sigma(x) = (1 + e^{-x})^{-1}, \tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})\). \(\odot\) denotes that each element is multiplied by each other. \(W_i, W_f, W_o\) and \(W_c\) are the weight matrices of \(h_{t-1}\). \(U_i, U_f, U_c\) and \(U_t\) are the weight matrices of \(e_t\). \(b_i, b_f, b_o\) and \(b_c\) are offset vectors. \(i_t', f_t', o_t'\) and \(c_t'\) represent input gate, forgetting gate, output gate, and cell state, respectively. The separable convolution BI-LSTM is shown in Fig. 3.

After extracting the features with separable convolution, the extracted features are input into the CRF layer.

2.2.3. CRF layer

In the training data, assuming that the sentence is \(x = (x_1, x_2, \cdots, x_n)\), its corresponding word segmentation annotation result is \(y = (y_1, y_2, \cdots, y_n)\), \(y_i \in \{B, M, E, S\}\). After the CRF layer processing, it predicts the word segmentation annotation result as \(\hat{y} = (\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_n)\), \(\hat{y}_i \in \{B, M, E, S\}\). At time \(t\), it processes the character \(x_t\) in sentence \(x\), the output through the BI-LSTM layer is \(h^t\), then it is input into CRF layer. \(P_t \in \mathbb{R}^k\) is defined as the score of \(x_t\) on \(\{B, M, E, S\}\). \(P_t = W_t h^t + b_t \in \mathbb{R}^{K \times 2k}\), \(b_t \in \mathbb{R}^k\) is trainable parameter. \(P_t, \hat{y}_i\) represents the score of \(x_t\) on the prediction annotation \(\hat{y}_i\). We define the transfer score matrix \(A \in \mathbb{R}^{4 \times 4}\), \(A_{ij}(i, j \in \{B, M, E, S\})\) that the score of the transformation from the predicted annotation result \(i\) at the previous time to the predicted annotation result \(j\) at the next time. Given \(x\) and \(\hat{y}\), the predicted score is defined as shown in equation (7).

\[
s(x, \hat{y}) = \sum_i (A_{\hat{y}_i, \hat{y}_i} + P_t, \hat{y}_i) \tag{7}
\]

The conditional probability \(P(\hat{y}|x)\) of \(x\) and \(\hat{y}\) is shown in equation (8).

\[
P(\hat{y}|x) = \frac{e^{s(x, \hat{y})}}{\sum_{\hat{y} \in \mathcal{Y}_x} e^{s(x, \hat{y})}} \tag{8}
\]

Where \(\mathcal{Y}_x\) represents the set of all possible annotation sequences of sentence \(x\). During the deep neural network
training, the loss function is shown in equation (9).

\[
J(\theta) = -\sum \log(P(y|x)) + \alpha \Omega(\theta)
\]

(9)

Where \(x\) and \(y\) are the sentence and the corresponding annotation sequence in the training data. \(\omega\) is a regular term added to prevent over-fitting in the neural network.

2.2.4. Domain dictionary features

The introduction of domain dictionary features into the neural network word segmentation model can improve domain adaptability. The domain dictionary feature vector of the current character is a vector composed of 0 and 1, it can determine whether the string with specific length before and after the current character is in the domain dictionary. The specific construction process is shown in reference [19]. The dictionary features are combined with the traditional BI-LSTM-CRF network using a hypernetwork architecture.

The LSTM layer of dictionary features is composed of BI-LSTM network. It inputs domain dictionary feature vector \(t_i\). The output \(h_i^{(t)}\) is sent to the main LSTM layer. The gate \(g \in \{i, c, f, o\}\) of the main LSTM layer is shown in equations (10 ~ 13).

\[
S_i^{(M)} = d_g h \Theta W_g^{(M)} h_{i-1}^{(M)} + d_x \Theta U_g^{(M)} e_i^{(M)} + b_g
\]

(10)

\[
d_g h = W_g^{(M)} z_g^{(t)} = W_g^{(M)} (W_{gh}^{(M)} h_i^{(t)} + b_{gh})
\]

(11)

\[
d_x = W_x^{(M)} z_x^{(t)} = W_x^{(M)} (W_{gx}^{(M)} h_i^{(t)} + b_{gx})
\]

(12)

\[
b_g = W_g^{(M)} z_{gb}^{(t)} = W_g^{(M)} (W_{gb}^{(M)} h_{i-1}^{(M)} + b_{gb})
\]

(13)

Formula (11) represents the hidden layer output \(h_i^{(t)}\) of the dictionary feature LSTM through linear transformation \(W_g^{(M)} h_i^{(t)} + b_{gh}\), then multiplied by the weight matrix \(W_{gh}^{(M)}\) to get \(d_g h\). Along with \(h_i^{(M)}\) of LSTM, it affects the gate state calculation of the main LSTM. Equations (12) and (13) are calculated similarly to equation (11).

Domain dictionary can be used during the prediction phase of the word segmentation model or added during the training phase. Adding the domain dictionary in the prediction stage does not need to change the trained word segmentation model, which has better domain adaptability. The proposed word segmentation model transfers the field training corpus, extracts the word segmentation features of the field and improves the word accuracy rate. In the training of domain training corpus, the adding of domain dictionary features can improve word segmentation ability. Therefore, this model adopts the method of adding domain dictionary in the training stage. The influence of these two ways on the word segmentation is as follows. The completeness of a domain dictionary affects the ability of domain segmentation. This paper adopts the method in literature [20] to construct a dictionary of software engineering.

3. Experiments and analysis

3.1. Experimental environment and data

The experimental environment of this paper is as follows. Processor: Inter (R)Core (TM) i7-6850k, CPU@3.6ghz, Graphics acceleration card: NVIDIA GeForce GTX 1060ti 11 GB, Memory: 16GB, Operating system: Ubuntu 16.04 LTS (64bit), Google open source deep learning framework: Tensor-Flow 0.12, General domain deep learning neural network model: the super LSTM network.

The generic domain segmentation model is trained on the PKU training corpus provided by SIGHAN CWS BAKE OFF 2005, with the corpus size of 7548KB. The dictionary features use the stuttering word segmentation dictionary.

The professional field dictionary contains legal terms extracted from the tool library of CNKI and software engineering. It carries out relevant word extension in Baidu encyclopedia. Finally, it is normalized. And it keeps the sentences within three words. The dictionary contains 40420 entries in the field. The domain dictionary is added to the model during professional domain segmentation training. Accurate P, recall rate R and comprehensive index F1 can be used for word segmentation evaluation. F1 is as the main evaluation index.

3.2. Parameters setting

The network parameters obtained by LSTM network model training on PKU corpus are as the initial values. When training, we select 90 % of the sentences as the training set and the remaining 10 % as the verification set. When the iteration number is 42, the F1 value on the verification set is 97.07 %. So we select the network parameter at this time as the initial value. The length of the character vector is 100 dimensions, and the initial value of the character matrix corresponding to the newly emerged characters in the professional field is randomized and adjusted in the subsequent training.

The influence of super-parameters should be considered in the field training corpus. First, the learning rate (LR) is analyzed and the appropriate LR is determined. When the Dropout is set to 50 %, the regular term coefficient \(\alpha\) is 0.001, KL divergence coefficient is 0, LR is set to \(10^{-5}\), \(10^{-4}\) and \(10^{-5}\) respectively. The changes of F1 and loss function after each iteration are compared, as shown in Fig. 4a and Fig. 4b respectively.

Compared with the general neural network model, the model in this paper achieves the optimal performance with
fewer iterations, mainly because the initial values of network parameters are well trained in the general domain corpus, which makes the network performance quickly reach the best point in the professional domain corpus training.

Then we determine the appropriate Dropout and the regular term coefficient $\alpha$. Set the learning rate LR as $10^{-3}$, the regular term coefficient $\alpha$ as 0.001, and the KL divergence coefficient as 0. The comparison results are shown in Fig. 5. It can be seen from Fig. 5 that there is little difference in the performance with the three Dropout. The learning rate LR is set as $10^{3}$, the Dropout is 50 %, and the KL divergence coefficient is 0. The influence on the F1 value of validation set is analyzed when $\alpha$ is set as 0.001, 0.002, 0.005 and 0.01. The comparison results are shown in Fig. 6. As can be seen from Fig. 6, F1 value decreases with the increase of $\alpha$. Therefore, we choose $\alpha$=0.001.

Finally, the influence of KL divergence coefficient on proposed model performance is analyzed. The learning rate LR is set as $10^{3}$, the regular term coefficient $\alpha$ is 0.001, the Dropout is 50 %, and the KL divergence coefficient is set as 0, 0.2, 0.4 and 0.6, respectively.

3.3. Performance comparison with different word segmentation methods

Through experimental comparison and analysis, we discuss the influence of super-parameters on the model in this paper. The final super parameters in the professional field are set as follows: the learning rate is LR=$10^{-3}$, the regular term coefficient $\alpha$ =0.001, and the Dropout=50 %. Three hundred sentences are selected from the text of software engineering to form the testing set. We conduct comparison with other word segmentation methods. Table 1 displays the, P, R, F1 results with different methods containing proposed method (KL divergence coefficient is 0, training with domain dictionary), NLPIR segmentation method [21], BI-LSTM-CRF model, BI-LSTM-CRF model (prediction with domain dictionary), proposed model (KL divergence coefficient is 0, without domain dictionary), proposed model (KL divergence coefficient is 0.2, without domain dictionary).

By comparing and analyzing the segmentation results in table 1, we can obtain that: The KL divergence coefficient is used to constrain the parameters training in the
Table 1. Result comparison with different methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>P/%</th>
<th>R/%</th>
<th>F1/%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed model(KL=0, training with domain dictionary)</td>
<td>95.19</td>
<td>94.62</td>
<td>94.91</td>
</tr>
<tr>
<td>NLPIR</td>
<td>78.69</td>
<td>85.93</td>
<td>82.15</td>
</tr>
<tr>
<td>BI-LSTM-CRF</td>
<td>86.48</td>
<td>89.32</td>
<td>87.88</td>
</tr>
<tr>
<td>BI-LSTM-CRF with domain dictionary of prediction</td>
<td>89.51</td>
<td>91.88</td>
<td>90.68</td>
</tr>
<tr>
<td>Proposed model(KL=0, without domain dictionary)</td>
<td>94.13</td>
<td>94.47</td>
<td>94.31</td>
</tr>
<tr>
<td>Proposed model(KL=0.2, without domain dictionary)</td>
<td>92.36</td>
<td>92.53</td>
<td>92.44</td>
</tr>
</tbody>
</table>

professional domain model with the general domain segmentation information to prevent over-fitting. However, the results of word segmentation on the test set show that the word segmentation results with the KL divergence coefficient have the same wrong word segmentation results as those in the NLPIR and BI-LSTM-CRF models. It decreases by 1.86 % compared to those without KL divergence coefficient. Because the distribution of word segmentation features in professional field training set is quite different from that in the general field, while the KL divergence coefficient makes the word segmentation model learn the general field features, resulting in the error of word segmentation results in the test set.

4. Conclusion

This paper proposes a Chinese word segmentation model based on deep learning in view of the poor word segmentation performance in professional fields. Meanwhile, it combines large-scale general domain word segmentation data, small-scale professional field training corpus and domain dictionary to improve the word segmentation ability and realize domain adaptive word segmentation. Taking the word segmentation task in the field of software engineering as an example, this paper proposes the word segmentation criterion for the field terms, carries out small-scale manual labeling, and designs the word segmentation experiment to compare the performance of each word segmentation model. The results show that the proposed word segmentation model in this paper can effectively improve the performance of the cross-domain word segmentation system. In the future, we will further study the proposed method and propose modified methods to improve the word segmentation efficiency.

References


Professional Caregivers’ Mental Health Problems and Burnout in Small-Scale and Traditional Long Term Care Settings for Elderly People With Dementia in The Netherlands and Belgium. *Journal of the American Medical Directors Association*, 13(5):486.e7–486.e11, 2012.


