Tibetan word segmentation method based on CNN-BiLSTM-CRF model

Lili Wang[2], Hongwu Yang[1; 2; 3],Xiaotian Xing[2], Yajing Yan[2]
1College of Educational Technology, Northwest Normal University, Lanzhou 730070, China
2College of Physics and Electronic Engineering, Northwest Normal University, Lanzhou 730070, China
3National and Provincial Joint Engineering Laboratory of Learning Analysis Technology in Online Education, Lanzhou 730070, China
Email:yanghw@nwnu.edu.cn

***Abstract*—We propose a Tibetan word segmentation method based on CNN-BiLSTM-CRF model that merely uses the characters of sentence as the input so that the method does not need large-scale corpus resources and manual features for training. Firstly, we use convolution neural network to train character vectors. Then the character vectors are searched through the character lookup table to form a matrix C by stacking searched results. Then the convolution operation between the matrix C and multiple filter matrices is carried out to obtain the character-level features of each Tibetan word by maximizing the pooling. We input the character vector into the BiLSTM-CRF model, which is suitable for Tibetan word segmentation through the highway network, for getting a Tibetan word segmentation model that is optimized by using the character vector and CRF model. For Tibetan language with rich morphology, fewer parameters and faster training time make this model better than BiLSTM-CRF model in the performance of character level. The experimental results show that character input is sufficient for language modeling. The robustness of Tibetan word segmentation is improved by the model that can achieves 95.17% of the F value.**

Keywords-Convolutional Neural Network; recurrent neural network; Conditional random field; Tibetan word segmentation

I. INTRODUCTION

Word segmentation is not only the most basic but also the most important part in natural language processing (NLP). Word segmentation provides important feature information of advanced NLP tasks involving topic recognition, topic tracking, information retrieval, machine translation and public opinion analysis. In the past, the rule-based method, the statistic-based method and the combination of rule-based method with statistic-based method are commonly used for word segmentation. In recent years, researchers have proposed many deep neural network-based methods in Chinese word segmentation and have achieved good results [1]. However, there are few studies on minority languages, such as Tibetan dialects that are usually not easy to get training corpus. Tibetan is a national language with a long history in China. It is widely used in Tibet, Qinghai, Gansu, Sichuan, Yunnan province as well as the part of Nepal, Bhutan, Pakistan and India. Not only the population of Tibetans is very large, but also the geographical location is widely distributed. Tibetan inherits and records the rich and colorful Tibetan culture. Tibetan belongs to the Tibetan-Burmese branch of the Sino-Tibetan Language Family. Tibetan is also a kind of Pinyin character. When sorting the letters, there are strict rules that should be written from left to right and from top to bottom. The basic syllable is the core position of each character, which is used to determine the central consonant position of the character. Tibetan characters are segregated by syllable point, but there is no segregation mark between words. An example of Tibetan word structure is shown in Figure 1. The word, expressing independent meaning, is the smallest language component. Only at the word level can we improve the certainty of Tibetan processing. In this way, a good foundation for Tibetan intelligent analysis is needed. At present, studies on Tibetan word segmentation are usually based on rule-based method [2], statistic-based method [3], or the combination of these two methods [4]. However, most of them rely on artificial features, which is time-consuming and laborious. Therefore, a hybrid deep learning model based on CNN-BiLSTM-CRF is proposed for Tibetan word segmentation. Firstly, we use the convolutional neural network (CNN) to capture the character-level feature vectors of Tibetan words. Then we input character-level feature vectors into the highway network to train the context-dependent information as the input of bi-directional long short-term memory (BiLSTM) network. In this way, we can acquire the implicit semantic features between sentences and words. Finally, the optimal probability distribution is obtained by conditional random field (CRF) layer.

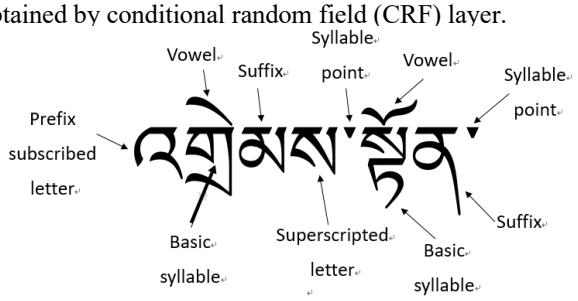

Figure 1. An example of the structure of Tibetan word.

II. ARCHITECTURE OF CNN-BILSTM-CRF MODEL

Proposed Tibetan word segmentation based on CNN-BiLSTM-CRF model is mainly composed of BiLSTM module, CNN module and CRF module. The first layer is the input layer. In this module, Tibetan words are used as the current input. The hidden state of the previous moment is utilized to predict the next Tibetan words. In the process of character embedding search, the found character vectors are stacked together to form a matrix C. Then a convolution operation is conducted between the matrix C and multiple filter matrices. In this layer, we use 12 filters, three filters with 2 widths, four filters with 3 widths and five filters with 4 widths. The character level features of each word are obtained by maximizing the pooling for inputting into the highway network. The output of the highway network is the input of BiLSTM that is the

second layer of the neural network module. Finally, the third layer of CRF module is used to decode the output of the second layer into an optimal probability distribution sequence. The architecture of the neural network in this paper is given in Figure 2.

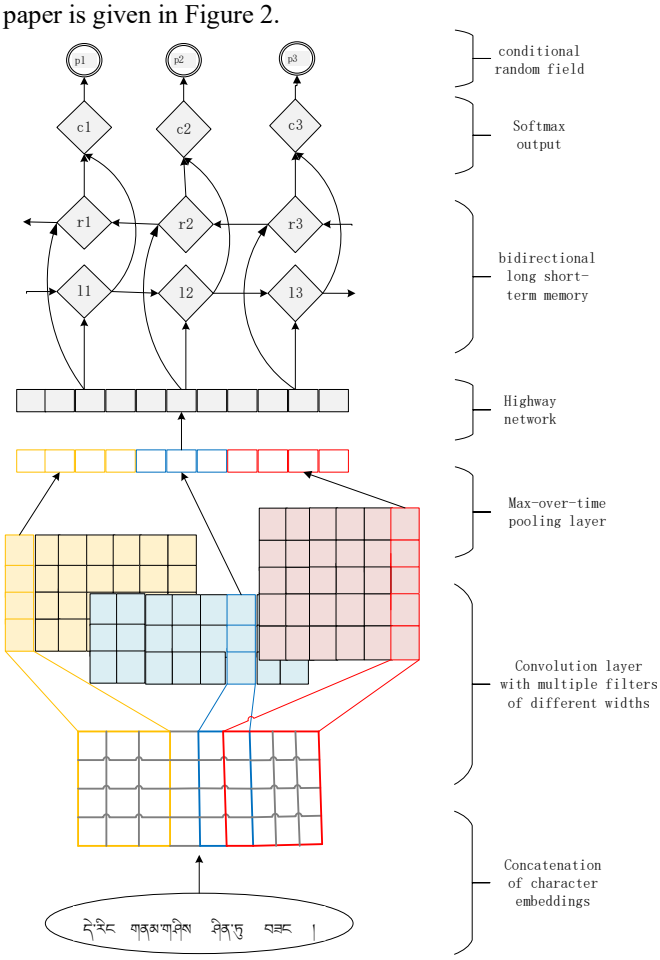

Figure 2. The flow chart of Tibetan word segmentation

A. *Character vector feature*

There are typical stick-in forms in Tibetan. The stick-in forms are commonly named as stick-in case, contraction word and contraction case. The stick-in forms mainly consist of various case markers, auxiliary words, conjunctions and end words. The syllables without postscript or with a postscript of ཨ are added to the stick-in forms, which forms a new character. This form seems like a new character. However, it is actually a condensation form of two characters, which is caused by spelling. In the meantime, the complex stick-in form is the most significant reason that affects the accuracy of word segmentation.

Through some changes such as rewriting, combination and auxiliary symbols, there are 41 consonant letters and 15 vowel letters in the contemporary Tibetan language. Besides, some numeric characters, punctuation symbols, textual modification symbols and other symbols have also been included in the Tibetan text. According to the combination of prefix subscribed letter, top-addition, bottom-addition, suffix and superscription letter, the letters forms different Tibetan characters. Tibetan characters appear in different positions such as the beginning, middle and end of a word, which form distinctive Tibetan words. According to the Chinese-Tibetan Contrast Dictionary, there are approximately more than 80,000 words. In order to better reflect the local information, the CNN model is used to obtain the character feature vector.

B. *Convolution Neural Network*

In convolution neural network [7], the local feature information of text data can be extracted by the convolution layer. Besides, the most representative part of the local feature information can be extracted as vectors by using convolution layer and maximum pooling layer. Existing studies have shown that CNN can extract morphological information (such as prefix or suffix) from the characters of words for encoding to form the character feature vector. [8] extracted character-level features by CNN that has achieved good results in the field of named entity recognition. Therefore, this paper uses CNN to extract the characteristic features of Tibetan words with rich morphology.

By using CNN network and highway network, we can effectively reduce model parameters and training time. At the same time, the performance of the Tibetan word segmentation can be improved effectively by the proposed method. The structure of the CNN model is shown at the bottom of Figure 2. It consists of a character vector table, convolution layers and pooling layers.

The character vector table contain totaling 64 characters that contains 41 consonant letters, 15 vowel letters, 8 punctuation marks and an uncertain character which is not in the character set. The corresponding character vector table is generated by 64 characters respectively. The function of the character vector table is to convert each character of words into a corresponding character vector to form the corresponding character vector matrix of Tibetan words. Because of the various length of Tibetan words, the size of the generated character vector is different. This paper takes length of the longest Tibetan word as the length of character vector and uses placeholder to equal the length of each character vector. Then all character vectors are superimposed to form matrix C. Convolution operation is conducted between matrix C and multiple filter matrices. In this layer, we use 12 filters, three filters with 2 widths, four filters with 3 widths and five filters with 4 widths. The character-level features of each word are obtained by maximizing the pooling and then are inputted into the highway network. In the training process of convolution neural network, character vector table updates character vector matrix automatically through back propagation mechanism.

C. *Highway Network Layer*

The propagation equation of traditional neural networks (ignoring bias and layer index) is as follows:

$$y = H(x, W_H) \tag{1}$$

Where H is a no-linear function, W is a weight, x is input, and y is the output. The propagation equation of highway networks [9] is as follows:

$$y = H(x, W_H) \cdot T(x, W_T) + x \cdot C(x, W_C) \tag{2}$$

Where T is transform gate, C is carry gate. When $C = 1 - T$, equation (2) is that:

$$y = H(x, W_H) \cdot T(x, W_T) + x \cdot (1 - T)(x, W_C) \tag{3}$$

Among them, the dimension of X , Y , $H(x, W_H)$, $T(x, W_T)$ must be the same. If the dimension is not enough, it needs to fill with zero, so we can get:

$$y = \begin{cases} x, & \text{if } T(x, W_T)=0 \\ H(x, W_H), & \text{if } T(x, W_T)=0 \end{cases} \quad (4)$$

That is to say, when the gate is 1, all the original x is output without activation. When the mean value of the gate is 0.5, half of all the original information input into the next layer is activated, which retains a lot of information. Moreover, in the process of back-propagation, more information gradient can directly return to the input layer without a non-linear transformation.

D. BiLSTM model

Long short-term memory [10] is a time recursive neural network, which can model long-distance dependent information effectively. The information is input from the input gate and then flows into the circulating cell unit which controls whether the information flows to the input gate or to the forgetting gate. The calculation equation s of each unit at each time are presented in the equations as follows:

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i]) \quad (5)$$

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \quad (6)$$

$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o) \quad (7)$$

$$c_t = f_t \times c_{t-1} + i_t \times (\tanh(w_c \cdot [h_{t-1}, x_t] + b_c)) \quad (8)$$

$$h_t = o_t \times \tanh(c_t) \quad (9)$$

Where, i_t , f_t , o_t , c_t respectively represent the outputs of input gate, forgetting gate, output gate and cell state at t time, h_t and x_t stand for hidden layer vectors and input vectors at t time respectively. σ represent the sigmoid activation function and it can output values between 0 and 1 to describe how much each part can pass. 0 stands for "no quantity is allowed to pass" and 1 stands for "any quantity is allowed to pass". ω and b stand for the weight matrix and bias vector respectively.

In spite of the good performance of LSTM network in Tibetan word segmentation, the model advances from left to right, which makes the weight of the front words smaller than that of the latter words. However, the weight of each word in the sentence should be the same for Tibetan word segmentation. Therefore, in order to obtain more accurate context information of Tibetan words, we use BiLSTM model. This model combines forward LSTM and backward LSTM models. In addition to the usage of previous input features and sentence-level markup information, it can also use future input features. The h_t can be expressed by equation (10).

$$h_t = \overrightarrow{h_t} \cup \overleftarrow{h_t} \quad (10)$$

E. CRF model

CRF layer, which is the decoding layer, mainly corrects the results of Tibetan word segmentation predicted by softmax. Although the context information is obtained, the output results are independent of each other. Softmax classifier merely selects a label output with the maximum probability, but it will produce sentence with incorrect grammatical structure. Hence, CRF layer is used to correct these errors.

The input of CRF layer is the output of softmax. CRF layer is a matrix P of $n*m$, where n is the number of Tibetan words and m is the type of label. p_{ij} is defined as a transition score matrix from the i tag to the j tag. For a predicted tag sequence $y = y_1, y_2 \ldots y_n$, probability distribution is shown in equation (11).

$$score(x, y) = \sum_{i=1}^{n} p_{i,y_i} + \sum_{i=1}^{n+1} A_{y_{i-1}, y_i} \quad (11)$$

The whole sequence probability consists of two parts. One is the P matrix output from BiLSTM layer, the other part is the transfer matrix A of CRF layer. y_0 and y_n in equation (11) are respectively the end and beginning markers of predictive sentences.

The likelihood function of maximizing markers for a training sample x, y^x in the training of CRF layer is shown in equation (12).

$$\log(p(y^x \mid x)) = score(x, y^x) - \log(\sum_{y'} e^{score(x, y')}) \quad (12)$$

Where y' represents the real markup value. In the prediction process, Viterbi algorithm of dynamic programming is used to solve the optimal path. As shown in equation (13).

$$y^* = \arg\max_{y'} score(x, y') \quad (13)$$

III. TRAINING OF THE NEURAL NETWORK

A. Word vector

In the experiment, 64 Tibetan characters and punctuation symbols were randomly distributed to initialize the character vector query table. The dimension of the character vector was 10, and its value range was [-0.5 0.5].

B. Optimization

At present, popular optimization algorithms in neural networks are SGD, Momentum, Adagrad, Adadelta, RMSprop, Adam, Adamax, etc. [13-14]. In this paper, SGD algorithm is utilized to optimize the model. The experimental results show that the performance of the model can be improved by SGD optimization algorithm. We set the learning rate as η_0, set the initial value as 0.001 and set the momentum as 0.9. The learning rate η_0 is updated automatically by equation $\eta_t = \eta_0 / 1 + \rho_t$ in each training cycle. Among them, we set the delay rate as ρ_t=0.5, and t as the number of training cycles that have been completed already.

C. Dropout parameter

Dropout [15] is a very useful technique in regularization methods. Generally speaking, it will delete some neurons randomly and train different neural network structures according to different batches. In the experiment, the value of Dropout and its position in the model are very important, which directly affects the performance of the model. In most neural networks, Dropout value is 0.5, which can effectively prevent over-fitting. However, in this experiment, due to the limited training data, the model

is cross-validated with different Dropout values. The experimental results show that the best recognition effect is achieved when the Dropout value is 0.3. In this paper, the parameters of the neural network are set as showed in Table 1.

TABLE I. PARAMETER SETTING OF CNN-BILSTM-CRF MODEL

parameter	value	parameter	value
Character vector dimension	10	Minimum frequency of Tibetan words	4
Character Feature Window	5	Number of convolution kernels	128
learning rate	0.01	Depth of LSTM	128
Dropout	0.3	Layer Number of LSTM	2

IV. EXPERIMENTS AND RESULTS ANALYSIS

A. Experimental Data Set

Due to the lack of publicly annotated data set for Tibetan word segmentation, a Tibetan word segmentation data set is established to verify the effectiveness of CNN-BiLSTM-CRF model in Tibetan word segmentation. The corpus of this data set is manually extracted from the Tibetan Language Network of China, Qinghai Tibetan Language Network, Kangba Satellite TV and other websites. It comes to a grand total of 3000 Tibetan articles which mainly include various news, celebrity anecdotes, novels and other topics. Afterwards, the text is filtered and proofread manually. We divide the text into sentences and then cut them into words by maximum matching method. Finally, the corpus containing 125 386 Tibetan sentences are obtained. It includes a total of 1196 907 Tibetan words. In the experiment, 80% of the sentences is randomly selected as the training set, 10% is used as the verification set, and the remaining 10% is used as the test set. In Table 2, we use a short statistic data set. The out-of-vocabulary (oov) refers to the words which has nothing to do with the train data set.

TABLE II. SETTING OF EXPERIMENTAL DATA SET

train	sent	20793
	word	281960
dev	sent	2607
	word	35200
	oov	985
test	sent	2828
	word	36772
	oov	1139

In the experiment, we use the recall rate, precision rate and F value of Tibetan words as the evaluation index of the model. In order to evaluate the effectiveness of the proposed model, we conduct a series of feature experiments on the test hierarchy.

B. Analysis of experimental results

Five groups of experiments are carried out under the above corpus to verify the effectiveness of CNN-BiLSTM-CRF model in Tibetan word segmentation. Among them, the recall rate, precision rate and F value are used.

Experiment 1. This experiment has two purposes. One is to test the performance of CRF model on the data set by using the CRF model as a benchmark model; the other is to summarize a series of problems found in Tibetan word segmentation using CRF model. In the experiment, CRF++ toolkit [16] is used which is popular at present. Since the tagging dataset is based on sentences, we merely consider the lexical features for CRF++ toolkit. After using CRF toolkit for word segmentation tasks, we find some problems.

Firstly, Tibetan words which are not included in the training corpus can't correctly be recognized by the CRF model.

Secondly, there are plenty of stick-in forms in Tibetan, but CRF model can't recognize the compact form. It seems like a new character, but it represents two words. For example, "ངའི་" (my) in "ངའི་ཨ་མ།" (my mother) is stick-in form. It is a stick-in form formed by word and genitive markers (ཡི་), which CRF model cannot accurately identify.

Thirdly, the lack of clear segmentation principles and uncertain segmentation units in Tibetan word segmentation has resulted in incomplete meaning and unreasonable grammatical structure. For example, the separation of "ནམ་གསལ་ལ་མ་གསལ་ལང་ནས་སོང" and "གསལ་ལ་ལམ་མ་གསལ" in "དྲམར་རམ་མེ་དམར" (The sky has dawned) is difficult to understand, which leads to unreasonable grammatical structure.

Fourthly, there are a lot of long words in Tibetan, especially some extended place names and long organization names, which make CRF model unable to recognize similar long words accurately. For example, "ཞིན་ཙང་ཡུ་གུར་རིགས་རང་སྐྱོང་ལྗོངས།" (Xinjiang Uygur Autonomous Region).

Finally, part of Tibetan corpus collection comes from some forums, microblogs, news and other networks. There are a lot of spelling errors in this text. CRF model can't recognize the misspelled words accurately. Its accuracy rate is 89.97%, recall rate is 91.01% and F value is 90.49%.

Experiment 2. This experiment has two purposes. The one is to study whether the advantages of deep neural network are more than those of the statistical model in Tibetan word segmentation. The other is to study whether the problems found in the CRF statistical model can be solved by the deep neural network.

In the experiment, we use simple RNN model, LSTM model and BiLSTM model to perform Tibetan word segmentation tasks. From Table 3, we can see that the performance of simple RNN model is almost the same as that of CRF model. The performance of LSTM model and BiLSTM model are better than that of CRF model. The F value of BiLSTM model is 2.31% which is higher than that of CRF model.

TABLE III. EXPERIMENTAL RESULTS OF BASELINE NEURAL NETWORK MODEL

model	P (%)	R (%)	F (%)
CRF	89.97	91.01	90.49
RNN	90.46	90.59	90.52
LSTM	90.12	91.74	90.92
BiLSTM	93.12	92.49	92.80

Experiment 3. The purpose of this experiment is to determine whether we add CRF model to LSTM model, and whether BiLSTM model can further improve the performance of Tibetan word segmentation. On the basis of experiment 2, the output of BiLSTM model is input into CRF layer, and the output sequence with the greatest probability is obtained. The experimental results in Table

4 show that the accuracy of LSTM-CRF model and BiLSTM-CRF model are improved after adding CRF layer, and the F value of BiLSTM-CRF model is 1.31% which is higher than that of BiLSTM model.

TABLE IV. EXPERIMENTAL RESULTS AFTER ADDING CRF MODELS

experimental design	P (%)	R (%)	F (%)
LSTM_CRF	93.43	92.56	92.99
BiLSTM_CRF	94.33	93.89	94.11

Experiment 4. The purpose of the experiment is to verify the validity of the CNN model.

CNN model is added to LSTM-CRF model and BiLSTM-CRF model. We use CNN model to acquire character features which are input into LSTM or BiLSTM model for training. The experimental results in Table 5 demonstrate that the performance of Tibetan word segmentation is improved when CNN network is concatenated to LSTM-CRF model and BiLSTM-CRF model. At the same time, we can see that the F value of CNN-LSTM-CRF model is 0.41% which is higher than that of LSTM-CRF model, and the F value of CNN-BiLSTM-CRF model is 1.06% which is higher than that of BiLSTM-CRF model.

TABLE V. COMPARISON OF EXPERIMENTAL RESULTS AFTER CONCATENATING CNN MODEL

experimental design	P (%)	R (%)	F (%)
CNN-LSTM-CRF	93.62	93.18	93.40
CNN-BiLSTM-CRF	94.71	95.64	95.17

Experiment 5. The purpose of the experiment is to verify the validity of the CNN model. Our operating system is CentOS Linux release 7.6.1810 (Core), Intel Xeon E5-2620 CPU, and 12GB graphics card is Tesla K40C GPU. With the concatenation of CNN model and highway model, the training speed of the model has been significantly improved, which can be observed in the experimental results in Table 6. The training speed of CNN-BiLSTM-CRF model with CNN model and highway model has been improved significantly, compared with BiLSTM-CRF model. The training speed is increased by 2.5 hours and the test speed is increased by 1.06 seconds. Regardless of the accuracy of word segmentation, the test is to calculate the time spent on segmenting the Tibetan text by loading the models that have been trained respectively.

TABLE VI. THE INFLUENCE OF CNN MODEL ON EXPERIMENTAL SPEED

experimental design	training time（h）	testing time（s）
LSTM-CRF	10.7	3.05
BiLSTM-CRF	11.5	3.18
CNN-LSTM-CRF	8.5	2.36
CNN-BiLSTM-CRF	9.0	2.12

V. CONCLUDING REMARKS

This paper takes the traditional CRF statistical model as the benchmark to perform experiments. We summarize the problems in Tibetan word segmentation to construct a neural network model based on CNN-BiLSTM-CRF framework. The model captures character level feature vectors in CNN layer, obtains the past and future context information of current words in BiLSTM layer, decodes the output of BiLSTM layer in CRF layer, and finally outputs the optimal tag sequence. Experiments based on the constructed corpus further verify the effectiveness of the CNN-BiLSTM-CRF framework for Tibetan word segmentation. The Tibetan text classification and the Tibetan speech synthesis system which combined with the word segmentation results achieved the best effect. Subsequently, we will further improve the corpus, add part-of-speech tagging feature information, and test the performance of Tibetan word segmentation and part-of-speech tagging based on the original corpus.

ACKNOWLEDGMENT

This research has received funding from the academic requirements for the National Science Foundation of China (NSFC) under grant No.11664036, No.31860285 and No.31660281 and High School Science and Technology Innovation Team Project of Gansu (2017C-03).We also want to thank the reviewers for their thoughtful comments and efforts towards improving our paper.

REFERENCES

[1] D. Cai and H. Zhao, “Neural word segmentation learning for chinese,”vol. 1, 06 2016, pp. 409–420.

[2] L. Huidan, N. Minghua, Z. Weina, W. Jian, and H. Yepin, “Segta practical tibetan word segmentation system,” journal 0f chinese information processing, vol. 26, no. 1, pp. 97–104, 2012.

[3] K. Caijun, L. Congjun, and J. Di, “Segmentation of tibetan abbreviated forms based on word position,” Computer Engineering and Applications, vol. 50, no. 11, pp. 218–222, 2014.

[4] L. Karten, Y. Yuanyuan, and Z. Xiaobin, “Tibetan automatic word segmentation based on conditional random fields and knowledge fusion,”journal 0f chinese information processing, vol. 29, no. 6, pp. 213–219, 2015.

[5] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, “Character-aware neural language models,” Computer Science, 2015.

[6] Long congjun and liu huidan, Research on the Theory and Method of Tibetan Automatic Word Segmentation. Intellectual Property Publishing House, 2016.

[7] F. Dernoncourt, J. Young Lee, and P. Szolovits, “Neuroner: an easy-touse program for named-entity recognition based on neural networks,”05 2017.

[8] X. Ma, “End-to-end sequence labeling via bi-directional lstm-cnns-crf,”03 2016.

[9] R. Kumar Srivastava, K. Greff, and J. Schmidhuber, “Highway networks,”05 2015.

[10] Y. Yao and Z. Huang, “Bi-directional lstm recurrent neural network for chinese word segmentation,” 02 2016.

[11] C. Kang, D. Jiang, and C. Long, “Tibetan word segmentation based on word-position tagging,” 08 2013, pp. 239–242.

[12] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” International Conference on Learning Representations, 12 2014.

[13] Z. Yang, R. Salakhutdinov, and W. Cohen, “Multi-task cross-lingual sequence tagging from scratch,” 03 2016.

[14] T. Mikolov, M. Karafi´at, L. Burget, J. Cernock´y, and S. Khudanpur,“Recurrent neural network based language model,” vol. 2, 01 2010, pp.1045–1048.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” Journal of Machine Learning Research, vol. 15, pp. 1929–1958, 06 2014.

[16] Y. Leng, W. Liu, S. Wang, and X. Wang, "A feature-rich crf segmenter for chinese micro-blog," vol. 10102, 12 2016, pp. 854-861.