Chinese Named Entity Recognition Based on BERT with Whole Word Masking

Chao Liu
College of Computer Science and Technology
Beijing University of Technology
Beijing, China
+86-15811580072
liuchaozb@foxmail.com

Cui Zhu
College of Computer Science and Technology
Beijing University of Technology
Beijing, China
+86-13910776809
cuizhu@bjut.edu.cn

Wenjun Zhu
College of Computer Science and Technology
Beijing University of Technology
Beijing, China
+86-18614086048
zhuwenjun@bjut.edu.cn

ABSTRACT
Named Entity Recognition (NER) is a basic task of natural language processing and an indispensable part of machine translation, knowledge mapping and other fields. In this paper, a fusion model of Chinese named entity recognition using BERT, Bidirectional LSTM (BiLSTM) and Conditional Random Field (CRF) is proposed. In this model, Chinese BERT generates word vectors as a word embedding model. Word vectors through BiLSTM can learn the word label distribution. Finally, the model uses Conditional Random Fields to make syntactic restrictions at the sentence level to get annotation sequences. In addition, we can use Whole Word Masking (wwm) instead of the original random mask in BERT's pre-training, which can effectively solve the problem that the word in Chinese NER is partly masked, so as to improve the performance of NER model. In this paper, BERT-wwm (BERT-wwm is the BERT that uses Whole-Word-Masking in pre training tasks), BERT, ELMo and Word2Vec are respectively used for comparative experiments to reflect the effect of bert-wwm in this fusion model. The results show that using Chinese BERT-wwm as the language representation model of NER model has better recognition ability.

CCS Concepts
• **Computing methodologies→Artificial Intelligence→Natural language processing→Information extraction.**

Keywords
Named entity recognition; BERT; whole word masking; Bi-LSTM; conditional random field.

1. INTRODUCTION
NER is a fundamental key task in NLP. From the perspective of natural language processing flow, NER can be seen as one of the unregistered word recognition in the analysis of word formation, which is the most difficult problem in recognition and has the greatest impact on the effect of word segmentation. At the same time, NER is also the basis of many NLP tasks, such as relationship extraction, event extraction, knowledge map, machine translation, question answering system, etc. Named entities in English have obvious formal signs, that is, the first letter of each word in the entity should be capitalized. Compared with English, Chinese named entity recognition task is more complex, and entity boundary recognition is more difficult than entity class annotation subtask.

In recent years, with the development of hardware capabilities and the emergence of word embedding, neural networks have become models that can efficiently handle many NLP tasks. This kind of method is similar to sequential annotation tasks (such as CWS, POS, NER). The token is mapped from the discrete one hot representation to the low-dimensional space to become a dense embedding. Subsequently, the embedding sequence of sentences was input into RNN, features were extracted automatically by neural network. Finally, softmax is used to predict the label of each token. This method makes model training an end-to-end process. This is a data-driven, feature independent approach rather than a traditional pipeline. One disadvantage of this method is that the process of tagging each token is an independent classification, so the predicted tag cannot be directly used (the information above can only be passed by implicit state), thus leading to the predicted tag sequence may be illegal. For this case, we can use the CRF layer after the RNN layer to make sentence level prediction, so that the annotation process is no longer the independent classification of each token, thus enhancing the performance of the recognition model. This paper uses a combined model including bidirectional LSTM and conditional random field for NER recognition.

Most previous methods used static word embedding, such as Word2Vec [1] and GloVe [2]. This static embedding is convenient, but it makes some unrealistic assumptions about language, especially when words have different meanings in different contexts. This kind of word embedding cannot distinguish polysemous words, it will assign the same word vector to the polysemous words in different contexts. In order to solve this problem, we propose to explore the use of contextualized word embedding, such as BERT (Bidirectional Encoder Representations from Transformers) [3]. BERT introduced the attention mechanism and stacked Transformer, which has stronger language representation ability. In addition, we will introduce the Whole- Word-Masking method [4] to replace the random masking method to ensure the integrity of words in the pre training task, so as to further optimize the model.

ICCAI '20, April 23–26, 2020, Tianjin, China.

ACM ISBN 978-1-4503-7708-9/20/04…$15.00
DOI: https://doi.org/10.1145/3404555.3404563

2. RELATED WORK

At present, in the field of named entity recognition, the main research methods at home and abroad can be divided into three categories: rules and dictionary methods, statistical machine learning methods and neural network methods. Rules and dictionary methods usually use linguistic experts to construct rule templates by hand, with matching of patterns and strings as the main means. Most of these systems rely on the establishment of knowledge base and dictionaries. At present, the commonly used methods of named entity recognition based on statistical machine learning include Hidden Markov Model (HMM)[5], Maximum Entropy Model (me)[6], Conditional Random Field (CRF)[7], etc. The main idea is: based on a large number of corpus of manual tagging, named entity recognition is regarded as a sequence tagging problem, and the corpus is used to learn the tagging model, so as to tag each position of the sentence. In recent years, the main method based on neural network is to use Bidirectional LSTM [8] to extract text features and obtain the annotation probability of each token. In reference [9] [10], after the LSTM layer, the CRF layer is connected to do sentence level label prediction, so that the labeling process is no longer an independent classification of each token. The introduction of the idea of CRF can actually be traced back to literature [11]. In reference [10], it is also proposed to use LSTM to construct words from letters for each word in the English NER task, and then input them into LSTM after splicing them into word vectors, so as to capture the morphological characteristics of letters, such as prefixes and suffixes of words. Reference[12] uses this method in Chinese NER task, and constructs Chinese characters with partial radicals.

In our proposed method, BERT is introduced to provide word embedding for Bidirectional LSTM, and word vectors with better features are obtained through attention mechanism and Transformer. We also improve the feature extraction ability of NER model by changing the word mask mode in BERT pre training method.

3. METHODOLOGY

In this section, we will introduce and analyze our proposed method in details.

Figure 1 shows the overall structure of our proposed model. Each part of the model will be explained below.

This model can be roughly divided into three modules: presentation layer, Bidirectional LSTM (BiLSTM) layer and CRF layer.

(1) Presentation layer. In this layer, The BERT is used to embed the data set to generate word vectors. Word embedding is simply to map the high-dimensional one hot vector in the high-dimensional space (the dimension of the space is usually the size of the dictionary) to the vector in the low-dimensional continuous space. The mapped vectors are called the word vectors.

(2) BiLSTM layer. The generated word vector is used as the input of BiLSTM. BiLSTM can extract the two-way features of each input sentence, and combine the forward and reverse LSTM to get a complete hidden state sequence. Finally, a linear layer is used to map the hidden state vector to the k-dimension vector (k is the number of label sets), and the scores of all labels of each word are obtained. In this way, features are extracted.

(3) CRF layer. Conditional Random Fields can add some constraints to the final prediction to ensure they are valid. By scoring the possible annotation sequences, the best one is obtained. At this point, we have finished the named entity recognition.

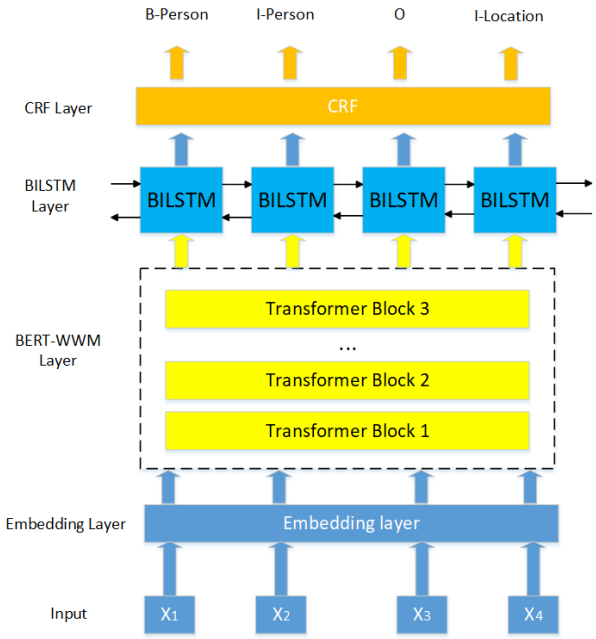

Figure 1. Model structure.

3.1 Chinese BERT-wwm (Whole-Word-Wasking)

The full name of BERT model is "Bidirectional Encoder Representations from Transformers". It is a new pre trained language model, whose performance surpasses many systems using task specific architecture. BERT-wwm is an improved version of BERT using Whole Word Masking in pre training tasks. We'll start with BERT. The following Figure 2 and Figure 3 are the structure diagrams of BERT and Transform Block respectively.

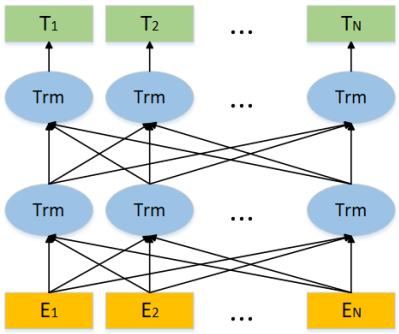

Figure 2. BERT is composed of TRM stacks and has the ability to extract bi-directional features of text.

BERT's network architecture uses a multi-layer Transformer structure[13], which transforms the distance between two words at any position into "1" through attention mechanism, effectively solving the thorny long-term dependency problem in NLP. The BERT structure is shown in Figure 2. The left part of Transformer [14] is a Transformer Block, corresponding to a "Trm" in Figure 2.

Transformer has abandoned the traditional CNN [15] and RNN, and the whole network structure is completely composed of attention mechanism. More precisely, Transformer only consists of self-attention and feedforward neural network. It uses the attention mechanism to reduce the distance between any two positions in the sequence to a constant, which has stronger feature extraction ability. In essence, Transformer is an encoder decoder structure, which is formed by stacking several encoders and decoders. The left part is the encoder, which is composed of multi head attention and a full connection. It is used to convert the input corpus into the feature vector. The right part is the decoder, whose input is the output of the encoder and the predicted result. It is composed of masked multi head attention, multi head attention and a full connection, which is used to output the conditional probability of the final result.

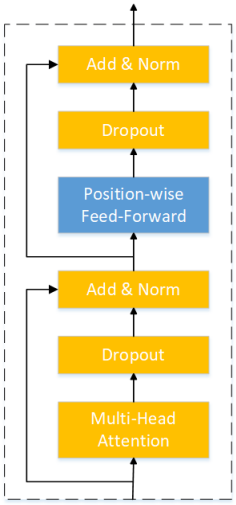

Figure 3. Transformer block.

BERT is a multitask model. Its tasks are composed of two self-monitoring tasks, namely, Masked Language Model (MLM) and Next Sense rediction (NSP). The core idea of MLM is taken from a paper published by Wilson Taylor in 1953.

Table 1. Original text and Participle text use the word segmentation method of WordPiece and LTP respectively. The [M] means to mask the original word.

Original text	他喜欢学数学和语文 (He likes learning mathematics and Chinese)
Participle text	他 喜欢 学 数学 和 语文
Original Mask	他 喜[M] 学 数[M] 和 和语文
Whole Word Mask	他 [M][M] 学 [M][M] 和 语文

Whole Word Masking (WWM) mainly changes the strategy of training sample generation in the original pre training stage. In short, the original word segmentation method based on WordPiece [17] will cut a complete word into several sub words. When generating training samples, these separated sub words will be randomly masked. In the method of Whole-Word-Masking, if a part of a complete word is masked, the other parts of the same word will also be masked. Similarly, in Google's official BERT, Chinese is segmented based on the granularity of words, without considering the Chinese segmentation in traditional NLP, so we apply the method of Whole-Word-masking in the preprocessing task.

Whole-Word-Masking uses the LTP (language technology platform) of Harbin University of technology as a word segmentation tool to segment text according to Chinese grammar, and then uses Chinese Wikipedia (including simplified and traditional) for training. In this way, all the Chinese characters that make up the same word are masked. As shown in the Table 1.

3.2 Bi-LSTM layer

In this layer, BiLSTM takes the word vectors output from BERT as the input, and passes the embedding sequence of each word in each sentence through a Bidirectional LSTM. The hidden state sequence of forward LSTM output and the hidden state sequence of reverse LSTM output in each position will be spliced to get a complete sequence. Finally, after the sequence passes through a linear layer, the hidden state vector is mapped from the original dimension to the K dimension. K is the number of label sets. In this way, BiLSTM automatically extracts sentence features.

LSTM is a variant of recurrent neural network (RNN) [16]. Each neuron in RNN can not only transfer the output to the next neuron, but also introduce the concept of timing. It can take the current output as the input of its next time, and make the information cycle in the network. However, if the information obtained before is too far away from the current interval, RNN's ability to concatenate relevant information will be weak. LSTM can solve the problem of long-term dependence and make up for the defect of RNN.

However, there is still a problem in using LSTM to model sentences: it is impossible to encode the information from the back to the front. BiLSTM is a combination of forward and reverse LSTM, so BiLSTM can better capture bi-directional semantic dependency [17].

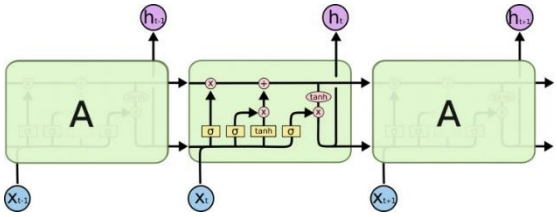

Figure 4. LSTM.

As shown in the following figure, the sentence S consists of (X_0, X_1, X_2, X_3, X_4), and X is the embedding of each word. After entering the bidirectional LSTM, the hidden state sequence $(\overrightarrow{h_1}, \overrightarrow{h_2}, \dots, \overrightarrow{h_n})$ is obtained through the forward LSTM, and the hidden state sequence $(\overleftarrow{h_1}, \overleftarrow{h_2}, \dots, \overleftarrow{h_n})$ is obtained through the reverse LSTM, and then the two hidden state sequences are spliced at corresponding positions to obtain the complete hidden state sequence. The splicing process and results can be expressed as 3.2-1 and 3.2-2, respectively, and H represents the final splicing sequence.

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}] \in \mathrm{Rn}^*\mathrm{m} \quad (1)$$

$$H = (\overrightarrow{h_1}, \overrightarrow{h_2}, \dots, \overrightarrow{h_n}) \in \mathrm{Rn}^*\mathrm{m} \quad (2)$$

After dropout is set, a linear layer is connected, and the hidden state vector H is mapped from m dimension to K dimension, so

that the sentence feature is automatically extracted. K is the number of labels in the annotation set. If you continue to access a softmax classifier, you can independently classify each location into k classes. The output of softmax layer is independent of each other. Although BiLSTM learns the context information, the output has no influence on each other. It just selects a label output with the maximum probability value at each step. This will lead to problems such as B-person followed by another B-person. There is a transfer feature in CRF, that is, it will consider the order between the output labels. So consider using CRF to build the output layer of BiLSTM.

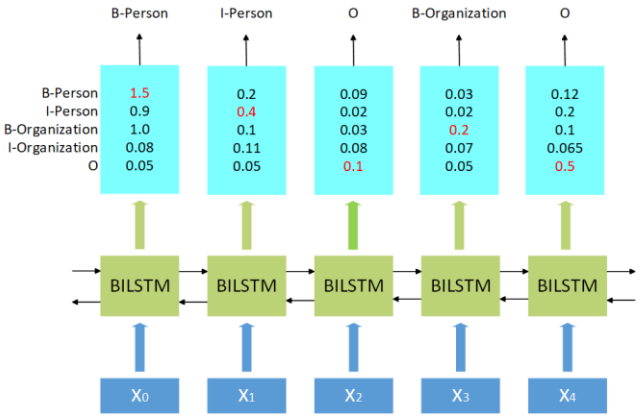

Figure 5. The word vector (X_1,X_2,…) generated from the BERT enters BILSTM to get the label score for each word.

As shown in Figure 5, after entering the word vectors into the BiLSTM, the scores of each label of each word vector are obtained. These scores are called Emission scores and will be used as parameters of CRF to calculate the scores of the last labeled sequence.

3.3 CRF Layer

We connect the Conditional Random Field behind BiLSTM. The CRF layer can learn some constraints from the training data set to ensure that the final predicted entity tag sequence is effective.

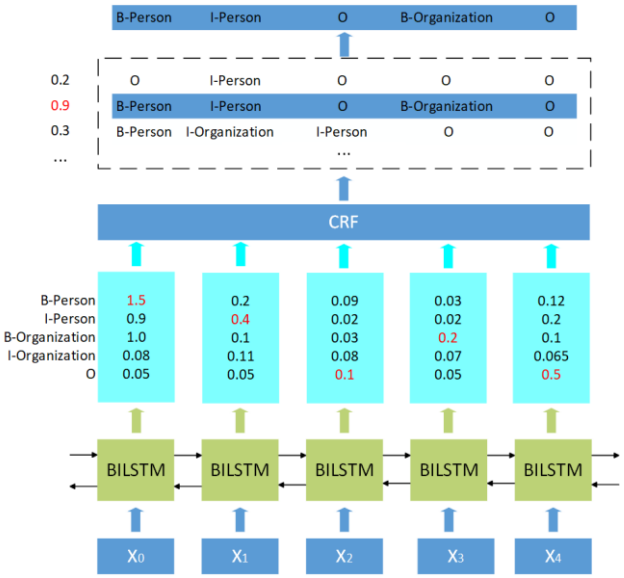

Figure 6. After the BiLSTM layer, the CRF layer was connected to add sentence level prediction to the model.

In the loss function of CRF layer, we have two kinds of fractions. These two scores are the key concepts of CRF layer. The first is Emission score. The emission score is the score of all labels of each word in the text output by BiLSTM, as detailed above. The second is Transition score. Transition score is the probability score of connecting another label (such as B) after A label. The higher the score, the more likely B label is to be behind a label. The lower the score, the less likely it is. This Transition score matrix reflects the grammatical meaning of the sentence. CRF constrains sequence annotation by this meaning.

Table 2. This table represents the Transition score matrix. "I-PER" is generally after "B-PER", so their conversion score is 0.9, while "I-ORG" is only 0.0006 after "B-PER". The meaning of labels can be found in Chapter 4.1

	B-PER	I-PER	B-ORG	I-ORG
B-PER	0.6	0.9	0.2	0.0006
I-PER	0.5	0.53	0.55	0.0003
B-ORG	0.5	0.0003	0.25	0.8
I-ORG	0.45	0.007	0.7	0.65

In fact, the matrix is a parameter of the BiLSTM-CRF model [18]. Before you train the model, you can randomly initialize all transition scores in the matrix. All random scores will be updated automatically during training. In other words, the CRF layer can learn these constraints on its own. We don't need to build the matrix manually. With the increase of training iterations, the score will become more and more reasonable.

With the Emission fraction and Transition fraction, we can use them to calculate the loss function of CRF. The CRF loss function consists of the actual path score and the total score of all possible paths, and the real path should have the highest score among all possible paths.

$$\text{Si} = \text{EmissionScore}+\text{TransitionScore} \tag{3}$$

$$P_{total} = P_1 + P_2 + \cdots + P_n = e^{S_1} + e^{S_1} + \cdots + e^{S_1} \tag{4}$$

$$\text{LossFunction} = \frac{\text{P}_{\text{RealPath}}}{\text{P}_1+\text{P}_2+\text{P}_3+\cdots+\text{P}_n} \tag{5}$$

In this way, through the CRF loss function, the best annotation sequence will be obtained, and the label of annotation is the named entity we want to obtain.

4. EXPERIMENT

In this section, we describe experimental settings and report empirical results.

4.1 Datasets

In order to evaluate the effectiveness of our model method, we use the Chinese NER data set published by Microsoft Asia Research Institute. We use the BIO annotation method in the data set. B-PRE and I-PRE represent the initial word and non-initial-word of person entity respectively. Similarly, B-LOC and I-LOC represent the initial words and non-initial-words of place entity. B-ORG and I-ORG represent the inital word and non-initial-word of organization name. O means that the word is not part of the named entity. Such as:

里 皮 表 示 对 中 国 队 很 失 望
B-PER I-PER O O O B-ORG I-ORG I-ORG O O O

Figure 7. BIO annotation. The English translation of the example is: Lippi is disappointed with the Chinese team.

4.2 Experiment Methods

We use four NER methods to compare the performance of each model, so as to analyze the performance parameters of the model proposed in this paper. We use four similar methods, including word embedding model, bidirectional LSTM and Conditional Random Field. But the difference is the model that produces the word vector. In this paper, Word2Vec, ELMo [19], BERT and BERT- wwm are used as word embedding models. Theoretically, their performance should be increased in turn.

4.3 Experimental Setting

We use the BERT model with a hidden size of 768, 12 self-attention heads and 12 Transformer 4 blocks. BERT-wwm is an improved version of BERT using whole-word-masking in pre training tasks. The BERT module is initialized by pre-trained BERT weights, while all remaining layers are learned from scratch. For experimental Data, we use ten percent of the training set as the development set and tune our model on the development set to determine the optimal parameters. We use Adam optimizer (with $\beta1 = 0.9$ and $\beta2 = 0.999$) to minimize the cross-entropy loss. The slanted triangular learning rate is used in our experiment, the base learning rate is 2e-5, and the warm-up proportion is 0.1. In the experiment we find that a lower learning rate helps to effectively use pre-trained BERT weights. Dropout probability is always kept at 0.1. The batch size is set to 32. During the training process, it was found that the BERT learning rate had a good effect at 3e-5, while the BERT-wwm was at 4e-5. All experiments are performed on a single NVIDIA GTX 1080 Ti GPU.

4.4 Overall Results

In this section, we will report the performance of four named entity recognition methods on datasets. The purpose of this paper is to demonstrate the performance of BERT-wwm (wholel word masking in BERT's pre training) combined with Bidirectional LSTM, Conditional Random Field model. The experimental data is shown in the following table. It can be seen that the accuracy and recall rate of the model using BERT as the representation layer are better than Word2Vec and ELMo. This shows that the fusion model of BERT and BiLSTM-CRF has a good effect on Chinese NER. Moreover, we can also find that BERT-wwm has better recognition effect than that of Bert. This shows that the Whole-Word-Masking used by BERT-wwm in the pre training task can reduce the negative impact of random masking on Chinese word segmentation.

Table 3. The four models are trained separately. P, R, and F1 are the Precision, recall, and FI values

	P	R	F1
Word2Vec - BiLSTM - CRF	89.53	88.86	89.19
ELMo – BiLSTM - CRF	92.71	92.39	92.56
BERT – BiLSTM - CRF	94.18	93.75	93.97
BERT(wwm)–BiLSTM - CRF	94.42	94.13	94.28

Here is a brief summary of the results of the experiment.

(1) BiLSTM-CRF is still an excellent and robust named entity recognition model. No matter which embedding method is adopted, it has a good recognition effect.

(2) When combining BiLSTM and CRF for Chinese named entity recognition, the fusion model uses BERT as the word embedding model, which has better recognition effect than Word2Vec and ELMo.

(3) In the pre training task of BERT, the Whole-Word-Masking method can effectively solve the problem of ignoring Chinese word segmentation by random masking, which makes BERT-wwm have better performance in Chinese NER.

5. CONCLUSIONS

In this paper, we propose a fusion model for Chinese named entity recognition using BERT-wwm, Bidirectional LSTM and Conditional Random Field. In order to solve the problem that Chinese words are partially masked due to random masking in BERT pre training, we adopt a new Chinese word segmentation method and use the whole-word-masking instead of the original random-masking. By using this method, the pre training efficiency of Bert and the recognition performance of the whole model are improved. Moreover, we also prove that the fusion of BERT and BiLSTM-CRF can show good vitality in Chinese NER, and the effect is significantly better than that of Word2Vec and ELMo.

In the future, we will study whether we can realize the simplification of BERT by adjusting and optimizing the structure of the transformers in BERT. In addition, we will try to visualize the attention mechanism in BERT, and continue to explore other pre-trained language representation models in the direction of NER, such as OpenAI GPT, ERNIE [20], etc.

6. REFERENCES

[1] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. arXiv preprint arXiv:1301.3781, 2013.

[2] Pennington J, Socher R, Manning C. Glove: Global vectors for word representation[C]//Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 1532-1543.

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

[4] Cui Y, Che W, Liu T, et al. Pre-Training with Whole Word Masking for Chinese BERT[J]. arXiv preprint arXiv:1906.08101, 2019.

[5] Beal M J, Ghahramani Z, Rasmussen C E. The infinite hidden Markov model[C]//Advances in neural information processing systems. 2002: 577-584.

[6] Ratnaparkhi A. A maximum entropy model for part-of-speech tagging[C]//Conference on Empirical Methods in Natural Language Processing. 1996.

[7] Tseng H, Chang P, Andrew G, et al. A conditional random field word segmenter for sighan bakeoff 2005[C]//Proceedings of the fourth SIGHAN workshop on Chinese language Processing. 2005.

[8] Sundermeyer M, Schlüter R, Ney H. LSTM neural networks for language modeling[C]//Thirteenth annual conference of the international speech communication association. 2012.

[9] Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging[J]. arXiv preprint arXiv:1508.01991, 2015.

[10] Lample G, Ballesteros M, Subramanian S, et al. Neural Architecturesfor Named Entity Recognition[C]//Proceedings of NAACL-HLT. 2016: 260-270.

[11] Collobert R, Weston J, Bottou L, et al. Natural language processing (almost) from scratch[J]. Journal of Machine Learning Research, 2011, 12(Aug): 2493-2537.

[12] Dong C, Zhang J, Zong C, et al. Character-Based LSTM-CRF with Radical-Level Features for Chinese Named Entity Recognition[C]//International Conference on Computer Processing of Oriental Languages. Springer International Publishing, 2016: 239-250.

[13] Luo L, Yang Z, Yang P, et al. An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition[J]. Bioinformatics, 2017, 34(8): 1381-1388.

[14] Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "Spatial transformer networks." Advances in neural information processing systems. 2015.

[15] Iandola F N, Han S, Moskewicz M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size[J]. arXiv preprint arXiv:1602.07360, 2016.

[16] Chen Lyu,Bo Chen,Yafeng Ren,Donghong Ji. Long short-term memory RNN for biomedical named entity recognition[J]. BMC Bioinformatics,2017,18(1).

[17] Chen T, Xu R, He Y, et al. Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN[J]. Expert Systems with Applications, 2017, 72: 221-230.

[18] Luo L, Yang Z, Yang P, et al. An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition[J]. Bioinformatics, 2017, 34(8): 1381-1388.

[19] Peters M E, Neumann M, Iyyer M, et al. Deep contextualized word representations[J]. arXiv preprint arXiv:1802.05365, 2018.

[20] Sun Y, Wang S, Li Y, et al. ERNIE: Enhanced Representation through Knowledge Integration[J]. arXiv preprint arXiv:1904.09223, 2019.