Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **6**, Number **3**, August **2021**

A Multi-Feature Fusion method Based on BiLSTM-Attention-CRF for Chinese Named Entity Recognition

Zhiyuan Zhang

School of Computer Science and Mathematics
Fujian University of Technology
No.33 Xuefu South Road, University New District, Fuzhou 350118, China
1107416742@qq.com

Shuihua Sun*

School of Computer Science and Mathematics
Fujian University of Technology
No.33 Xuefu South Road, University New District, Fuzhou 350118, China
*Correponding Author: shuihua.11109029@gmail.com

Shiao Xu

School of Computer Science and Mathematics
Fujian University of Technology
No.33 Xuefu South Road, University New District, Fuzhou 350118, China
946831760@qq.com

Fan Xu

School of Computer Science and Mathematics
Fujian University of Technology
No.33 Xuefu South Road, University New District, Fuzhou 350118, China
455345562@qq.com

Jianhua Liu

School of Computer Science and Mathematics
Fujian University of Technology
No.33 Xuefu South Road, University New District, Fuzhou 350118, China
jhliu@fjnu.edu.cn

Received May 2021; Revised June 2021

Abstract. *Named entity recognition (NER) aims to identify entities with specific meanings in texts, which is an important basic work in application fields such as information extraction, question answering systems, and syntax analysis. The existing character-based Chinese NER system has problems that the glyph structure information of Chinese characters is easily ignored and the word information is insufficient. This paper proposes a Chinese NER model based on BiLSTM-Attention-CRF with multi-feature fusion. The model introduces a new type of Five-stroke Recurrent Convolutional Neural Network (FS-RCNN) structure to capture the stroke-level glyph feature of Chinese characters from the five-stroke input method and uses the embedding of left and right characters to extract the potential affix feature in words. The character vector, glyph feature vector, and affix feature vector are fused, and input into the BiLSTM-Attention-CRF model for Chinese NER. Experimental results on the SIGHAN Bakeoff MSRA data set and the People's Daily data set show that the model effectively improves the performance of NER, with F-score reaching 91.44% and 92.68% respectively.*
Keywords: Natural language processing; Chinese NER; BiLSTM-Attention-CRF model; Stroke-level glyph feature; Affix feature

1. **Introduction.** Named entity recognition (NER) is an important subtopic in the research field of natural language processing (NLP). It aims to identify and classify the name of persons (PERs), locations (LOCs), and organizations (ORGs) in text and provides important technical support for the research of downstream tasks such as relation extraction [1], event extraction [2], question answering system [3].

In the early stage, NER methods mainly include the rule-based method and based on statistical machine learning method. The rule-based method is to identify named entities through rules manually formulated by domain experts and linguists. This method is easy to implement and has a better effect in NER, but making rules is time-consuming and laborious, and the domain migration is poor. Based on statistical machine learning method usually rely on manually extracting features to ensure the performance of the system. This method can learn the correlation and importance of features, but the design of features requires repeated tests, adjustments, and selections. In recent years, with the successful application of neural networks in the fields of image processing [4–6] and speech recognition [7,8], the deep learning method has been increasingly used in NLP tasks. Deep learning can automatically learn the features of sentences without manual intervention. Therefore, the NER method based on deep learning is widely concerned. The BiLSTM-CRF model [9–11] is one of the best combinations of current NER. Its BiLSTM layer can effectively learn the context information of the previous and subsequent texts, and the CRF layer can learn the constraints between tags so that the tagging process is no longer a separate classification of each word or character.

The BiLSTM-CRF model is used for Chinese NER, including the word-based method and the character-based method. The word-based method [12] needs to use word segmentation tools to segment the text, and the wrong segmentation of the word segmentation tools will cause the accuracy of NER to decrease. Therefore, most of the existing Chinese NER systems are developed using the character-based method [13], but this method still has limitations: (1) As Chinese characters are pictographs, it is easy to ignore the semantic information contained in their glyphs. (2) Due to the limitation of window size, word information cannot be obtained. Yang et al. [14] used Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) respectively to extract the stroke-level features of Chinese characters, which are connected with the context-independent character vectors and input into the CNN-BiLSTM-CRF model. This method can help to extract the deeper features of characters and obtain multiple N-gram features by using

multiple convolution windows, but RNN or CNN alone cannot effectively extract the semantic information of strokes. Zhang et al. [15] concatenated the character vector and the double-character vector and input into the BiLSTM-CRF model for NER. This method effectively captures the potential word information but loses the boundary information of the original word sequence. Based on the above method, this paper proposes a Chinese NER model based on BiLSTM-Attention-CRF with multi-feature fusion. Firstly, we introduce a new type of Five-stroke Recurrent Convolutional Neural Network (FSRCNN) structure, Recurrent Convolutional Neural Network (RCNN) can not only combine the advantages of RNN to process sequence data and CNN to extract local features, but also solve the problems that RNN is easy to ignore the important information of the preceding part of the sentence and the CNN convolution window is not easy to determine. This structure effectively captures the stroke-level glyph feature of Chinese characters from the five-stroke input method by RCNN, so as to obtain the internal semantic information of Chinese characters. Secondly, we use the left and right character embeddings to extract the potential affix feature in words. Finally, a variety of feature vectors are concatenated with the original character vectors and then input into the BiLSTM-Attention-CRF model for training to construct a Chinese NER model.

The main contributions of this paper are as follows:

- This paper introduces a new type of FSRCNN structure, which applies RCNN to the five-stroke input method to capture the stroke-level glyph feature of Chinese characters, and uses left and right character embeddings to extract potential affix feature in words. The network model can effectively extract the semantic information inside Chinese characters and solve the problem of the insufficient word information in the character-based model.
- The original character vector, glyph feature vector, and affix feature vector are fused and input into the BiLSTM-Attention-CRF model, which is helpful for the identification of entity boundaries and the elimination of entity ambiguity. At the same time, the attention mechanism can enhance the use of key characters. Compared with the BiLSTM-CRF model, this model has a certain increase in F-score.
- The effectiveness of the model is verified on the SIGHAN Bakeoff MSRA data set [16] and the People's Daily data set.

2. **Related Work.** NER was first proposed at the MUC-6 conference in the 1990s. Since then, this task has been a hot topic in the field of NLP. In the early days, NER mainly uses the rule-based method, which constructs rules manually by analyzing the constitutive features and context features of entities, and utilizes rules to match entities from text. Grishman [17] wrote different extraction rules for different types of entities and achieved good results in NER. Collins et al. [18] defined a seed rule set, and unsupervised training is carried out on the set according to the untagged corpus to obtain more rules, and the expanded rule set is used for NER. In the late 1990s, the research on NER gradually turned to machine learning methods. Based on statistical machine learning methods include Support Vector Machine(SVM), Maximum Entropy(ME), Conditional Random Field(CRF), etc. These methods extract features such as parts of speech, semantics, and affixes from the hand-tagged corpus and construct feature vectors, which are used as the input of the machine learning model to recognize named entities in text. Ke et al. [19] combined a large amount of untagged corpus with a small amount of tagged corpus, and jointly trained the SVM model and CRF model. Compared with a single model, the F-score increased by 10%. Azpeitia et al. [20] used the ME model for NER, which comprehensively considers the features of word suffixes, neighboring words, word prefixes,

and word lengths. McCallum et al. [21] proposed a CRF inductive feature model, which uses the feature induction method and Viterbi algorithm to find the optimal tag sequence.

In recent years, with the development of artificial intelligence technology [22], deep learning has made great breakthroughs in the field of NLP. Compared with the NER methods based on rules and statistical machine learning, the deep learning neural network method can automatically learn the hidden features in sentences without relying on feature engineering. Collobert et al. [23] used CNN to automatically extract features in the sentence and added a sentence-level log-likelihood function for NER, the experimental results showed that the performance of the model has been improved. Huang et al. [24] applied the word-level BiLSTM-CRF model to the NLP benchmark sequence tagging data set for the first time and achieved good results on the tasks of part-of-speech tagging, block, and NER. Luo et al. [25] integrated the document-level attention mechanism into the BiLSTM-CRF model and used the attention mechanism to focus on the same tags of different sentences in the document. This method can effectively obtain document-level global information, solve the problem of inconsistent tags, and achieve a better effect in NER with only a small amount of feature engineering. Liu et al. [26] compared character-based and word-based models and found that the former can simplify the system without reducing system performance. How to add more relevant key features such as Chinese character glyph information and word information into the character-based model to improve the accuracy of NER has attracted the attention of scholars. Lample et al. [27] proposed a neural network model for NER. This model uses not only a small amount of supervised data and a large amount of untagged corpus, but also adds BiLSTM based on character embedding. Rei et al. [28] adopted an attention model similar to a gating mechanism to dynamically select word-level information and character-level information as the input of the model. Compared with only word vectors or character vectors as the input of the model, the method's effect of NER on multiple data sets has significantly improved. Dong et al. [29] divided each Chinese character into multiple radicals, the BiLSTM is used to extract radical-level features, which is integrated to the character-level BiLSTM-CRF. This method has achieved good results in Chinese NER. Jia et al. [30] used TrueType font to convert each Chinese character font into an 8-bit grayscale image with the size of 48x48. Each Chinese character glyph feature vector is extracted by CNN, which is concatenated with the pre-trained character vector, then input into the BiLSTM-CRF model. The experimental results showed that the glyph feature vector effectively improves the performance of NER.

3. **Methodology.** NER is a classic sequence tagging problem. In this paper, the input sentence of the model is represented as $X = \{x_1, x_2, ..., x_n\}$, where x_i represents the ith character of sentence X, and the tag sequence of the sentence is represented as $y = \{y_1, y_2, ..., y_n\}$, where y_i represents the prediction tag for the ith character. The goal is to learn a function $f_\theta : X \to y$ to obtain the tag type of all characters in the input text.

3.1. **Overall Model Architecture.** This paper proposes a Chinese NER model based on BiLSTM-Attention-CRF with multi-feature fusion. This model is a character-based sequence tagging model. The overall architecture is shown in Figure 1 and includes the following three parts: input feature layer, BiLSTM-Attention layer, and CRF output layer. Firstly, the original character vector, the stroke-level glyph feature vector of Chinese character captured by the FSRCNN module from the five-stroke input method, and the affix feature vector extracted by using the left and right character embeddings are fused to construct a comprehensive character vector representation to enrich the input feature layer information. Its structure is shown in Figure 2. Secondly, BiLSTM is used to

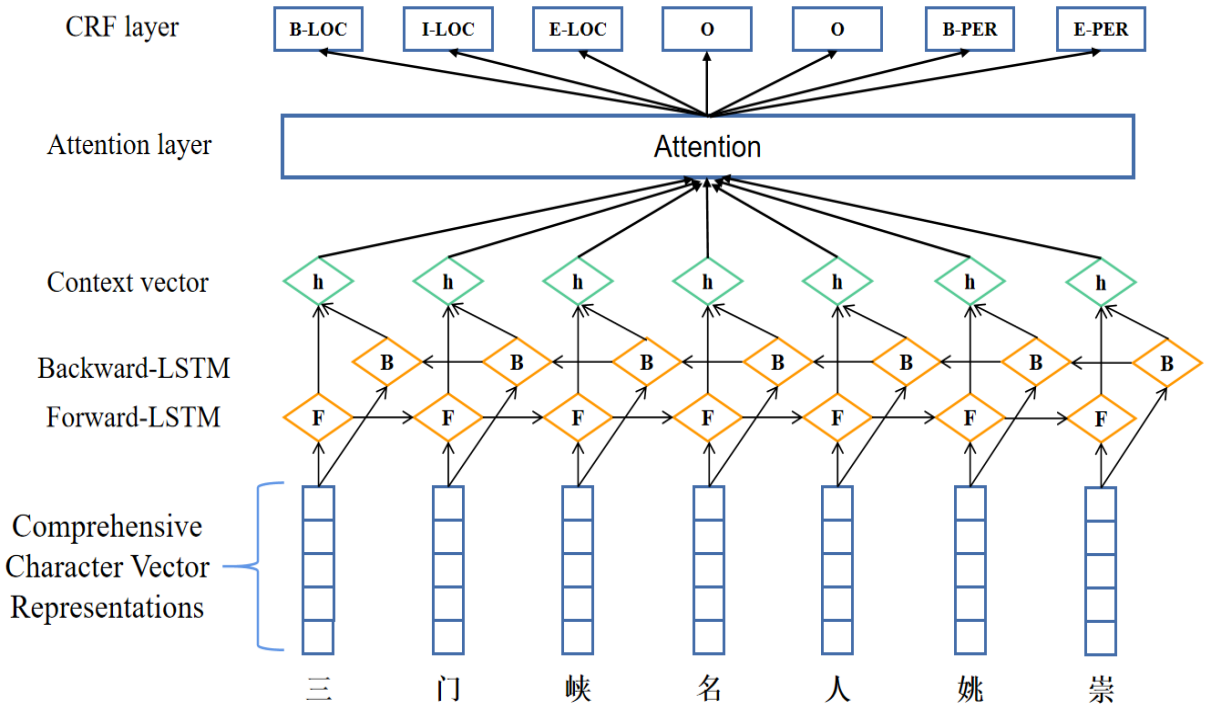

FIGURE 1. The overall architecture of the model

encode the context information of the comprehensive character vector representation, so as to capture the advanced features of the sentence, and the attention mechanism is adopted to enhance the model's utilization of key characters in the sentence to improve the output of BiLSTM. Finally, CRF is used to classify the output of BiLSTM-Attention, and the category tag corresponding to each character of the input sentence of the model was obtained.

3.2. **Input Feature Layer.**

3.2.1. *Character Embedding.* Assuming that the input of the model is a sentence $X = \{x_1, x_2, ..., x_n\}$ composed of n characters, each character x_i in the original text is mapped to a character vector $u_i \in R^{d_c}$ by word2vec. For each character in the sentence X, there is a character embedding matrix lookup table $e^c \in R^{|V| \times d_c}$, where $|V|$ represents the size of the character set, and d_c represents the dimension of the character vector. Each character can be converted into a character vector representation through the character embedding matrix lookup table, as shown in equation (1):

$$u_i = e^c(x_i) \tag{1}$$

The character sequence of sentence X undergoes the above mapping to generate a character vector sequence $U = \{u_1, u_2, ..., u_n\} \in R^{n \times d_c}$.

3.2.2. *FSRCNN Module.* As hieroglyphics, Chinese characters contain abundant glyph information such as radical, stroke, and five-stroke. The addition of glyph information to the character embedding can enhance the semantic expression of characters and effectively improve the performance of Chinese NER. Chinese characters with similar radical usually have similar semantic information. For example, the Chinese characters "晴 (clear)", "明 (bright)", and "晓 (dawn)" all have the radical "日 (sun)", so the meaning of these characters are related to the sun. However, the radicals only provide part of the glyph information of Chinese characters, and cannot fully express the semantic features of Chinese

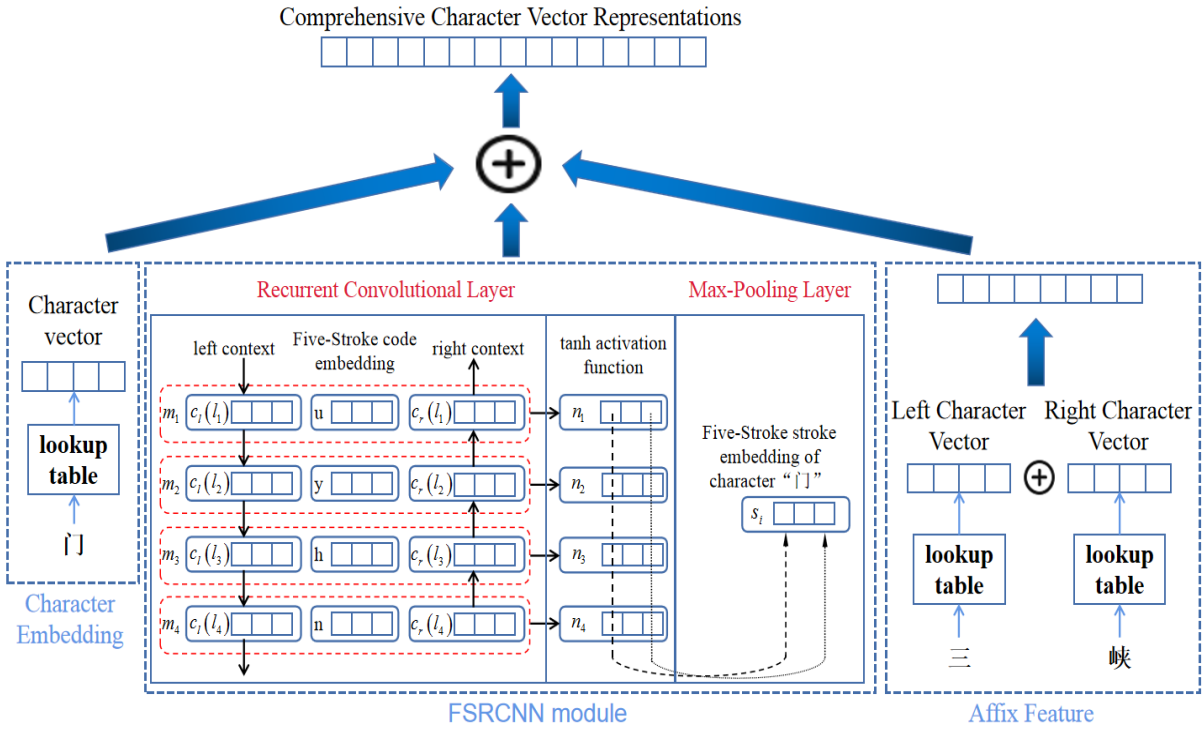

FIGURE 2. The overall architecture of the model

characters. Cao et al. [31] proposed a word embedding model based on n-gram strokes, which splits each Chinese character in the word into multiple strokes, each stroke is represented by an integer of 1 to 5. The neural network is used to extract the stroke features of Chinese characters and add them into the model, which can effectively learn the semantic features and morphological features of Chinese characters, and achieve good results in semantic analysis, text classification, and NER tasks. However, this method will bring fuzzy information to the system in the case of different characters with the same stroke. This paper uses the five-stroke input method to encode the input characters, and each Chinese character is represented by four English letters (keys) at most. In the rules of the five-stroke input method, English letters a-y are divided into 5 regions, and each region represents a basic stroke, as shown in Table 1. Each key corresponds to multiple roots, and each root is composed of these five basic strokes, "z" stands for a wildcard. For example, ("三", "门", "峡", "名", "人", "姚", "崇") which means ("Sanmenxia", "celebrity", "Yao Chong") in English, can be encoded into ("dggg", "uyhn", "mguw", "qkf", "wwww", "viqn", "mpfi"). When using the n-gram strokes method proposed in [31], the characters "右 (right)" and "石 (stone)" are encoded into the same representation form, while the five-stroke method can encode the two characters into different representation forms, so as to better distinguish the characters with similar structure, as shown in Table 2.

TABLE 1. Five regions of the five-stroke input method and their corresponding strokes

Regions	asdfg	hjklm	qwert	yuiop	xcvbn
Strokes	Horizontal(一)	Vertical(丨)	Left-falling(丿)	Right-falling(丶)	Hook(乙)

TABLE 2. Comparison of Stroke and Five-Stroke methods

Characters	n-gram Strokes		Five-Stroke	
	Decomposition	Code	Decomposition	Code
右	一ノ 丨ㄱ 一	13251	ナ口	dkf
石	一ノ 丨ㄱ 一	13251	石一ノ一	dgtg

RNN can analyze the text character by character, and store the semantic information of the preceding part of the text in a fixed-size hidden layer, so as to learn the sentence context information effectively. However, RNN is a biased model, which leads to the higher importance of the later characters in a sentence, and it is easy to forget the previous characters. This may affect the classification result, because the key characters may appear anywhere in the sentence. CNN is an unbiased model, which can effectively extract local features of sentences and use the max-pooling layer to obtain the most important features. However, the convolution window of CNN is not easy to determine. Selecting too small may lose key features, and selecting too large may result in huge parameter space.

In order to solve the above problems, the FSRCNN module was constructed by referring to the RCNN method in [32]. The structure of the FSRCNN module is shown in Figure 2. The RCNN is used to automatically capture the stroke-level glyph feature of Chinese characters from the five-stroke input method to obtain the semantic information inside Chinese characters. Firstly, through the five-stroke lookup table, each character in the sentence is encoded into an English letter sequence $\{w_1, w_2, ..., w_k\}\,(0 \leq k \leq 4)$, which is converted into vector representing $\{l_1, l_2, ..., l_k\} \in R^{k \times d_m}$ by random initialization embedding matrix, that is, the current five-stroke encoding embedding of the character. Where d_m is the dimension of vector l_j. To ensure that the vector matrix of the five-stroke code embedding of each character is 4-dimensional, for the letter sequence of $k < 4$, $4 - k$ dimensional 0 vector needs to be filled. Secondly, forward and backward LSTMs are used to capture the left context information $c_l\,(l_j) \in R^{d_l}$ and right context information $c_r\,(l_j) \in R^{d_l}$ of l_j respectively. Where the dimension of the hidden layer of the forward and backward LSTMs are both d_l, and their structure is shown in Figure 3 and Figure 4. Finally, $c_l\,(l_j)$, the vector representation l_j of w_j, and $c_r\,(l_j)$ are concatenated by equation (2) to obtain the final vector representing $m_j \in R^{d_m+2d_l}$ of the five-stroke code l_j for each character.

$$m_j = [c_l\,(l_j)\,; l_j; c_r\,(l_j)] \tag{2}$$

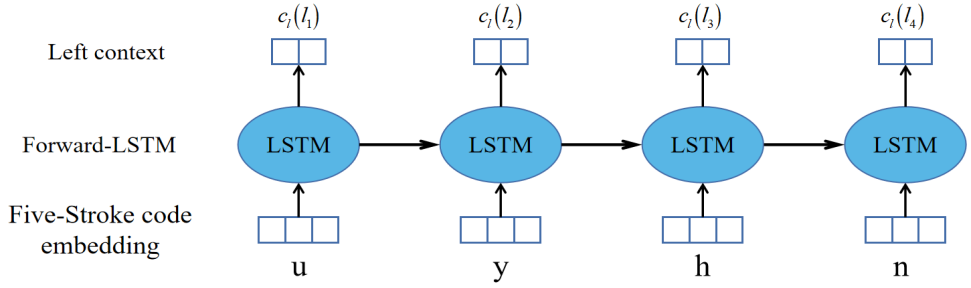

FIGURE 3. The left context information of the five-stroke code of the Chinese character "丨冂"

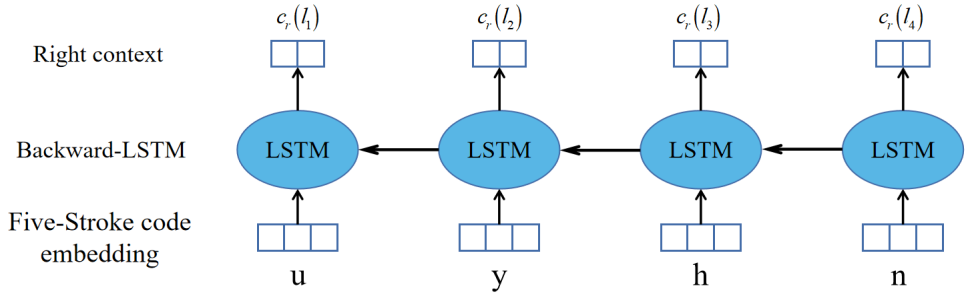

FIGURE 4. The right context information of the five-stroke code of the Chinese character "讠冂"

Since m_j already contains the own information of the five-stroke code l_j of each character and its context information, a convolution kernel matrix W_k with the number of rows is 1 and the number of columns is the same as the dimension of m_j is constructed, and the convolution operation is performed with m_j. Where the stride and number of the convolution kernel are set as 1 and d_k respectively. After convolution, the *tanh* activation function is applied to obtain the nonlinear result n_j , as shown in equation (3):

$$n_j = tanh\left(W_k \cdot m_j + b_k\right) \tag{3}$$

where b_k is a bias term. To reduce the number of parameters in the model and extract the most representative feature vector in the hidden semantic vector, this paper uses max-pooling to pool n_j to obtain the five-stroke level feature vector representation $s_i \in R^{d_k}$ of each character, as shown in equation (4) and equation (5):

$$s_i^l = max\left\{n_1^l, n_2^l, ..., n_k^l\right\} (l = 1, 2, ..., d_k) \tag{4}$$

$$s_i = \left\{s_i^1, s_i^2, ..., s_i^l, ..., s_i^{d_k}\right\} \tag{5}$$

where n_k^l represents the *lth* dimension vector in n_k, s_i^l represents the *lth* dimension vector in s_i. The character sequence of sentence X forms a stroke-level feature vector sequence $S = \{s_1, s_2, ..., s_n\} \in R^{n \times d_k}$ through the above mapping.

3.2.3. *Affix Feature.* In the task of Chinese NER, the left and right characters of each character in the sentence may represent the prefix and suffix information of the word, such as "冀州 (Jizhou)", "徐州 (Xuzhou)" and other LOC with "州 (state)" as the suffix. Therefore, this paper adds left and right character vectors to learn potential affix features. Assuming that the input of the model is a sentence $X = \{x_1, x_2, ..., x_n\}$ consisting of n characters, after the corresponding character vector sequence U is obtained through the character embedding matrix lookup table e^c , the left character vector and the right character vector of each character are sequentially extracted and concatenated to form the affix feature vector $v_i \in R^{2d_c}$ corresponding to each character vector u_i , as shown in equation (6). The left character vector of the first character of the sentence and the right character vector of the last character of the sentence are replaced by their own vectors:

$$v_i = \left[e^c\left(x_{i-1}\right); e^c\left(x_{i+1}\right)\right] \tag{6}$$

where $e^c\left(x_{i-1}\right)$ represents the left character vector of the *ith* character, $e^c\left(x_{i+1}\right)$ represents the right character vector of the *ith* character. Through the above mapping, the character sequence of sentence X forms the affix feature vector sequence $V = \{v_1, v_2, ..., v_n\} \in R^{n \times 2d_c}$.

3.2.4. *Comprehensive Character Vector Representations.* After concatenating the original character-level embedding, the stroke-level feature vector, and the affix feature vector, the comprehensive character vector representation of each character is formed, which are used as the input of the BiLSTM layer. The comprehensive character vector representation of each character is shown in equation (7):

$$c_i = u_i \oplus s_i \oplus v_i \tag{7}$$

where $c_i \in R^{3d_c+d_k}$, $\oplus$ is a concatenated operation. The character sequence of sentence X generates the comprehensive character vector sequence $C = \{c_1, c_2, ..., c_n\} \in R^{n\times(3d_c+d_k)}$ through the above mapping.

3.3. **BiLSTM-Attention Layer.** RNN can effectively capture the timing information, but gradient disappearance and gradient explosion will occur when obtaining the long-distance dependent information. As a variant of RNN, LSTM [33] can effectively deal with the long-distance dependence and gradient disappearance problems faced by RNN. It is mainly due to the existence of three "gate" structures with different functions in the LSTM unit, which are the forget gate, the input gate, and the output gate. This kind of "gate" structure can control the choice of information in the transmission process. At time t, the LSTM unit implements information update through equation (8)-equation (11):

$$\begin{bmatrix} f_t \\ i_t \\ \tilde{c}_t \end{bmatrix} = \begin{bmatrix} \sigma\left(W_{xf} \cdot x_t + W_{hf} \cdot h_{t-1} + b_f\right) \\ \sigma\left(W_{xi} \cdot x_t + W_{hi} \cdot h_{t-1} + b_i\right) \\ tanh\left(W_{xc} \cdot x_t + W_{hc} \cdot h_{t-1} + b_c\right) \end{bmatrix} \tag{8}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \tag{9}$$

$$o_t = \sigma\left(W_{xo} \cdot x_t + W_{ho} \cdot h_{t-1} + b_o\right) \tag{10}$$

$$h_t = o_t \cdot tanh\left(c_t\right) \tag{11}$$

where f_t, i_t, and o_t respectively represent the forget gate, input gate, and output gate of the LSTM unit, $\tilde{c}_t$ represents the state of the temporary unit, c_t represents the unit state at time t, h_t represents the output of the hidden layer at time t, x_t represents the input vector of LSTM at time t, σ represents the sigmoid activation function, $tanh$ represents the hyperbolic tangent activation function, W_{xf}, W_{xi}, W_{xc}, W_{xo} represent the weight matrix of the input vector corresponding to different control gates, W_{hf}, W_{hi}, W_{hc}, W_{ho} represent the weight matrix of the hidden layer, b_f, b_i, b_c, b_o represent the bias matrix.

The one-way LSTM can only obtain the information of the preceding part of the sentence. In order to make full use of the context information of the sentence, this paper uses BiLSTM to analyze the sentence from the forward and backward respectively, which not only saves the information of the preceding part of the sentence but also considers the future information of the sentence, so that it can achieve better results in the task of NER. Figure 1 shows the BiLSTM structure used in this paper. Assumed that the input of BiLSTM is the comprehensive character vector sequence $C = \{c_1, c_2, ..., c_n\}$ of sentence X generated by equation (7). The BiLSTM is used to generate forward and backward hidden state sequences $\overrightarrow{H} = \left\{h_1^f, h_2^f, ..., h_n^f\right\}$ and $\overleftarrow{H} = \left\{h_1^b, h_2^b, ..., h_n^b\right\}$ respectively, and then h_i^f and h_i^b are concatenated to obtain each hidden state h_i of BiLSTM, as shown in equation (12):

$$h_i = h_i^f \oplus h_i^b \tag{12}$$

Considering that different component information in the sentence has a different influence on the result of NER of the model, this paper introduces the attention mechanism after the BiLSTM layer. The attention mechanism can imitate the cognitive mode of human beings and selectively allocate the limited information processing ability, that is,

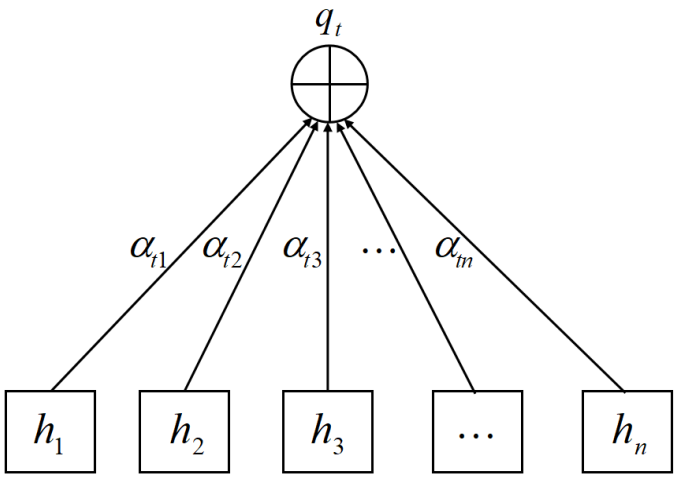

FIGURE 5. The structure of the attention layer

selectively pay attention to some important information and ignore other information received at the same time. Figure 5 shows the structure of the attention layer used in this paper. Assume that the feature vector matrix output by BiLSTM is $H = \{h_1, h_2, ..., h_n\}$, the attention mechanism is used to calculate the corresponding weights of all feature vectors, and the product of each feature vector and the corresponding weight are summed to form a vector matrix $Q = \{q_1, q_2, ..., q_n\}$, whose expression form is shown in equation (13)-equation (15):

$$e_{tj} = V_a^{\mathsf{T}} tanh\left(W_a q_{t-1} + U_a h_j\right) \tag{13}$$

$$\alpha_{tj} = \frac{exp\left(e_{tj}\right)}{\sum_{j=1}^{n} exp\left(e_{tj}\right)} \tag{14}$$

$$q_t = \sum_{j=1}^{n} \alpha_{tj} h_j \tag{15}$$

where e_{tj} represents the attention weight between the output of the attention layer at time t and the feature vector h_j output by the BiLSTM layer, which is calculated from the feature vector q_{t-1} output by the attention layer at the previous moment and h_j. W_a represents the weight of q_{t-1}, U_a represents the weight of h_j, V_a represents the overall weight, V_a^{T} represents the transposition of V_a, α_{tj} represents the normalized weight. q_t represents the feature vector output by the attention layer at the current moment, which is calculated by the sum of the product of each feature vector $h_j\,(j = 1, 2, ..., n)$ and the corresponding weight α_{tj}.

3.4. **CRF Layer.** CRF can consider the relationship between adjacent tags, enhance the constraints of the front and back tags, and obtain a globally optimal tag sequence. These constraints can be automatically learned by the CRF layer during training data. For example, B-PER cannot be followed by I-ORG. Therefore, this paper uses CRF to decode the information vector generated by BiLSTM-Attention to obtain the optimal tag sequence. Assuming that the input of the model is a sentence $X = \{x_1, x_2, ..., x_n\}$ consisting of n characters, the matrix P is defined as the output result of the vector matrix Q through the fully connected layer. Where n is the number of characters in the sentence, the size of P is $n \times k$, k is the number of tags in the tag set. For a predicted tag

sequence $y = \{y_1, y_2, ..., y_n\}$, its probability can be expressed as shown in equation (16):

$$S(X, y) = \sum_{i=1}^{n} P_{i,y_i} + \sum_{i=0}^{n} A_{y_i, y_{i+1}} \tag{16}$$

where P is the state matrix, P_{i,y_i} represents the probability that the predicted tag of the ith character is y_i, y_i is one of the tags in the tag set, A is the transition matrix, $A_{y_i,y_{i+1}}$ represents the probability of transferring from tag y_i to tag y_{i+1}. Adding y_0 and y_{n+1} indicates the tags of the begin and end of the predicted sentence, so A is a square matrix of size $k + 2$. The normalized probability of the predictive tag sequence y is shown in equation (17):

$$P(y|X) = \frac{e^{s(X,y)}}{\sum_{\tilde{y} \in Y_X} e^{s(X,\tilde{y})}} \tag{17}$$

where Y_X represents the set of all possible tag sequences, including tag sequences that do not comply with the BIEOS tagging rules, $\tilde{y}$ represents a possible tag sequence in Y_X.

In the training process, the log-likelihood method is used to maximize the likelihood probability $P(y|X)$ of the correct tag sequence:

$$\log(p(y|X)) = S(X, y) - \log\left(\sum_{\tilde{y} \in Y_X} e^{s(X,\tilde{y})}\right) \tag{18}$$

An effective and reasonable output sequence can be obtained by equation (18). When decoding, the Viterbi algorithm is used to predict a tag sequence y^* with the largest overall probability in Y_X , as shown in equation (19):

$$y^* = \underset{\tilde{y} \in Y_X}{argmax}\ s(X, \tilde{y}) \tag{19}$$

4. **Experiments.**

4.1. **Data Sets and Tagging Strategy.**

In order to evaluate the performance of the model proposed in this paper, we have carried out experimental verification on the SIGHAN Bakeoff MSRA data set and the People's Daily data set respectively. Both data sets contain the same three entity types: PER, LOC, and ORG. From the MSRA data set, select 70% as the training set, 15% as the testing set, and 15% as the validation set. From the People's Daily data set, select 65% as the training set, 20% as the testing set, and 15% as the validation set.

The tagging strategies of named entities include BIO, BIEOS, etc. In this experimental data set, the BIEOS tagging strategy [34] is adopted, and each character is tagged as one of B (Begin), I (Inside), E (End), O (Outside), and S (Single). Where B, I, and E represent the begin, inside, and end of named entities, S represents a single character entity, and O represents nonentities. For example, the first character of LOC is tagged as B-LOC, the middle character of LOC is tagged as I-LOC, and the last character of LOC is tagged as E-LOC. Dai et al. [35] report that the BIEOS strategy can demarcate entity boundaries more clearly than the BIO strategy because BIEOS can have more detailed position information.

4.2. **Evaluation Indexes.**

In this experiment, Precision (P), Recall (R), and F-score (F) are used to evaluate the performance of the model, where F can reflect the overall evaluation effect. The experimental samples are divided into positive and negative. The number of positive samples predicted to be positive is marked as TP (TruePositive). The number of negative samples predicted to be positive is marked as FP (FalsePositive). The number of positive samples predicted to be negative is marked FN (FalseNegative). The

number of negative samples predicted to be negative is marked as TN (TrueNegative). The calculation methods of the three evaluation indexes are shown in equation (20)-equation (22).

$$P = \frac{TP}{TP + FP} \times 100\% \tag{20}$$

$$R = \frac{TP}{TP + FN} \times 100\% \tag{21}$$

$$F = \frac{2 \times P \times R}{P + R} \times 100\% \tag{22}$$

4.3. **Hyper-parameter Settings.** This experiment is based on the Pytorch deep learning framework. The error back-propagation is used to train the model, and the network parameters are fine-tuned by back-propagating gradients. After constant adjustment, the experimental hyper-parameter settings are shown in Table 3.

Table 3. Hyper-parameter settings

Parameters	**Details**
Character embedding size	$d_c = 200$
Five-Stroke code embedding size	$d_m = 50$
LSTM dimension in FSRCNN	$d_l = 50$
Number of convolution kernels	$d_k = 150$
LSTM dimension	$d_h = 200$
Optimizer	SGD
Learning rate	$\alpha = 0.005$
Decay rate	0.0001

4.4. **Experimental Results and Analysis.** In order to verify the performance of the NER model proposed in this paper, comparative experiments were carried out on the MSRA data set and the People's Daily data set with the current representative NER models. The comparative experimental models include traditional based on statistical machine learning models and the current mainstream based on deep learning neural network models. Table 4 shows the results of comparative experiments between the model proposed in this paper and the representative models on the two data sets.

The data in Table 4 shows that the performance of the NER model proposed in this paper is better than these of the comparison models on the MSRA data set, with the value of F-score reaching 91.44%. Compared with the traditional based on statistical machine learning model, the method proposed in this paper does not rely on any manual features and external knowledge bases, and can automatically extract the features in sentences with better performance. The F-score is 4.93% and 0.26% higher than that of [36] and [37] respectively. Compared with the models based on deep learning neural network, the NER effect of the model constructed in this paper is also improved, and the F-score is 0.49%, 0.35%, and 0.57% higher than that of [29], [30], and [38] respectively. The reason is that the model constructed in this paper can make full use of stroke-level glyph features, affix features, and key character information in original sentences, so as to obtain more semantic information and high-quality hidden features for NER tasks.

The data in Table 4 also show that the performance of the NER model proposed in this paper on the People's Daily data set is better than these of the comparison models based on deep learning neural network, with F-score reaching 92.68%. Compared with [39] and [40], F-score is increased by 3.26% and 1.58% respectively.

Table 4. Results of different models on two data sets

	Model	MARA			People's Daily		
		P	R	F	P	R	F
Traditional machine learning model	Zhou et al. [36]	88.94%	84.20%	86.51%	—	—	—
	Zhang et al. [37]	92.20%	90.18%	91.18%	—	—	—
Neural network model	Dong et al. [29]	91.28%	90.62%	90.95%	—	—	—
	Jia et al. [30]	91.63%	90.56%	91.09%	—	—	—
	Zhang et al. [38]	90.59%	91.15%	90.87%	—	—	—
	Wu et al. [39]	90.70%	87.56%	89.10%	90.30%	88.56%	89.42%
	Hou et al. [40]	—	—	—	92.65%	89.61%	91.10%
	Ours	**92.77%**	90.15%	**91.44%**	**94.08%**	**91.32%**	**92.68%**

4.5. **Ablation Analysis.** In order to verify the influence of different features and attention mechanism module in the model of this paper on NER task, ablation tests are carried out on the MSRA data set and the People's Daily data set. The input feature layer of the model in this paper includes three types of features: character vector features, stroke-level glyph features, and affix features. According to different combinations of input features, four groups of experiments were performed on the BiLSTM-CRF model, and the above four groups of experiments were redone by adding the Attention module to the BiLSTM-CRF model. The experimental results are shown in Table 5.

Table 5. Evaluation of different components on two data sets

Model	Features	MARA			People's Daily		
		P	R	F	P	R	F
BiLSTM-CRF	Char Emb	89.87%	88.65%	89.26%	90.66%	88.48%	89.56%
	+ Affix feature	91.87%	89.55%	90.69%	93.37%	90.54%	91.93%
	+ Stroke-level feature	91.65%	89.61%	90.62%	92.41%	91.31%	91.86%
	+ Stroke-level feature + Affix feature	93.04%	89.38%	91.17%	94.14%	90.99%	92.54%
BiLSTM-Attention-CRF	Char Emb	90.90%	88.96%	89.92%	91.65%	89.60%	90.61%
	+ Affix feature	92.15%	89.41%	90.76%	93.64%	90.33%	91.96%
	+ Stroke-level feature	91.76%	89.70%	90.72%	93.04%	91.19%	92.10%
	+ Stroke-level feature + Affix feature	92.77%	**90.15%**	**91.44%**	94.08%	**91.32%**	**92.68%**

From the data in Table 5, it can be seen that the stroke-level feature and affix feature have a significant improvement in the effect of NER, which indicates that the internal semantic information of Chinese characters extracted by the stroke-level feature and the word boundary information extracted by the affix feature have a positive impact on NER. When the character vector feature + stroke-level feature + affix feature are used as the input of the BiLSTM-CRF model, the effect of NER is the best. The F-score on the MSRA data set and the People's Daily data set reaches 91.17% and 92.54% respectively.

The data in Table 5 also shows that the attention mechanism can effectively improve the performance of the NER model. The above four groups of experiments were repeated by

adding the attention mechanism module to the BiLSTM-CRF model, and the F-score of each group has improved to different degrees. Among them, when character vector feature + stroke-level feature + affix feature are used as the input of the BiLSTM-Attention-CRF model, the effect of NER is the best. The F-score on the MSRA data set and the People's Daily data set reaches 91.44% and 92.68% respectively. It is indicated that the attention mechanism can enhance the model's utilization of key characters in sentences, thereby improving the performance of the NER model.

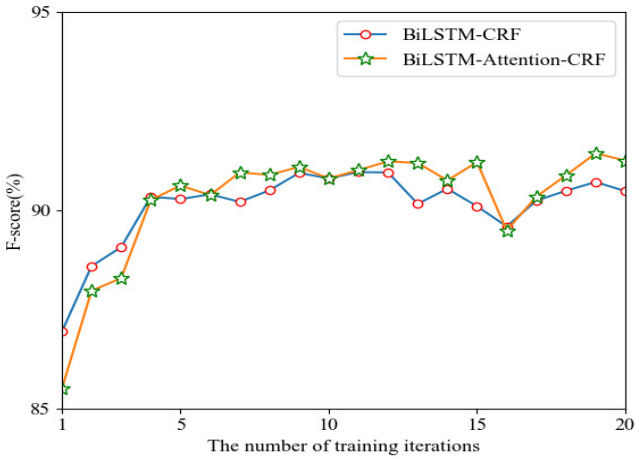

FIGURE 6. The comparison figure of the F-score of the two models on the MSRA data set changes with the number of training iterations

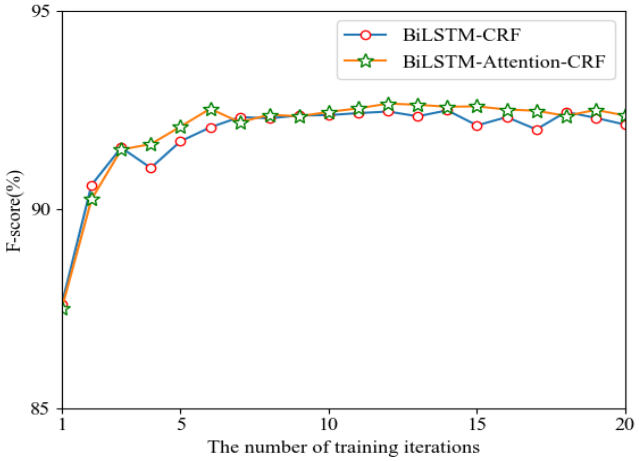

FIGURE 7. The comparison figure of the F-score of the two models on the People's Daily data set changes with the number of training iterations

In order to further analyze the effect of the attention mechanism, when the input features are character vector feature + stroke-level feature + affix feature, the comparison of the F-score of the BiLSTM-Attention-CRF model and the BiLSTM-CRF model on the MSRA data set and the People's Daily data set as the number of training iterations increases is plotted as shown in Figure 6 and Figure 7. It can be seen from Figure 6 that after the fourth iteration, the overall recognition effect of the BiLSTM-Attention-CRF model is better than that of the BiLSTM-CRF model, and the optimal F-score of

91.44% is reached in the 19th iteration. It can be seen from Figure 7 that after the third iteration, the overall recognition effect of the BiLSTM-Attention-CRF model is better than that of the BiLSTM-CRF model, and the optimal F-score of 92.68% is reached in the 12th iteration.

5. **Conclusions.** This paper proposes a Chinese NER model based on BiLSTM-Attention-CRF with multi-feature fusion. The model introduces a new type of FSRCNN structure to capture the stroke-level glyph feature of Chinese characters from the five-stroke input method and uses the embedding of left and right characters to extract the potential affix feature in words. Concatenate the original character vector with multiple feature vectors and then input them into the BiLSTM-Attention-CRF model for Chinese NER. It gives full consideration to the morphological information of Chinese characters and potential word information and effectively solves the problem of insufficient semantic information in models that only use character vector. The experimental results on the MSRA data set and the People's Daily data set show that the Chinese NER model constructed in this paper has better performance than most of the existing mainstream models without introducing any manual features and external knowledge bases.

In future research work, one is to explore different neural network combination models and more effective input features, so as to design a model structure that is more suitable for Chinese NER tasks. The second is to try to introduce a language model BERT with stronger feature coding capabilities, in order to further improve the existing NER effect.

Acknowledgment. This work is supported by the Natural Science Foundation of Fujian Province of China (No. 2019J01771) and the Scientific Research Project Foundation of Fujian University of Technology (No. GY-Z20046).

REFERENCES

[1] P. Zhou, W. Shi, J. Tian, Z. Qi, B Li, H. Hao, and B. Xu, Attention-based Bidirectional Long Short-term Memory Networks for Relation Classification, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Berlin, Germany, pp. 207–212, 2016.

[2] Y. Zeng, H. Yang, Y. Feng, Z. Wang, and D. Zhao, A Convolution BiLSTM Neural Network Model for Chinese Event Extraction, *International Conference on Computer Processing of Oriental Languages*, Springer, Cham, pp. 275–287, 2016.

[3] Z. Li, J. Huang, Z. Zhou, H. Zhang, S. Chang, and Z. Huang, LSTM-based Deep Learning Models for Answer Ranking, *2016 IEEE First International Conference on Data Science in Cyberspace (DSC)*, IEEE, pp. 90–97, 2016.

[4] Y.Abbaspour-Gilandeh, A. Molaee, S. Sabzi, N. Nabipur, S. Shamshirband, and A. Mosavi, A Combined Method of Image Processing and Artificial Neural Network for The Identification of 13 Iranian Rice Cultivars, *Agronomy*, vol. 10, no. 1, 117, 2020.

[5] Y. Shi, Y. Y. Zhu, J. Fang and Z. S. Li, Pose Measurement of Excavator Based on Convolutional Neural Network, *Journal of Network Intelligence*, vol. 6, no. 2, pp. 392–400, 2021.

[6] F. Zhang, T. Y. Wu, J. S. Pan, G. Ding and Z. Li, Human Motion Recognition Based on SVM in VR Art Media Interaction Environment, *Human-centric Computing and Information Sciences*, vol. 9, no. 1, pp. 1–15, 2019.

[7] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, Audio-visual Speech Recognition Using Deep Learning, *Applied Intelligence*, vol. 42, no. 4, pp. 722–737, 2015.

[8] E. K Wang, X. Liu, C. M Chen, S. Kumari, M. Shojafar, and M. S Hossain, Voice-Transfer Attacking on Industrial Voice Control Systems in 5G-Aided IIoT Domain, *IEEE Transactions on Industrial Informatics*, doi: 10.1109/TII.2020.3023677, 2020.

[9] B. Y. Lin, F. F. Xu, Z. Luo, and K. Zhu, Multi-channel Bilstm-Crf Model for Emerging Named Entity Recognition in Social Media, *Proceedings of the 3rd Workshop on Noisy User-generated Text*, Copenhagen, Denmark, pp. 160–165, 2017.

[10] G. Yang, and H. Xu, A Residual BiLSTM Model for Named Entity Reecognition, *IEEE Access*, vol. 8, pp. 227710–227718, 2020.

[11] P. Tang, P. Yang, Y. Shi, Y. Zhou, F. Lin, and Y. Wang, Recognizing Chinese Judicial Named Entity Using BiLSTM-CRF, *Journal of Physics: Conference Series*, IOP Publishing, vol. 1592, no. 1, 012040, 2020.

[12] S.Huang, X. Sun, and H. Wang, Addressing Domain Adaptation for Chinese Word Segmentation with Global Recurrent Structure, *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 184–193, 2017.

[13] Y. Zhu, G. Wang, and B. F. Karlsson, CAN-NER: Convolutional Attention Network for Chinese Named Entity Recognition, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, MN, USA, pp. 3384–3393, 2019.

[14] F. Yang, J. Zhang, G. Liu, J. Zhou, C. Zhou, and H. Sun, Five-stroke Based CNN-BiRNN-CRF Network for Chinese Named Entity Recognition, *CCF International Conference on Natural Language Processing and Chinese Computing*, Hohhot, China, pp. 184–195, 2018.

[15] Y. Zhang and J. Yang, Chinese NER Using Lattice LSTM, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, VIC, Australia, pp. 1554–1564, 2018.

[16] G. A. Levow, The Third International Chinese Language Processing Bakeoff: Word Segmentation and Named Entity Recognition, *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, Sydney, NSW, Australia, pp. 108–117, 2006.

[17] R. Grishman, The NYU System for MUC-6 or Where's The Syntax?, *Proceedings of the 6th conference on Message Understanding (MUC-6)*, Columbia, MD, USA, 1995.

[18] M. Collins and Y. Singer, Unsupervised Models for Named Entity Classification, *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 100–110, 1999.

[19] X. Ke and S. Li, Chinese Organization Name Recognition Based on Co-training Algorithm, *2008 3rd International Conference on Intelligent System and Knowledge Engineering*, Xiamen, China, pp. 771–777, 2008.

[20] A. Azpeitia, M. Cuadros, S. Gaines, and G. Rigau, NERC-fr: Supervised Named Entity Recognition for French, *International Conference on Text, Speech, and Dialogue*, Cham, Switzerland: Springer, pp. 158–165, 2014.

[21] A. McCallum and W. Li, Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-enhanced Lexicons, *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL*, pp. 188–191, 2003.

[22] K. Wang, C. M Chen, Z. Liang, M. M Hassan, G. M. L Sarne, L. Fotia, and G. Fortino, A Trusted Consensus Fusion Scheme for Decentralized Collaborated Learning in Massive IoT Domain, *Information Fusion*, vol. 72, pp. 100–109, 2021.

[23] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, Natural Language Processing (Almost) from Scratch, *Journal of machine learning research*, vol. 12, pp. 2493–2537, 2011.

[24] Z. Huang, W. Xu, and K. Yu, Bidirectional LSTM-CRF Models for Sequence Tagging, *arXiv preprint*, arXiv: 1508.01991, 2015.

[25] L. Luo, Z. Yang, P. Yang, Y. Zhang, L. Wang, H. Lin, and J. Wang, An Attention-based BiLSTM-CRF Approach to Document-level Chemical Named Entity Recognition, *Bioinformatics*, vol. 34, no. 8, pp. 1381–1388, 2018.

[26] Z. Liu, C. Zhu, and T. Zhao, Chinese Named Entity Recognition with A Sequence Labeling Approach: Based on Characters, or Based on Words?, *International Conference on Advanced Intelligent Computing Theories & Applications*, Changsha, China, pp. 634–640, 2010.

[27] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, Neural Architectures for Named Entity Recognition, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 260–270, 2016.

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

[29] C. Dong, J. Zhang, C. Zong, M. Hattori, and H. Di, Character-based LSTM-CRF with Radical-level Features for Chinese Named Entity Recognition, *Natural Language Understanding and Intelligent Applications*, Kunming, China, pp. 239–250, 2016.

[30] Y. Jia and X. Xu, Chinese Named Entity Recognition Based on CNN- BiLSTM-CRF, *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, pp. 1–4, 2018.

[31] S. Cao, W. Lu, J. Zhou, and X. Li, cw2vec: Learning Chinese Word Embeddings with Stroke n-gram Information, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, pp. 5053-5061, 2018.

[32] S. Lai, L. Xu, K. Liu, and J. Zhao, Recurrent Convolutional Neural Networks for Text Classification, *Proceedings of the AAAI Conference on Artificial Intelligence*, Austin, TX, USA, pp. 2267–2273, 2015.

[33] S. Hochreiter and J. Schmidhuber, Long Short-term Memory, *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] L. Ratinov and D. Roth, Design Challenges and Misconceptions in Named Entity Recognition, *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, Boulder, CO, USA, pp. 147–155, 2009.

[35] H. Dai, P. T. Lai, Y. C. Chang, and R. T. H. Tsai, Enhancing of Chemical Compound and Drug Name Recognition Using Representative Tag Scheme and Fine-grained Tokenization, *Journal of cheminformatics*, vol. 7, no. 1, S14, 2015.

[36] J. Zhou, L. He, X. Dai, and J. Chen, Chinese Named Entity Recognition with A Multi-phase Model, *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, Sydney, Australia, pp. 213–216, 2006.

[37] S. Zhang, Y. Qin, J. Wen, and X. Wang, Word Segmentation and Named Entity Recognition for SIGHAN Bakeoff3, *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, Sydney, NSW, Australia, pp. 158–161, 2006.

[38] N. Zhang, F. Li, G. Xu, W. Zhang, and H. Yu, Chinese NER Using Dynamic Meta-Embeddings, *IEEE Access*, vol. 7, pp. 64450–64459, 2019.

[39] Y. Wu, X. Wei, Y. Qin, and Y. Chen, A Radical-based Method for Chinese Named Entity Recognition, *Proceedings of the 2nd International Conference on Big Data Technologies*, pp. 125–130, 2019.

[40] C. Hou, M. Wang, and C. Li, Entity Subword Encoding for Chinese Long Entity Recognition, *China Conference on Knowledge Graph and Semantic Computing*, Singapore: Springer, pp. 123–135, 2019.