Design Engineering

ISSN: 0011-9342
Issue: 11 | Pages: 68 - 80

Chinese Named Entity Recognition based on Transformer Encoder and BiLSTM

Xiaoran Guo[1], Ping Luo[2], Tiejun Wang[1], Weilan Wang[3,*]

[1]School of Mathematics and Computer Science Institute, Northwest Minzu University, Lanzhou, Gansu, China

[2]School of Electronic and Information Engineering, LanZhou JiaoTong University, Lanzhou, Gansu, China

[3]Key Laboratory of China's Ethnic Languages and Information Technology, Northwest Minzu University, Lanzhou, Gansu, China

*Corresponding Author: Weilan Wang

Abstract:

Transformer encoder, which is often used for feature extraction, is ineffective in named entity recognition tasks due to the damage of word embedding, the loss of position information and direction information. In this paper, a method of Chinese named entity recognition at character level based on BiLSTM, Transformer encoder and CRF is proposed, which improves the use of position vector in Transformer, splices word embedding and position vector as character representation layer to avoid loss of word embedding information and position information. Extracting context features and incorporating direction information into position vector through BiLSTM. Besides, Transformer encoder is introduced to extract inter-word relationship features, and finally CRF is used to decode globally. Experiments on universal MSRA and Thangka datasets achieves 81.4% F1 value and 88.3% F1 value respectively, and show that the method effectively improves the performance of Chinese named entity recognition.

***Keywords**: Named entity recognition, Transformer encoder, BiLSTM, Position coding.*

I. INTRODUCTION

Named entity recognition is a key research content in natural language processing, most of which are used to identify and extract names of people, places, institutions or distinguished name of special fields from unstructured texts for knowledge graph, automatic question answering, machine translation and so on [1].

Named entity recognition is a typical sequential labeling problem, in which each word in a sentence is tagged as a definite label by a model, and then successive words with the same label are jointly extracted as entity names. Currently, the main method is to extract text features using word embedding as the representation of words in the deep neural network, then access the model such as Conditional Random Fields (CRF) to make sentence-level label prediction using label transfer probability [2]. Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) are widely used in named entity recognition tasks [3-4].

Transformer [5] is an encoder-decoder frame proposed by Google in 2017 based on the self-attention mechanism, with strong parallel computing ability and long-distance feature capture ability, making it perform better than CNN and RNN in machine translation, pre-training language model [6], text summarization and other natural language processing tasks. Transformer encoder, which is often used for feature extraction, is ineffective in Chinese named entity recognition tasks due to the damage of word embedding information, the loss of position information and direction information.

Transformer encoder uses sine-cosine function to encode absolute position, and reflects the distance relationship of relative position by product of the two. But Hang Yan et al. [7] proved that this relative position relationship can not distinguish direction, and the characteristics of relative position information would disappear after self-attention calculation. Additionally, the initial input of Transformer is a simple addition of the pre-training word embedding and the position coding vector, which invisibly damages the pre-trained word embedding.

For resolving the above problems, in this paper, a Chinese named entity recognition method based on BiLSTM, Transformer and CRF is proposed, in which the pre-training word embedding and the position coding vector are spliced, instead of simply adding, so as to avoid the damage of the pre training word embedding information and the loss of position information in the subsequent processing; BiLSTM is introduced to expand the semantic of the word embedding combined with the context information, and the direction information is integrated into the position vector; then Transformer encoder is used to further extract the character relationship features; finally, the CRF model is used to output the global labeling results. BiLSTM-Transformer-CRF is hereinafter used to represent the proposed method. Experiments on the universal dataset MSRA and the domain dataset Thangka corpus show that the BiLSTM-Transformer-CRF method can significantly improve the recognition accuracy, recall rate and F1 value without relying on external resources and pre-training language models.

II. RELATED WORK

At this stage, the main methods of named entity recognition are statistical machine learning and deep learning, and the former applies statistical algorithms to language models to some extent to solve the problems of difficulty in template formulation and poor portability in rule-based methods by MEM, HMM, SVM and CRF commonly, which often relies heavily on feature selection since it is based on statistical model. Moreover, manual design of feature templates is not only time-consuming, but also easy to introduce errors due to insufficient prior knowledge.

In recent years, named entity recognition methods based on deep neural network have gradually become the mainstream [8-9], because neural networks can extract feature information from text without manual intervention, which improves the ability of feature expression and data fitting. RNN has a natural advantage over CNN in processing sequential data, but with the increase of sentence length, RNN is prone to gradient disappearance or gradient explosion. Hochreater et al. [10] proposed Long short-term memory (LSTM) alleviates such problems to a certain extent by using special gate mechanism and enhances the long-distance dependence ability of the model. In order to have the ability of bidirectional representation, Graves et al. [11] proposed bidirectional LSTM (BiLSTM), which can fully consider the context information of current words to model text. Huang et al. [12] applied BiLSTM-CRF model to sequence tagging task for the first time with remarkable effect, which gradually becomes a widely used named entity recognition model.

With the rapid development of deep neural networks, in 2017, the Google team proposed the Transformer model, whose core is the attention mechanism which is essentially a resource allocation model that focuses on the key points of things at a given time. After attention processing, a vector at a location contains not only the information of itself, but also the information related to other location elements, thus enriching the feature expression.

Studies have shown that the integration of attention mechanism in BiLSTM-CRF model further improves the performance of named entity recognition tasks. For example: Rei M et al. [13] used attention mechanism and combined vectors to effectively improve F1 values for entity recognition. Li Mingyang et al. [14] proposed a method that uses BiLSTM with self-attention, which increases the recall rate of social media named entity recognition by about 5%. Because the Transformer model can not capture the order relationship between words, Vaswani et al. [5] used sine-cosine function to encode absolute positions and reflected the distance relationship of relative positions by the product of the two. However, Hang Yan et al. [7] demonstrated that the position coding of [5] cannot distinguish directions, and the relative location information disappears after self-attention calculation. Therefore, Hang Yan et al. proposed a self-attention with direction and relative position information, which uses an improved Transformer encoder to code the character sequence to represent words, and splices it with an external pre-training word embedding, and then uses the Transformer encoder to model the context information at the word level again, which has achieved better results on English datasets, but poor improvement of the effect on Chinese datasets.

The structure of this paper is arranged as follows. In section 2, the method BiLSTM-Transformer-CRF proposed for Chinese named entity recognition at word level is described. In section 3, the comparative experiments and results are analyzed. Finally, in section 4 the conclusion and future research direction are proposed.

III. PROPOSED MODEL

Generally, words in Chinese text have complete meanings, but errors caused by improper word segmentation will negatively affect the task, while data sparsity at the word level and OOV (Out-of-vocabulary) problems will also lead to model over-fitting. Therefore, in this paper, all corpora are segmented according to Chinese character level and labeled according to BIO labeling strategy, i.e., the named entity heading is labeled “B”, the non-heading of entity is “I” and the non-entity word is “O”. Fig 1 shows a BiLSTM-Transformer-CRF model, which consists of four parts: (1) a character level representation layer; (2) a BiLSTM-based context encoding layer; (3) a character-to-character relationship feature extraction layer based on Transformer encoder; and (4) a CRF decoding layer.

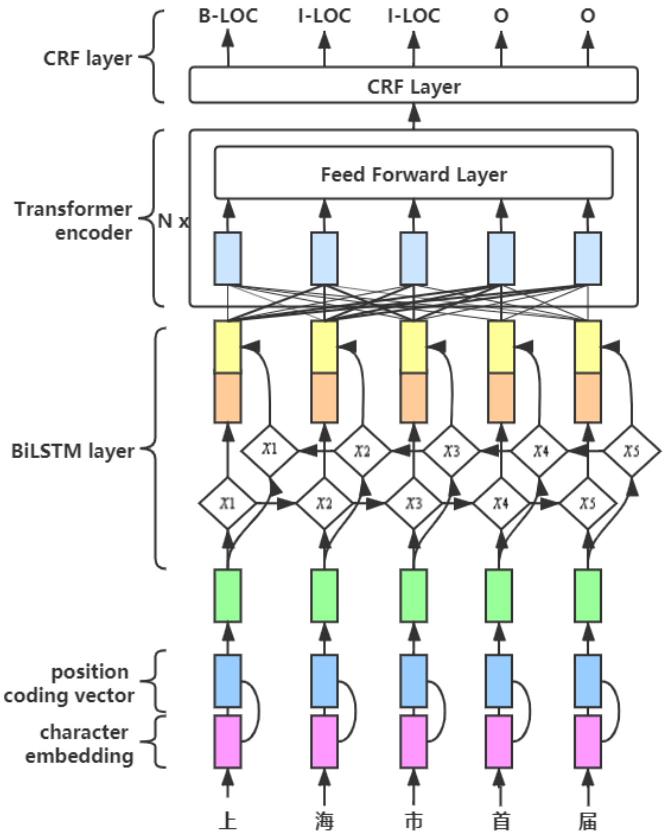

Fig 1: BiLSTM-Transformer-CRF model

3.1 Character Level Representation Layer

Text is first vectorized into a continuous dense vector of a certain length before it is fed into a deep neural network. Chinese character embedding refers to the word embedding represented by a single character, which is usually trained on language model by using large-scale unmarked corpus, the most famous of which is Word2Vec. The basic idea of Skip-gram model of Word2Vec is to predict the peripheral words in a certain window based on the central word, making a three-layer neural network to get the maximum probability as the training target and get the hidden layer parameters of the network. The character embedding trained by Skip-gram model can better reflect the correlation between Chinese characters.

In order to avoid the loss of position information in subsequent processing, an additional position vector is added in this paper to represent the order of each character in the sentence sequence with the same dimension as the character embedding, coded and represented in the manner of document [16], in which a position pos is mapped to a d_{model} -dimensional position vector, and i represents the position in the vector. It is calculated as follows according to the different parity:

$$PE_{(pos,i2)} = \sin(\frac{pos}{10000^{2i/d_{model}}}) \tag{1}$$

$$PE_{(pos,i2+1)} = \cos(\frac{pos}{10000^{2i/d_{model}}}) \tag{2}$$

Because character embedding and position vectors represent completely different meanings, their respective performance will be impaired by simple addition, so in this paper, the joint vectors which are spliced together by character embedding and position vectors are used as the representation of input words.

3.2 BiLSTM-based Context Encoding Layer

LSTM uses a specially designed gate mechanism and memory cells to process the input data, which reduces the risk of RNN gradient disappearance or gradient explosion and improves the network's long-range dependence. A typical LSTM unit structure is shown in Fig 2.

Within the unit of LSTM, input, forgetting and output gates control the flow of information in the hidden layer state. For each gate, the input is the current input x_t and the hidden layer state h_{t-1} of the previous moment, and the output is calculated by the full connection layer whose activation function is $sigmoid$, as shown in formulas (3) to (5). The forgetting gate f controls whether the cell state of the previous moment is forgotten with a certain probability. The input gate i processes the new input information and creates a new candidate value g_t for the state of cell. The results of the forgetting gate and the input gate can update the cell state c_t. Ultimately, the hidden layer state h_t of the current moment is determined by both the output gate and the memory cell state. The calculation formulas are (6) to (8):

$$f_t = \sigma(W_f[h_{t-1}; x_t] + b_f) \tag{3}$$

$$i_t = \sigma(W_i[h_{t-1}; x_t] + b_i) \tag{4}$$

$$o_t = \sigma(W_o[h_{t-1}; x_t] + b_o) \tag{5}$$

$$g_t = \tanh(W_c[h_{t-1}; x_t] + b_c) \tag{6}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes g_t \tag{7}$$

$$h_t = o_t \otimes \tanh(c_t) \tag{8}$$

Where, σ is $sigmoid$ function; tanh is activation functio; $\otimes$ is dot product of vector matrix; $W_f, W_i, W_o, b_f, b_i, b_o$ are weight matrices and bias terms of forgetting, input and output gates at the moment of t.

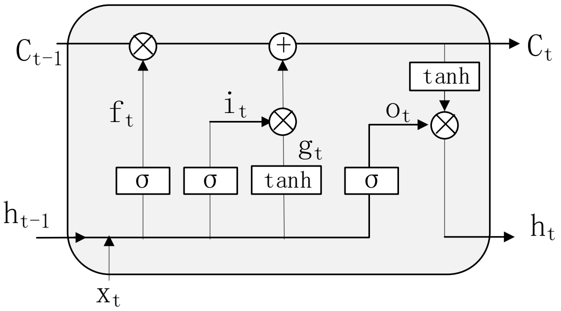

Fig 2: LSTM unit

BiLSTM network, consisting of forward and backward LSTM, can model sentence sequences in two directions, taking full advantage of contextual information and helping to label and identify named entities compared to one-way LSTM. Assuming the length of the sentence sequence is n and the input joint vector sequence is $S=(e_1,e_2,...,e_n)$, the forward and reverse eigenvectors $p=(p_1,p_2,...,p_n)$ and $q=(q_1,q_2,...,q_n)$ are obtained simultaneously through the BiLSTM layer, and finally the the forward and backward feature representation $z_i=(p_i;q_i)$ is obtained by position splicing.

The function of BiLSTM is to extend the character embedding semantically by combining the context information of the current word in the corpus, and to incorporate the direction information into the position vector to facilitate the subsequent Transformer processing. LSTM is more like a Markov decision process in general, and it is more difficult to use global information. Therefore, it is necessary to use a Transformer encoder to further obtain the global features of text after BiLSTM extracts context features.

3.3 Chinese Character-to-character Relationship Feature Extraction Layer based on Transformer Encoder

Transformer encoder consists of several stacks of layers, each of which has a multi-head self-attention layer, a feed forward neural network layer, and an add & normalization layer running through it, the core of which is the attention mechanism, as shown in Fig 3.

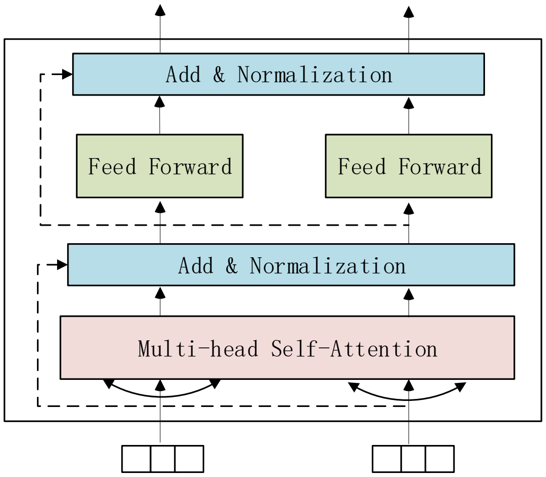

Fig 3: Transformer coding layer

Self-attention is an association calculation for different locations within a sequence. Specifically, it maps input information linearly to three different spaces, establishes a query and scoring mechanism, calculates the degree of correlation between words in a sentence, and makes the model pay more attention to words carrying important information by giving higher weights to important word. Assuming that the input represented by a joint character vector is a matrix $Z \in R^{n \times d}$, n is a sequence length , d is the dimension of the input, Z is mapped to different spaces Q, K, V by three different weight matrices W_q, W_k and W_v, and the dimension of the weight matrix is $R^{d \times d_k}$, the attentions are calculated using the scaling point product as follows:

$$Q, K, V = ZW_q, ZW_k, ZW_v \tag{9}$$

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{10}$$

Where, d_k is the dimension of self-attention layer; $\sqrt{d_k}$ acts as a zoom to prevent too big inner product QK^T . In order to capture features from multi-angles and multi-layers, Transformer introduces multi-head attention, passes through self-attention several times without sharing parameters (W_q, W_k, W_v), and finally splices the results. The calculation method is shown in formulas (11) and (12).

$$head^{(h)} = Attention(HW_q^{(h)}, HW_k^{(h)}, HW_v^{(h)}) \tag{11}$$

$$MultiHead(H) = [head^{(1)}; \ldots; head^{(m)}] \tag{12}$$

Where, h is the number of attention head, ranging $[1, m]$. Generally, $d_k \times m = d$, so the output dimension of above processing is still $R^{l \times d}$.

The above output is fed into the feed forward layer after add and data normalization, and is calculated as follows:

$$FFN(x) = \max\left(0, xW_1 + b_1\right)W_2 + b_2 \quad (13)$$

Where, W_1, W_2, b_1, b_2 are network parameters of feed forward layers, and $W_1 \in R^{d \times d_f}$, $W_2 \in R^{d_f \times d}$, $b_1 \in R^{d_f}$ and $b_2 \in R^{d}$. In order to better train deep networks, add and normalization layer are used again after the feed forward layer.

In this paper, the Transformer encoder is mainly used to complete the calculation of the relation between Chinese characters in a sentence, so that the named entity recognition model can identify the relation between characters and the importance of each character, so as to obtain more global feature information. Transformer's scalable multilayer structure also facilitates the extraction of features at different levels.

Relative position and direction information between words is very important for named entity recognition, for example, the sentence: Shao Yifu Foundation, founded by Shao Yifu, funded a new library of HK$5 million for 30,000 square meters, in Chinese is “Shao Yifu chuang li de Shao Yifu ji jin hui zi zhu 500 wan gang yuan jian she 3 wan ping fang mi de xin tu shu guan”. Each element in the sentence is actually a Chinese character, which is shown in pinyin. There is no obvious word boundary in Chinese, and the words before “chuang li” are likely to be a personal name and the words before “ji jin hui” are likely to be an institutional name. In addition, the entity name should be a continuous character, and the “Shao Yifu” at the beginning of the sentence and the “ji jin hui” in the sentence will not constitute an entity. Although the original Transformer has added the position coding, the relative position information will disappear after self-attention, so it is unable to capture these important position information. In this paper, the purpose of adding position vector in character representation layer and integrating into direction information through BiLSTM is to help Transformer encoder acquire global characteristics while retaining relative position and direction information between Chinese characters.

3.4 CRF Decoding Layer

There are strong constraints between context labels in named entity recognition, for example, two “B” labels cannot appear next to each other. As a discriminant model based on conditional probability, CRF models the target sequence on the basis of given input sequence, and can get the globally optimal labeling result according to the output of the previous layer and considering the relationship between contextual tabs. Assuming that the output of Transformer, that is, the input sequence of CRF, is $R = (r_1, r_2, ..., r_n)$, and one of the possible predictive label sequences is $y = (y_1, y_2, ..., y_n)$, the scoring function is defined as:

$$s(R, y) = \sum_{i=0}^{n} T_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i} \quad (14)$$

Where, T is transfer matrix; $T_{i,j}$ is transition probability from label i to label j; P_{i,y_i} is score of the $y_i - th$ label of the character. The probability of y appearing in all possible prediction is:

$$p(y|R) = \frac{e^{s(R,y)}}{\sum_{\tilde{y}\in Y_r} e^{s(R,\tilde{y})}} \tag{15}$$

During CRF training, the loss function by the maximum likelihood estimation is as follows:

$$L = s(R,y) - \log\left(\sum_{\tilde{y}\in Y_r} e^{s(R,\tilde{y})}\right) \tag{16}$$

In the prediction, the candidate with the highest probability is chosen according to the well trained parameters as the final result.

IV. EXPERIMENTS AND RESULTS ANALYSIS

4.1 Experimental Data

To validate this method, two different types of datasets were tested, one is the MSRA dataset of SIGHAN Chinese Named Entity Recognition Assessment in 2006 and the other is the Thangka dataset labeled by the Key Laboratory of China's Ethnic Languages and Information Technology. Entities in MSRA datasets are names, place names, and organization names, with label suffixes PER, LOC, and ORG, respectively. The Thangka dataset mainly comes from the Grand Dictionary of Buddhism and the descriptive text of Thangka images crawled on the Internet, all of which are the names of religious gods with the label suffix NSL. Both corpuses are tagged using the BIO labeling strategy. The statistical results of the experimental dataset is shown in TABLE I.

TABLE I. Statistics for experimental datasets

Corpus	Category	Training set	Validation set	Test set
MSRA	Sentences	32421	8105	4631
	PER	11274	2818	1973
	LOC	23372	5842	2877
	ORG	13166	3291	1331
Thangka	Sentences	1591	397	692
	NSL	3154	788	1061

4.2 Experiment Setting

The experimental model in this paper is built with Keras 2.3 under the Tensorflow framework. The experimental parameters are as follows: the input sentence length is 100; the training word embedding has a dimension of 100; the position vector has a dimension of 100;

the number of BiLSTM hidden layer neurons is 128; the number of Transformer coding layers is 6; the number of heads is 2; and the dimension of self-attention is 128. In addition to some variables in the Transformer coding layer, variables in the other layers are initialized by glorot_uniform. The MSRA dataset epoch is set to 10 and the Thangka dataset epoch to 100. The batch_size of both train and test sets are set to 32. To prevent over-fitting, dropout is introduced and set to 0.2. Adam method is used to update the parameters of the optimization function.

4.3 Evaluation Criterion

The experimental results are uniformly evaluated by Precision (P), Recall (R) and F1 value, that are defined as follows:

$$P = \frac{Number \quad of \quad correctly \quad identified \quad entities}{Number \quad of \quad identified \quad entities} \times 100\% \tag{16}$$

$$R = \frac{Number \quad of \quad correctly \quad identified \quad entities}{Number \quad of \quad test \quad data \quad entities} \times 100\% \tag{17}$$

$$F1 = \frac{2 \times P \times R}{P + R} \times 100\% , \tag{18}$$

4.4 Experimental Results and Analysis

For ease of description, Ours is used here to represent the BiLSTM-Transformer-CRF model presented in this paper. Ours was used to train and test on MSRA and Thangka datasets in accordance with section 4.2. At the same time, CRF and BiLSTM-CRF were constructed, together with seven named entity recognition models including BiLSTM-Attention-CRF, Transformer-CRF, Transformer*-CRF without position vector and Ours* without position vector for contrast experiment. The recognition effects of different models are shown in Table II.

TABLE II. Comparison of recognition effects of different models on datasets(in %)

Coups	Model	P	R	F1
MSRA	CRF	52.9	31.9	39.8
	BiLSTM-CRF	81.2	76.3	78.7
	BiLSTM-Attention-CRF	83.9	74.6	79.0
	Transformer-CRF	66.7	60.3	63.3
	Transformer*-CRF	67.3	61.3	64.2
	Ours*	81.8	79.0	80.4
	Ours	87.1	76.4	81.4
Thangka	CRF	80.3	68.9	74.1
	BiLSTM-CRF	87.5	80.0	83.6
	BiLSTM-Attention-CRF	87.6	80.7	84.1

	Transformer-CRF	80.3	63.9	71.2
	Transformer*-CRF	90.1	81.4	85.5
	Ours*	90.1	80.6	85.1
	Ours	93.8	81.1	86.9

The result data show that the recognition effect of deep neural network model is much higher than that of traditional machine learning model CRF, in which Ours model performs best on both datasets, with F1 values of 81.4% and 86.9% on MSRA and Thangka datasets respectively, which are 2.7% and 3.3% higher than that of BiLSTM-CRF model respectively, indicating that adding Transformer encoder to extract the feature of character-to- character relationship based on BiLSTM can effectively improve the accuracy of Chinese named entity recognition.

Comparison between the BiLSTM-CRF, BiLSTM-Attention-CRF and Transformer-CRF models reveals that the attention mechanism after BiLSTM improves the recognition slightly, but the F1 of the Transformer-CRF model is 10% lower than that of BiLSTM-CRF. In addition, the Transformer*-CRF without position vectors performs better on both datasets than the original Transformer model, which illustrates the inadequacy of Transformer's ability to use position coding to represent position information, and the wrong way in which the position vectors are simply added to the word embedding, not only destroying the pre-training word embedding, but ultimately failing to retain the relative position information and forward and backward directions between words, which also verified the conclusion in reference [20] that the relative distance information contained in the position coding matrix is lost in subsequent attention calculations.

Further comparison of the experimental results reflects that the comprehensive F1 value of Ours model is 18.1% and 15.7% higher than Transformer-CRF model on MSRA and Thangka datasets respectively, and about 1.0% and 1.8% higher than Ours*, indicating that the processing method of splicing position vector with pre-training word vector and then feeding it into BiLSTM network can alleviate the problems of position information disappearance and direction information missing between words to a certain extent, while retaining the original word embedding data, which helps improve the effect of Transformer on Chinese named entity recognition.

The identification of the Thangka test set by BiLSTM-Attention-CRF model (Model 1) and Ours model (Model 2) is also counted as shown in TABLE III.

TABLE III. Recognition results of two models in the Thangka test set

	Length of named entity					**Total**
	1	**2**	**3~5**	**6~10**	**>10**	
Number of entities in Test set	0	55	930	69	2	1056

Recognized by Model1	54	46	821	52	0	973
Correctly recognized by Model1	0	28	774	51	0	863
Accuracy of Model1	0	60.8%	94.2%	98.1%	-	87.7%
Recognized by Model2	11	51	797	52	1	912
Correctly recognized by Model2	0	33	773	50	0	856
Accuracy of Model2	0	64.7%	96.9%	96.1%	0	93.9%

Statistical results show that there is no single-word entity in the Thangka test set, and the length of entity name is mostly concentrated in 3-5. The trained Model1 and Model2 have approximately the same total number of entity names identified on the test set, with similar recognition accuracy of 3 - 10 words or more. Overall, Model2 is more accurate because only 11 incorrectly named word entities have been identified, much less than 54 in Model1.

The statistical results on the Thangka test set reflect that the Ours model can capture the contextual feature and the character-to-character relational features, and learn sentence structure jointly, so that there are fewer errors in identifying single-word entities, while word recognition errors are often proportional to entity boundary errors, so the effect of Ours model is better than other models as a whole.

V. CONCLUSIONS

In this paper, a named entity recognition method based on Transformer model is proposed, which makes up for the problems of position information loss and direction information missing of the original Transformer model, and makes full use of BiLSTM's directional characteristics and context feature extraction capability as well as the powerful character-to-character feature capture capability of the Transformer coding layer. Experiments on MSRA and Thangka datasets show that the method is effective and applicable for Chinese named entity recognition in general and special fields without reliance on external resources and pre-training language models. In future research, it is considered improving the internal attention mechanism algorithm of Transformer to make the model structure more suitable for named entity recognition tasks and improve training efficiency and recognition effect.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant No.61162021; Program for Innovative Research Team of SEAC under Grant No.2018(98); Special Programs of the State Ethnic Affairs Commission of Central Universities under Grant No. 1001160448.

REFERENCES

[1] Chinchor N. (1995) MUC-6 named entity task definition (version2.1)// Proceedings of the 6th Message Understanding Conference, Columbia, Maryland, November, 1995.Stroudsburg: ACL: 317-332.

[2] Liu Liu, Wang Dongbo. (2018) Summary of named entity recognition. Journal of the China Society for Scientific and Technical Information 37 (003): 329-340.

[3] Ma X, Hovy E. (2016) End-to-end sequence labeling via Bidirectional LSTM-CNNs-CRF. arXiv.1603.01354.

[4] Han Xinxin, Ben Kerong, Zhang Xian. (2020) Named entity recognition technology in military software testing field. Journal of Frontiers of Computer Science and Technology 14(5): 740-748.

[5] Vaswani A, Shazeer N, Parmar N, et al. (2017) Attention is all you need. Proceedings of Advances in Neural Information Processing Systems: 5998-6008.

[6] Devlin J, Chang M W, Lee K, et al. (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[7] Hang Yan, Bocao Deng, Xiaonan Li, et al. (2019) TENER: Adapting Transformer Encoder for Named Entity Recognition. arXiv preprint arXiv:1911.04474, 2019.

[8] Habibi Maryam, Mariana Neves, David Luis Wiegandt. (2017) Deep learning with word embeddings improves biomedical named entity recognition. Bioinformatics 33(14):37-48

[9] Ji Y, Tong C, Liang J, et al. (2019) A deep learning method for named entity recognition in bidding document. Journal of Physics Conference Series 1168(3):032076.

[10] Hochreiter S, Schmidhuber, Jurgen. (1997) Long Short-Term Memory. Neural Computation 9(8): 1735-1780.

[11] Graves A, Jurgen Schmidhuber. (2005) Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks 18(5):602-610.

[12] Huang Z,Xu W,Yu K. (2015) Bidirectional LSTM-CRF models for sequence tagging. Computer Science.

[13] Rei M, Crichton G K O, Pyysalo S. (2016) Attending to Characters in Neural Sequence Labeling Models. arXiv preprint arXiv:1611.04361.

[14] Li Mingyang, Kong Fang. (2019) Social media named entity recognition incorporating self-attention mechanism. Journal of Tsinghua University (Natural Science Edition) 6: 461-467.