Hindawi
Complexity
Volume 2021, Article ID 6696064, 9 pages
https://doi.org/10.1155/2021/6696064

Research Article

MAF-CNER : A Chinese Named Entity Recognition Model Based on Multifeature Adaptive Fusion

Xuming Han,[1] Feng Zhou,[2] Zhiyuan Hao,[3] Qiaoming Liu,[4] Yong Li,[2] and Qi Qin[2]

[1]*College of Information Science and Technology, Jinan University, Guangzhou 510632, China*
[2]*School of Computer Science and Engineering, Changchun University of Technology, Changchun 130012, China*
[3]*School of Management, Jilin University, Changchun 130022, Jilin, China*
[4]*School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150006, China*

Correspondence should be addressed to Zhiyuan Hao; 15391910163@163.com and Qiaoming Liu; cslqm@hit.edu.cn

Received 18 December 2020; Revised 27 December 2020; Accepted 11 January 2021; Published 9 February 2021

Academic Editor: Abd E.I.-Baset Hassanien

Named entity recognition (NER) is a subtask in natural language processing, and its accuracy greatly affects the effectiveness of downstream tasks. Aiming at the problem of insufficient expression of potential Chinese features in named entity recognition tasks, this paper proposes a multifeature adaptive fusion Chinese named entity recognition (MAF-CNER) model. The model uses bidirectional long short-term memory (BiLSTM) neural network to extract stroke and radical features and adopts a weighted concatenation method to fuse two sets of features adaptively. This method can better integrate the two sets of features, thereby improving the model entity recognition ability. In order to fully test the entity recognition performance of this model, we compared the basic model and other mainstream models on Microsoft Research Asia (MSRA) and "China People's Daily" dataset from January to June 1998. Experimental results show that this model is better than other models, with F1 values of 97.01% and 96.78%, respectively.

1. Introduction

Word representation learning has been widely concerned as a basic problem in the field of natural language processing. Unlike traditional one-hot representations, low-dimensional distributed vocabulary representations (also called word embeddings) represent words as low-dimensional dense real number vectors, which can better capture the associated information between natural language words. This form of representation is very useful in some downstream tasks of natural language processing, for example, text classification [1], NER [2, 3], relation extraction [4, 5], and sentiment analysis [6, 7]. Therefore, how to obtain a better semantic representation of words is crucial.

In recent years of research, the NER model is mainly based on deep learning, and with the development of deep learning, more and more remarkable results have been achieved. The main basic model framework of English NER is Bidirectional Long Short-Term Memory-Conditional Random Field (BiLSTM-CRF) [8], which uses word embedding as the basic unit of predicting labels. English is a kind of phonetic alphabet, but Chinese characters represent typical meanings, so these research methods cannot be directly applied to Chinese. Unlike English, Chinese sentences do not have as obvious separators as in English. Therefore, when processing Chinese NER tasks, first use word segmentation tools to segment sentences, and then implement a sequence tagging model based on segmentation words. This method results in poor performance of CNER because CNER faces the following difficulties: (1) the quality of sentence segmentation has a great impact on the performance of NER. For example, "武汉市长江大桥" (Wuhan Yangtze River Bridge) as a whole location named entity, after segmentation by word segmentation tools, it may be segmented into "武汉" (Wuhan), "市长" (mayor), and "江大桥" (river's bridge). When these participles are used as input to the NER model, they will be recognized as three different named entities. (2) In order to solve the problem of word-level embedding, character-level embedding is widely used in NER tasks, but it still has many shortcomings. As shown in Figure 1, “人” (people), “八” (eight), and “乂” (yi) are semantically unrelated, and the stroke sequence is “丿” (left-falling stroke) and “㇏” (right-falling stroke). Chinese characters with the same stroke sequence have completely different semantics, and the stroke sequence of a Chinese character cannot uniquely identify a Chinese character. Similarly, using the radical feature alone will also encounter the same problem. In order to solve this problem, we can introduce another internal characteristic of Chinese characters—the roots; the radicals of “人”, “八”, and “乂” are “人”, “八” , and “丿” , respectively. Combining the two characteristics of this stroke and the radical can distinguish Chinese characters well. Figure 1

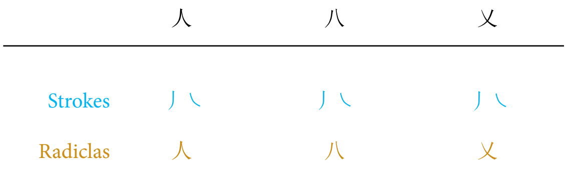

Figure 1: Strokes and radicals of “人”, “八”, and “乂”.

Integrating the internal characteristics of Chinese characters is effective for learning Chinese word embedding [9]. For example, Yin et al. used Convolutional Neural Networks (CNNs) to extract radical features, aiming to capture the intrinsic and intrinsic correlation of characters. Experimental results show that the model has achieved good performance in the field of Chinese clinical NER [10]. Chinese characters have rich internal structural features. How to better learn and use these features to improve the quality of Chinese character embedding is very important. It can be further studied on how to better combine the character characteristics of Chinese characters with the internal characteristics of Chinese characters. This article designs a multifeature adaptive fusion (MAF) method to fuse the stroke features and radical features of Chinese characters. This method can adaptively calculate the weight of the fusion stroke feature and radical feature. The main contributions of this article can be summarized as follows:

(1) This article integrates the characteristics of characters, strokes, and radicals into the BiLSTM-CRF model to fully represent the semantic information of Chinese characters.

(2) To achieve a better and more balanced fusion of the two sets of features, this article adopts the method of weighted series adaptive fusion features.

(3) Evaluation results show that this model achieves good performance on both the MSRA dataset and the 1998 China People’s Daily dataset.

The remainder of this article is structured as follows. Section 2 introduces the related work of CNER. Section 3 gives a detailed description of the MAF-CNER model. Section 4 presents extensive experiments to verify the effectiveness of our proposal, and Section 5 summarizes this work.

2. Related Works

The traditional solution to the NER problem mainly includes three methods: rule-based method, statistics-based method, and dictionary-based method [11]. The method based on rules and dictionaries requires professional linguists to write rules by hand, requires a lot of time, and has poor portability in different fields. In the task of NER, statistical methods mainly use Conditional Random Field (CRF) and Hidden Markov Model (HMM) [12, 13]. Although the accuracy rate is improved compared with the method based on rules and dictionaries, it still has disadvantages such as long training time.

With the continuous development of deep learning, researchers began to apply deep learning to NER tasks. Compared with traditional models, neural network models can learn deeper semantic feature information with almost no need for feature engineering [14] and domain knowledge. These models further improve the accuracy of entity recognition, especially the BiLSTM-CRF model [15, 16], and can significantly improve the performance of NER tasks.

The standard model for solving NER problems in the English domain is the BiLSTM-CRF model proposed by Huang et al. [17], which is more robust and less dependent on word embedding. Based on this structure, Lample et al. proposed to use BiLSTM to extract word representations on character-level embedding. Cho et al. proposed a deep learning NER model that effectively represents biomedical word tokens through the design of a combinatorial feature embedding, enhanced by integrating two different character-level representations extracted from CNN and BiLSTM [18]. In the Chinese field, CNER is more challenging [19]. Wang et al. proposed a CNN model based on a gating mechanism (GCNN) [20]. Cao et al. used Chinese character strokes as features and proposed the stroke n-gram model, which not only excavated the feature information of Chinese character strokes but also more effectively used the semantic information of Chinese characters to train word vectors [21]. Cao et al. proposed a novel adversarial transfer learning framework to make full use of the boundary information shared by tasks and prevent the task-specific functions of Chinese word segmentation [22]. Xu et al. proposed a simple and effective neural network framework ME-CNER (Multiple Embeddings for Chinese Named Entity Recognition), which embeds rich semantic information at multiple levels from radicals, characters to words [23]. Wu et al. proposed a radical-based CNER RCBC (R-CNN-BiLSTM-CRF). The RCBC-based model uses CNNs to automatically extract the semantics of the radicals of Chinese characters and combines the word vectors and radical vectors into joint vector. This method can reduce the semantic deviation of radical features and capture semantic information more accurately [24]. Ye et al. proposed a CNER model based on character-word vector fusion. This model reduces the dependence on the accuracy of word segmentation algorithms and effectively utilizes the semantic features of words [25]. In order to solve the ambiguity of Chinese words and the lack of word boundaries, Wu et al. proposed a novel fine-grained character-level representation method to capture the semantic information of Chinese characters [26]. Although the above methods have achieved good results, none of them have a more in-depth exploration of the internal characteristics of Chinese characters, and the fusion methods between multiple characteristics can be studied more deeply.

3. MAF-CNER Model

This section introduces the network layer organization structure of the “multifeature adaptive fusion Chinese named entity recognition model” model, as shown in Figure 2. The model is divided into three layers: character, stroke, and radical multifeature vector fusion layer; BiLSTM layer; CRF layer. The radical and stroke feature representations are calculated by the BiLSTM neural network, merged using the weighted concatenation method and concatenated with the character vector to form the final input vector. BiLSTM extracts the context features of the current input vector. The input of the CRF layer is the output vector of the BiLSTM layer, and the CRF layer will decode the information and obtain the best tag sequence. We will introduce the components of the Chinese clinial NER model based on MAF from bottom to top, as follows.

3.1. Character, Stroke, and Radical Multifeature Vector Fusion Layer. For a given sentence sequence $x = (c_1, c_2, \ldots, c_n)$, the embedding vector is composed of Chinese characters. Character characteristics are $e_c \in R^{d_c}$, radical characteristics, $e_r \in R^{d_r}$, and stroke characteristics, $e_s \in R^{d_s}$, respectively. As shown in Figure 3, the embedding vector of each character C_i can be expressed as follows:

$$e_i = e_i^c \oplus \left(m * e_i^r \oplus n * e_i^s\right). \quad (1)$$

3.2. Character Embedding. Character-level embedding has been widely used in natural language processing. Research shows that embedding pretrained characters in a specific field can improve system performance. For example, adding character-level features in neural machine translation [27, 28] can improve the translation performance of the system, text classification [29, 30], and NER also uses character-level representation. Therefore, the pretrained character embedding is better than the random initial character embedding. This article uses the Chinese Wikipedia corpus of May 2020 to pretrain Chinese character embedding through Word2Vec. After preprocessing, about 171M training corpus is finally obtained. The pretraining of character embedding is implemented with the Python version of Word2Vec in Gensim, and the dimension of the feature vector is set to 100.

3.3. Adaptive Fusion Representation of Strokes and Radical Features

3.3.1. Radical Features. A Chinese character is a kind of pictograph, and the radical is the first stroke or shape of a Chinese character. One of the most notable features of Chinese characters is that they contain a lot of semantic information at the radical level. The radicals of Chinese characters have a very important impact on the semantics of Chinese characters. For example, “胖” (fat) , “胸” (chest), and “肺” (lung). The main radical “月” (moon) is a simplified form of “肉” (flesh), which stands for meat, indicating that these characters are related to organs. A total of 228 radicals such as “鹿” (deer), “卤” (halogen), and “丶” (dot) are numbered from 1 to 228. However, the research in the traditional model mainly focuses on the semantic research at the phrase level.

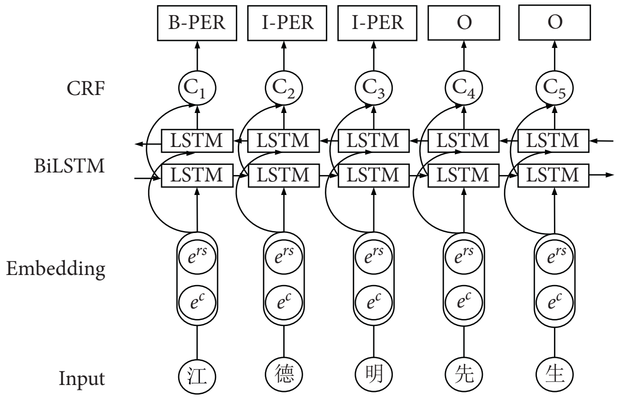

Figure 2: MAF-CNER model network structure.

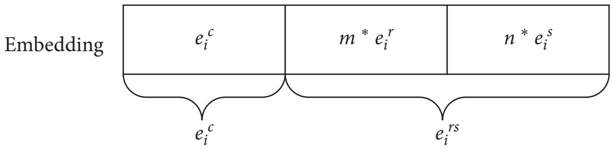

Figure 3: Characters, strokes, and radical vector links.

This article uses the BiLSTM network to extract the semantic information of the corresponding radicals of Chinese characters. Figure 4 shows the overall structure of the model in detail. The expression is as follows:

$$e_r = \lfloor \overrightarrow{h_t}; \overleftarrow{h_t} \rfloor. \quad (2)$$

In formula 2, $\left\lfloor \overrightarrow{h_t}; \overleftarrow{h_t} \right\rfloor$ is the hidden layer vector obtained by training BiLSTM network.

3.3.2. Stroke Characteristics. Stroke usually refers to the uninterrupted dots and lines of various shapes that compose Chinese characters, such as horizontal (“一”), vertical (“丨”), and left-falling stroke (“丿”) and dot (“丶”). It is the smallest continuous stroke unit that constitutes a Chinese character. As shown in Table 1, we divide the strokes into five types with the corresponding numbers of 1 to 5.

The Chinese character writing system provides a guide for the stroke order of each Chinese character. With this stroke information, we can decompose Chinese characters into strokes in a specific stroke order. This sequence information can be used when learning the internal semantic information of Chinese characters. Therefore, this article

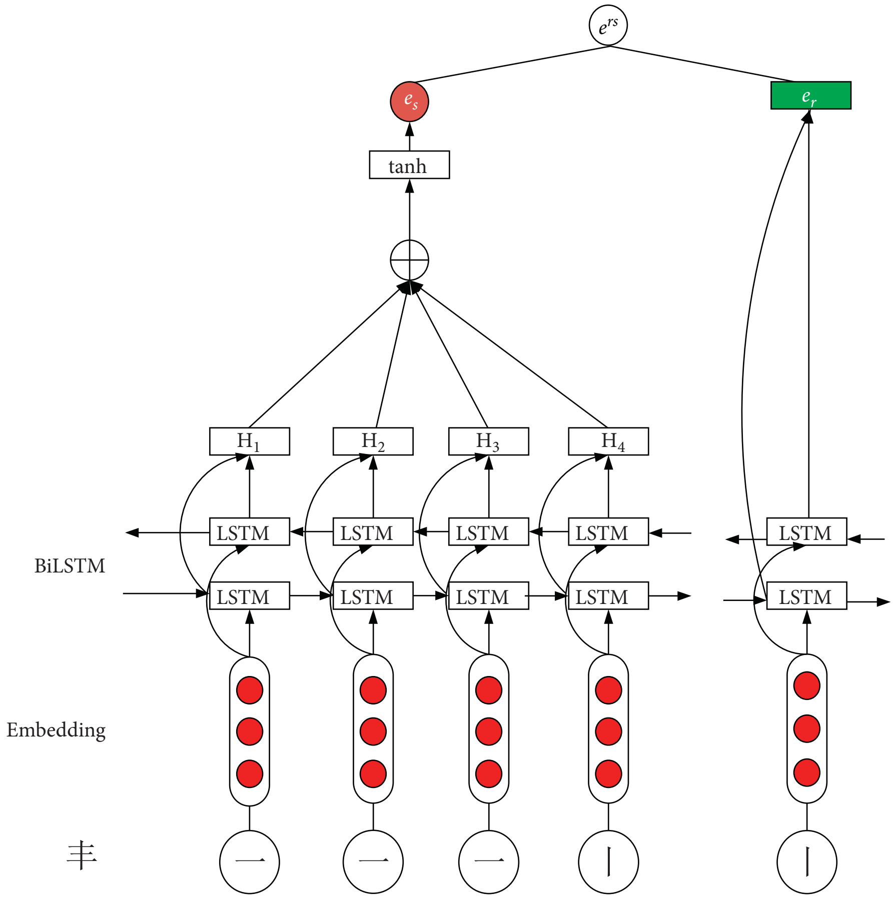

Figure 4: Stroke and radical feature calculation and fusion.

Table 1: Stroke type and number.

Stroke name	Horizontal	Vertical	Left-falling stroke	Right-falling stroke	Others
Type	㇐	㇑	㇓	㇏	㇛, ㇞ ㇉
Number	1	2	3	4	5

uses the BiLSTM network to extract the contextual semantic information of Chinese character strokes. Figure 4 shows the model structure. This method can learn more Chinese character graphic features. The expression is as follows:

$$e_s = \tanh\left(\sum_{j=1}^{N} H_{i,j}\right), \tag{3}$$

where $H_{i,j}$ is the j-th stroke feature vector of the i-th Chinese character.

3.3.3. Adaptive Feature Fusion. As shown in Figure 4, this article takes the stroke feature as the main feature, calculates its similarity with the character vector obtained by Word2Vec training, and determines its weight $\boldsymbol{m}$ according to formula 4.

$$m = e^{-\ \left(e_c \cdot e_s / \|e_c\| \times \|e_s\|\right)}, \tag{4}$$

where e_c is a character vector and e_s is a stroke vector.

The radical feature is used as an auxiliary feature, and the importance of the radical itself is calculated according to formulas 5 and (6), and its weight n is determined, and the weighted series method is used to fuse the two sets of features. This method can not only learn more graphic features of Chinese characters but also make the combination of the two features more balanced.

$$w_i = a \cdot e_r + b, \tag{5}$$

$$n = \operatorname{sigmoid}(w_i). \tag{6}$$

In formula 5, a and $\boldsymbol{b}$ are trainable parameters.

The features of adaptive fusion are expressed as follows:

$$e_i^{rs} = me_i^s \oplus ne_i^r. \tag{7}$$

3.4. BiLSTM Layer. Long Short-Term Memory (LSTM) is a variant of Recurrent Neural Network (RNN), which has been widely used in many natural language processing (NLP) tasks, such as NER, text classification, and sentiment analysis. It introduces the cell state and uses input gates, forget gates, and output gates to maintain and control information, which can effectively overcome the gradient explosion and gradient loss caused by the long-distance dependence of the RNN model. The mathematical expression of the LSTM model is as follows:

$$\begin{aligned}
i_t &= \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right),\\
f_t &= \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right),\\
o_t &= \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right),\\
\widetilde{c_t} &= \tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right),\\
c_t &= i_t * \widetilde{c_t} + f_t * c_{t-1},\\
h_t &= o_t * \tanh\left(c_t\right),
\end{aligned} \tag{8}$$

where σ represents the sigmod activation function. tanh represents the hyperbolic tangent function. X_t represents unit input. i_t, f_t, o_t represent the input gate, forget gate, and output gate at time W and b, respectively, which represent the weight and deviation of the input gate, forget gate, and output gate. $\widetilde{c_t}$ represents the current state of the input. c_t represents the update status at h_t that is the output at $\boldsymbol{t}$.

In order to use character context information at the same time, the model in this article uses BiLSTM to get the context vector of each character, which is a combination of forward LSTM and reverse LSTM. For a given sentence $x = (x_1, x_2, \ldots, x_n)$, we use $\overrightarrow{h_t}$ to represent the hidden layer state of the forward LSTM at time t, whereas $\overleftarrow{h_t}$ represents the reverse LSTM at time t . Hidden layer state, by linking the corresponding forward and reverse LSTM states, gets the final context vector $h_t = \left\lfloor \overrightarrow{h_t}; \overleftarrow{h_t} \right\rfloor$.

3.5. CRF Layer. Compared with the HMM, CRF does not have the strict requirements of the independence assumption of HMM and can effectively use both the internal information of the sequence and the external observation information, avoiding the problem of labeling bias and directly assuming the possibility of labeling and performing differentiated modeling. CRF can capture more dependencies: for example, “I-LOC” tags cannot follow “B-PER” [20]. In CNER, the input of CRF is the context feature vector learned from the BiLSTM layer. For input text sentence,

$$x = \left(x_1, x_2, x_3, \ldots, x_n\right). \tag{9}$$

Let $P_{i,j}$ denote the probability score of the j-th label of the i-th Chinese character in the sentence. For a prediction sequence $y = \{y_1, y_2, \ldots, y_n\}$, the CRF score can be defined as follows:

$$f(x, y) = \sum_{n=0}^{n+1} M_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i}, \tag{10}$$

where M is the transition matrix and $M_{i,j}$ represents the transition score from label i to $\boldsymbol{j}$. y_0 and y_{n+1} represent the start and end tags, respectively. Finally, we use softmax function to calculate the probability of the sequence y as follows:

$$P(y|x) = \frac{e^{f(x,y)}}{\sum_{\widetilde{y}\in Y_x} e^{f(x,\widetilde{y})}}. \tag{11}$$

During the training process, maximize the log probability of the correct label sequence:

$$\log(P(x, y)) = f(x, y) - \log\left(\sum_{\widetilde{y}\in Y_x} e^{f(x,\widetilde{y})}\right). \tag{12}$$

In the decoding stage, we predict that the maximum score obtained by the output sequence is as follows:

$$y^* = \underset{\widetilde{y}\in Y_x}{\operatorname{argmax}} f(x, \widetilde{y}). \tag{13}$$

In the prediction stage, the dynamic programming algorithm, Viterbi, is used to solve the optimal sequence.

4. Experiments and Results

4.1. Experimental Data and Evaluation Indicators. In order to evaluate the model proposed in this article on the task of CNER, this article conducted experiments on two different widely used datasets, namely, the MSRA dataset and the “China People’s Daily” dataset from January to June 1998. Table 2 shows the statistical information of the data set used in this article.

4.1.1. MSRA. It is a general dataset for CNER. The dataset contains three named entities: PER (person), LOC (location), and ORG (organization). The training set contains 46364 sentences, and the test set contains 4365 sentences. This article uses the ternary tag set {B, I, O} to mark, B represents the first word of the entity, I represents the remaining words of the entity, and O represents the non-entity word.

4.1.2. China People’s Daily. The China People’s Daily corpus was released by the Institute of Computational Linguistics of Peking University from January to June 1998. The entity categories are PER (person), LOC (location), and ORG (organization), also using the ternary tag set {B, I, O} for labeling. This article uses the data from January to May 1998 as the training set and the validation set. The validation set is 1/5 of the total data from January to May. The data from June 1998 is used as the test set.

Table 2: Dataset statement summary table.

Data set	Number of sentences		
	Train	Dev	Test
MSRA	46364	—	4365
China People’s Daily	82116	20529	21131

In order to fully evaluate the performance of the model, we use Precision (P), Recall (R), and harmonic average F1-score (F1) as the evaluation criteria for model performance, which is defined as follows:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \times 100\%,$$
$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \times 100\%, \tag{14}$$
$$F1 = \frac{2 * P * R}{P + R} \times 100\%.$$

TP (True Positive) indicates the correct number of samples in the positive examples, FP (False Positive) indicates the number of incorrect samples in the negative examples, and FN (False Negative) indicates the number of incorrect samples in the positive examples.

4.2. Model Building and Parameter Setting. The model in this article is built using PyTorch. PyTorch was launched by the Facebook Artificial Intelligence Research Institute (FAIR) in January 2017 based on Torch and is widely used in applications such as NLP. The experimental parameters are set as follows: embedding dimension (embedding_dim) is 300, input dimension max_length is 80, and training set batch_size: China People’s Daily dataset is 100, MSRA dataset is 128, and MSRA dataset is 64. The training learning rate is set to 0.001, in order to prevent overfitting during training; the weight decay factor weight_decay is set to $5e^{-4}$, dropout technology is used to prevent overfitting, and the value is set to 0.5.

4.3. Experimental Results. In order to more objectively evaluate the model performance of this model on the MSRA dataset and the “China People’s Daily” dataset, LSTM, BiLSTM, and BiLSTM-CRF models are used for performance analysis. The experimental results are shown in Table 3.

In Table 3, from the comparison of the experimental results of LSTM and BiLSTM, it can be seen that the latter performs better than the former. This verifies that the BiLSTM network can better capture the context information of the serialized text, with stronger learning ability that is better than LSTM . In the comparison between BiLSTM and BiLSTM-CRF, after adding the CRF module, it can be seen that the BiLSTM-CRF model has various aspects. Both are better than BiLSTM, which is mainly due to the fact that CRF considers the global label information in the sequence during the decoding process, which improves the performance of the model. Our model introduces two features of strokes and radicals on the basis of character-level embedding, and the test results on the two datasets achieve the best performance.

In order to verify the effectiveness of this method, it is compared with other mainstream NER methods. The specific results are shown in Tables 4 and 5. In Table 4, Chen et al. used CRF based on character features, and the F1 value was 86.20% [31]. The model of Zhou et al. used a multistage model. They used a character-level CRF model to segment the sequence. Then, word-level CRF layer was used to identify named entities, and the F1 value on the MSRA dataset reached 86.51% [32]. Zhou et al. took CNER as a joint recognition and classification task based on a global linear model [33]. The model used the rich manual feature model proposed in the literature [41] to greatly improve the performance of CNER. The F1 value of another BiLSTM-CRF neural network model proposed by Dong et al. was close to 90.95%. This model used both character-level and radical-level representations in the input of the model structure [34]. Zhang et al. used a lattice LSTM model for CNER. This model encodes the input character sequence and all possible words matching the dictionary. The F1 value of the model reached 93.18%, but the authors did not use the development dataset and trained the lattice LSTM mode [35]. Zhao et al. used a pretrained language model to encode the input sequence as a contextual representation and designed a new model that combines neural networks with BERT; the F1 value of the model reaches 95.28% [36]. However, using the model in this article, the F1 value reaches 97.01%. Johnson proposed the comprehensive embedding, which can take character, word, and position into account, has a valid structure, and can seize effective information. Regarding the test performance on MSRA dataset, F1 value reached 92.99%. Compared with the above model, this model has the best performance.

Table 5 shows test model performance using China People’s Daily dataset. Collobert et al. used a feedforward neural network, combined with preprocessing, affixes, and capitalization features, and achieved a result of 88.50% F1 [38]; Lample et al. input character-level word vectors into the BiLSTM-CRF model and achieved F1 value of 90.08% [19]; Chiu et al. combined BiLSTM with the CNN model and

Table 3: Comparison results between the model in this article and the basic model.

MSRA				People's Daily		
Model	Precision (%)	Recall (%)	F1 (%)	Precision (%)	Recall (%)	F1 (%)
LSTM	84.56	83.25	83.90	87.25	85.76	86.49
BiLSTM	86.77	84.93	85.84	88.34	87.61	87.97
BiLSTM-CRF	89.45	87.20	88.31	90.45	89.72	90.08
Ours	**97.21**	**96.83**	**97.01**	**97.08**	**96.47**	**96.78**

Table 4: Experimental results of MSRA dataset.

Model	Precision (%)	Recall (%)	F1 (%)
Chen [31]	91.22	81.71	86.20
Zhou [32]	88.94	84.20	86.51
Zhou [33]	91.86	88.75	90.28
Dong [34]	91.28	90.62	90.95
Zhang [35]	93.57	92.79	93.18
Zhao [36]	95.46	95.09	95.28
Johnson [37]	93.71	92.29	92.99
Ours	**97.21**	**96.83**	**97.01**

Table 5: . Experimental results of China People's Daily.

Model	Precision (%)	Recall (%)	F1 (%)
Collobert [38]	88.43	87.68	88.05
Lample [19]	90.45	89.72	90.08
Chiu [39]	91.94	90.06	90.99
Shen [40]	91.46	90.18	90.81
Ours	**97.08**	**96.47**	**96.78**

achieved 91.49% of advanced results [39]. Shen et al. proved that when deep learning is combined with active learning, the amount of labeled training data can be reduced. Although active learning can improve sample efficiency, it may be computationally expensive due to iterative retraining. In order to speed up the introduction of a lightweight architecture, the CNN-CNN-LSTM model consists of a convolutional character and word encoder and a LSTM tag decoder [40]. This article uses BiLSTM-CRF as the basic model and introduces two kinds of internal semantic information of Chinese character strokes and radicals. Model performance F1 increased to 96.78%.

5. Conclusion

In view of the insufficient representation of potential features of Chinese characters, this article uses BiLSTM network to learn the internal strokes and radical semantic information of Chinese characters and combines with the BiLSTM-CRF model to construct an adaptive multifeature fusion embedded CNER model. The assessment was conducted on the MSRA corpus and the corpus of China People's Daily" from Janu"ary to June 1998. Compared with other mainstream methods, the model in this article achieves the best results on both corpora. The biggest advantage of this model is that the weighted concatenation method is used to adaptively fuse two kinds of semantic information in Chinese characters, while previous research only stayed at the word-level embedding or used one kind of internal characteristic semantic information of Chinese characters. This will make the embedding layer insufficiently represented, the performance of the model will be relatively reduced, and the named entity cannot be correctly identified. Combining the two internal features can make Chinese character features more fully represented, avoiding the problem where a single feature cannot correctly distinguish Chinese characters, and the proportion of the two semantic information combinations is more balanced through weighting, and the best combination effect is achieved.

Data Availability

The datasets in the article could be downloaded from the URL: https://github.com/zhooufeng/Data.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The research was supported by the National Science Foundation of China under Grant Nos. 61572225 and 61472049, the Foundation of Jilin Provincial Education Department under Grant No. JJKH20190724KJ, the Jilin Province Science & Technology Department Foundation under Grant Nos. 20190302071GX and 20200201164JC, and the Development and Reform Commission Foundation of Jilin province under Grant No. 2019C053-11.

References

[1] H. A. Almuzaini and A. M. Azmi, “Impact of stemming and word embedding on deep learning-based Arabic text categorization,” *IEEE Access*, vol. 8, pp. 127913–127928, 2020.

[2] X. Yu, S. Mayhew, M. Sammons, and D. Roth, “On the strength of character language models for multilingual named entity recognition,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, October 2018.

[3] Y. Wang, Y. Sun, Z. Ma, L. Gao, and Y. Xu, “Named entity recognition in Chinese medical literature using pretraining models,” *Scientific Programming*, vol. 2020, Article ID 8812754, 9 pages, 2020.

[4] M. Miwa and M. Bansal, “End-to-end relation extraction using lstms on sequences and tree structures,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, August 2016.

[5] C. Alt, A. Gabryszak, and L. Hennig, “Probing linguistic features of sentence-level representations in relation extraction,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Washington, DC, USA, July 2020.

[6] Z.-X. Liu, D. G. Zang, G.-Z. Luo, M. Lian, and B. Liu, “A new method of emotional analysis based on CNN–BiLSTM hybrid neural network,” *Cluster Computing*, vol. 23, pp. 2901–2913, 2020.

[7] B. Peng, J. Wang, and X. Zhang, “Adversarial learning of sentiment word representations for sentiment analysis,” *Information Sciences*, vol. 541, pp. 426–441, 2020.

[8] M. Seok, H.-J. Song, C.-Y. Park, J.-D. Kim, and Y.-S. Kim, “Named entity recognition using word embedding as a feature,” *International Journal of Software Engineering and Its Applications*, vol. 10, no. 2, pp. 93–104, 2016.

[9] J. Yu, X. Jian, H. Xin, and Y. Song, “Joint embeddings of Chinese words, characters, and fine-grained subcharacter components,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, September 2017.

[10] M. Yin, C. Mou, K. Xiong, and J. Ren, “Chinese clinical named entity recognition with radical-level feature and self-attention mechanism,” *Journal of Biomedical Informatics*, vol. 98, Article ID 103289, 2019.

[11] N. Zhang, G. Xu, Z. Zhang, and F. Li, “MIFM: multi-granularity information fusion model for Chinese named entity recognition,” *IEEE Access*, vol. 7, pp. 181648–181655, 2019.

[12] N. V. Sobhana, P. Mitra, and S. K. Ghosh, “Conditional random field based named entity recognition in geological text,” *International Journal of Computer Applications*, vol. 1, no. 3, pp. 143–147, 2010.

[13] G. D. Zhou and S. Jian, “Named entity recognition using an HMM-based chunk tagger,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA, USA, July 2002.

[14] M. Zhang, Y. Zhang, and D. Tin Vo, “Neural networks for open domain targeted sentiment,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, September 2015.

[15] M. Habibi, L. Weber, M. Neves, D. L. Wiegandt, and U. Leser, “Deep learning with word embeddings improves biomedical named entity recognition,” *Bioinformatics*, vol. 33, no. 14, pp. i37–i48, 2017.

[16] Q. Wang, Y. Zhou, T. Ruan, D. Gao, Y. Xia, and P. He, “Incorporating dictionaries into deep neural networks for the Chinese clinical named entity recognition,” *Journal of Biomedical Informatics*, vol. 92, Article ID 103133, 2019.

[17] Z. Huang, W. Xu, and K. Yu, “Bidirectional LSTM-CRF models for sequence tagging,” 2015, http://arxiv.org/abs/1508.01991.

[18] M. Cho, J. Ha, C. Park, and S. Park, “Combinatorial feature embedding based on CNN and LSTM for biomedical named entity recognition,” *Journal of Biomedical Informatics*, vol. 103, Article ID 103381, 2020.

[19] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural architectures for named entity recognition,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, CA, USA, June 2016.

[20] C. Wang, W. Chen, and B. Xu, “Named entity recognition with gated convolutional neural networks,” in *Proceedings of the Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pp. 110–121, Nanjing, China, October 2017.

[21] S. Cao, W. Lu, J. Zhou, and X. Li, “cw2vec: learning Chinese word embeddings with stroke n-gram information,” in *Proceedings of the The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pp. 5053–5061, New Orleans, LA, USA, February 2018.

[22] P. Cao, Y. Chen, K. Liu, J. Zhao, and S. Liu, “Adversarial transfer learning for Chinese named entity recognition with self-attention mechanism,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, October 2018.

[23] C. Xu, F. Wang, J. Han, and C. Li, “Exploiting multiple embeddings for Chinese named entity recognition,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, Beijing, China, November 2019.

[24] Y. Wu, X. Wei, Y. Qin, and Y. Chen, “A radical-based method for Chinese named entity recognition,” in *Proceedings of the 2nd International Conference on Big Data Technologies*, Jinan, China, August 2019.

[25] N. Ye, X. Qin, L. Dong, X. Zhang, and K. Sun, “Chinese named entity recognition based on character-word vector fusion,” *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8866540, 7 pages, 2020.

[26] G. Wu, G. Tang, Z. Wang, Z. Zhang, and Z. Wang, “An attention-based BiLSTM-CRF model for Chinese clinic named entity recognition,” *IEEE Access*, vol. 7, pp. 113942–113949, 2019.

[27] J. Chung, K. Cho, and Y. Benjio, “A character-level decoder without explicit segmentation for neural machine translation,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, August 2016.

[28] M.-T. Luong and C. D. Manning, “Achieving open vocabulary neural machine translation with hybrid word-character models,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, August 2016.

[29] Y. Xiao and K. Cho, “Efficient character-level document classification by combining convolution and recurrent layers,” 2016, http://arxiv.org/abs/1602.00367.

[30] B. Liu, Y. Zhou, and W. Sun, “Character-level hybrid convolutional and recurrent neural network for fast text categorization,” in *Proceedings of the International Conference on Extreme Learning Machine*, Singapore, November 2018.

[31] A. Chen, F. Peng, R. Shan, and G. Guo-Zheng Sun, “Chinese named entity recognition with conditional probabilistic models,” in *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, Sydney, Australia, July 2006.

[32] J. Zhou, H. Liang, D. Xinyu, and C. Jiajun, “Chinese named entity recognition with a multi-phase model,” in *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, Sydney, Australia, July 2006.

[33] J. Zhou, W. Qu, and F. Zhang, “Chinese named entity recognition via joint identification and categorization,” *Chinese Journal of Electronics*, vol. 22, no. 2, pp. 225–230, 2013.

[34] C. Dong, J. Zhang, and C. Zong, “Character-based LSTM-CRF with radical-level features for Chinese named entity recognition,” in *Proceedings of the International Conference on Computer Processing of Oriental Languages National CCF Conference on Natural Language Processing and Chinese Computing*, pp. 239–250, Kunming, China, December 2016.

[35] Y. Zhang and Z. Yang, “Chinese ner using lattice lstm,” in *Proceedings of the The 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, Melbourne, Australia, July 2018.

[36] H. Zhao, M. Xu, and J. Cao, “Pre-trained language model transfer on Chinese named entity recognition,” in *Proceedings of the 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, October 2019.

[37] S. Johnson, S. Shen, and Y. Liu, “CWPC_BiAtt: character-word-position combined BiLSTM-attention for Chinese named entity recognition,” *Information*, vol. 11, no. 1, p. 45, 2020.

[38] 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, no. 12, pp. 2493–2537, 2011.

[39] J. P. C. Chiu and E. Nichols, “Named entity recognition with bidirectional LSTM-CNNs,” *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 357–370, 2016.

[40] Y. Shen, H. Yun, Z. C. Lipton, and Y. Kronrod, “Deep active learning for named entity recognition,” 2017, http://arxiv.org/abs/1707.05928.

[41] Y. ZhangS. Clark et al., “A fast decoder for joint word segmentation and POS-tagging using a single discriminative model,” in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, MIT Stata Center, MA, USA, October 2010.