Journal of Intelligent & Fuzzy Systems 45 (2023) 3779–3790
DOI:10.3233/JIFS-231554
IOS Press

Local and global character representation enhanced model for Chinese medical named entity recognition

Yan Xiang, Wei Liu, Junjun Guo* and Li Zhang
Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming, China

Abstract. Chinese medical named entity recognition (CMNER) aims to extract entities from Chinese unstructured medical texts. Existing character-based NER models do not comprehensively consider character's characteristics from different perspectives, which limits their performance in applying to CMNER. In this paper, we propose a local and global character representation enhanced model for CMNER. For the input sentence, the model fuses the spacial and sequential character representation using autoencoder to get the local character representation; extracts the global character representation according to the corresponding domain words; integrates the local and global representation through gating mechanism to obtain the enhanced character representation, which has better ability to perceive medical entities. Finally, the model sent the enhanced character representation to the Bi-LSTM and CRF layers for context encoding and tags decoding respectively. The experimental results demonstrate that our model achieves a significant improvement over the best baseline, increasing the F1 values by 1.04% and 0.62% on the IMCS21 and CMeEE datasets, respectively. In addition, we verify the effectiveness of each component of our model by ablation experiments.

Keywords: Named entity recognition, Chinese characters, medical entity, local and global representation

1. Introduction

Chinese medical named entity recognition (CMNER) aims to extract specific entities from unstructured medical texts and identify their types, such as symptoms, drug names, body parts, etc. This task is the basis of downstream tasks such as information retrieval and intelligent question-answering in the medical domain. In the past researches, named entity recognition(NER) is usually formalized as a sequence labeling problem, in which a sentence is divided into multiple tokens as the input, and the model outputs the corresponding entity tags of tokens. For Chinese NER, tokens commonly have two forms, i.e. character granularity and word granularity [1, 2]. The sequence model based on word granularity may encounter the problem of improper word segmentation, resulting in wrong entity boundaries. Therefore, most Chinese NER models are based on character granularity. However, Chinese characters do not contain independent semantics, and character-based models may result in the incomplete entity or wrong categories. To solve this problem, researchers mainly consider how to introduce much more word information for the character-based model. For example, Zhang et al. [3] proposed Lattice model, integrating the embedding of matched words with character embedding in a unique way to enhance character representation. Ma

*Corresponding author. Junjun Guo, Faculty of Information Engineering and Automation, Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming, 650500, China. E-mail: guojjgb@163.com.

ISSN 1064-1246/$35.00

et al. [4] proposed SoftLexicon, which classifies the matched words into four sets and integrates the word embedding of the four sets with character embedding. Liu and Xiao et al. [5] proposed a model that injects lexical features into the bottom layer of BERT, and character information and word information can fully interact in BERT.

The above methods show that character feature enhancement can make up for the ability deficiency of Chinese characters to independently perceive entities. However, these character enhancement methods do not comprehensively consider the character's characteristics from different perspectives, which limits their performance in applying to CMNER. In fact, Chinese characters in medical domain have useful characteristics for identifying domain entities, which is mainly manifested in two aspects:

1) The local characteristic. A Chinese character has a kind of glyph structure commonly composed of radicals and other components, and we call it the local characteristic. A character containing some specific radical usually belongs to a medical entity, and Chinese characters with the same medical radical component have similar medical entity categories. For example, Chinese characters constructed by the medical radical component "疒" are usually the characters of medical entities and mostly related to disease entities, such as "癌cancer", "疤Scar", "瘟plague" and "疫epidemic". Characters composed of the medical radical "月" are related to body parts, such as "脑brain", "肝liver", "肾kidney", "肺lung", etc. This local characteristic can be obtained by the spacial and sequential ways. As shown in Fig. 1, for the spacial way, a Chinese character can be regarded as a two-dimensional image with specific radical. On the other hand, from the perspective of sequence, a character can be disassembled into sequences of the radical and the other components. Spatial and sequential representation can highlight the radical from different aspects to enhance the ability of Chinese characters indicating medical entities.

2) The global characteristic. Medical words constituted by specific Chinese characters also indicate a specific entity. For example, the character "脑brain" in Figure 1 constitutes medical words such as "膜炎meningitis" and "脑梗cerebral infarction", while those domain words belong to medical entities. Medical words responding to a character need to be obtained from the entire medical dataset, so we call it the global characteristic. Using these medical words to obtain the global representation can further improve the entity-representing ability of Chinese characters.

Drawing on the aforementioned analysis, we introduce a novel approach to address the limitations of existing CMNER models. Our proposed method integrates both local and global information of Chinese characters, aiming to enhance the model's ability to accurately perceive potential entity boundaries and categories. By leveraging this comprehensive information, we expect to improve the overall performance of CMNER and provide more precise and reliable results in identifying medical entities. Comparing with previous works, the major contributions of our paper are three-fold:

1) We propose a local and global character representation enhanced model for CMNER. To the best of our knowledge, it is the first attempt to integrate the local and global information of Chinese characters to enhance their ability of perceiving the potential entity boundaries and categories, and thereby to improve the performance of NER.

(2) We use the autoencoding mechanism to fuse the spatial local representation and the sequential local representation, and use the interaction gating mechanism to control the contribution of local and global representation, so as to obtain the comprehensive character representation from different perspectives.

(3) Extensive experiments on two benchmark datasets show that our proposed method significantly outperforms the other baseline methods, and the in-depth analysis shows the effectiveness of local and global representation, as well as fusing and gating mechanism.

2. Related works

Before the emergence of deep learning, researchers mainly adopted traditional machine learning methods for NER. Lafferty [6] first proposed CRF, and McCallum [7] first successfully applied it to NER. Settles [8] proposed the method of combining CRF with feature set for biomedical NER task. Ju [9] uses SVM to identify the names of the specified type from the biomedical text, and Tang [10] developed a NER system based on SVM to identify the clinical entities in the hospital charging summary. In addition, they extracted two different types of word features and integrated them with the clinical NER system based on SVM. Liu [11] first established a medical dictionary, and then studied the role of different types of features in the Chinese clinical text NER

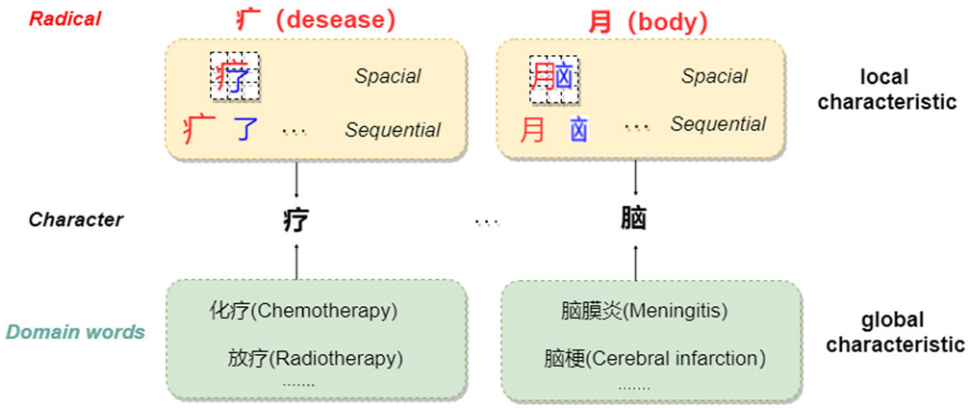

Fig. 1. Local and global characteristic of characters.

task based on CRF. Although the research has made great progress, they have to spend a lot of energy on feature engineering [12] to obtain better recognition results. In addition, there are still many problems such as high-dimensional sparse data, poor scalability, cold start and difficulties of user preference modeling, which makes medical NER based on traditional machine learning unsatisfactory.

For medical NER, Yao et al. [13] used neural networks to train a large number of medical texts to generate word vectors, and then build multi-layer CNNs for NER. Li et al. [14] used Bi-LSTM as the basic NER structure, and Zeng et al. [15] combined BiLSTM and CRF to achieve good results in the drug name recognition task. After that, the attention mechanism was introduced to highlight the important information in the input sequence. Luo et al. [16] constructed a model combining BiLSTM-CRF and the attention mechanism in document-level compound named entity recognition, and obtained global information by introducing attention mechanism to ensure the consistency of the same entity labeling in document-level data. Medical texts contained intricate information, including a large number of domain words and professional knowledge, which requires more efficient ways to solve the task of medical NER.

For Chinese NER, the comparison between word-based method and character-based method indicated that the latter was a better choice empirically [17, 18]. In order to compensate for the lack of semantic expression by characters alone, researchers integrated lexical information into character-based models. The typical method was Lattice-LSTM [3] which can automatically find more useful words from the context and pass to each character to enhance the character representation for NER. On this basis, many methods tried to improve Lattice method. Gui et al. [19] proposed the LR-CNN model to solve the problem that Lattice structures cannot effectively handle lexical information conflicts or parallelized computing. SoftLexicon [4] introduced lexical information through label and probability methods at the character representation layer. Yan et al. [20] took full advantage of these models in enhancing the ability of capturing remote context dependencies. Besides, there were also many methods exploiting external knowledge for Chinese NER [21, 22]. He et al. [23] used knowledge-graph to enhance word representation. Yin et al. [24] introduced the radical-level features to obtain more structural information, and used the self-attention mechanism to capture the dependencies between characters. Meng et al. [25] used Chinese character images to extract features such as strokes and structure of Chinese characters, achieving promising performance. In the above works, it has been proved that the feature extension and enhancement of Chinese character can result in better NER performance.

3. Methodology

In this section, we will introduce our proposed approach. Formally, we denote a Chinese sentence $\boldsymbol{C} = \{c_1, c_2 \cdots, c_i \cdots, c_n\}$ as an model input, where c_i is the i th character in the sentence. The model outputs BIO{Symptom, Drug, Body $\cdots$} tags for each character. The overall framework is illustrated in Fig. 2, which mainly consists of five modules: 1) local character representation, 2) global character representation, 3) local and global character representations integration, 4) context representation

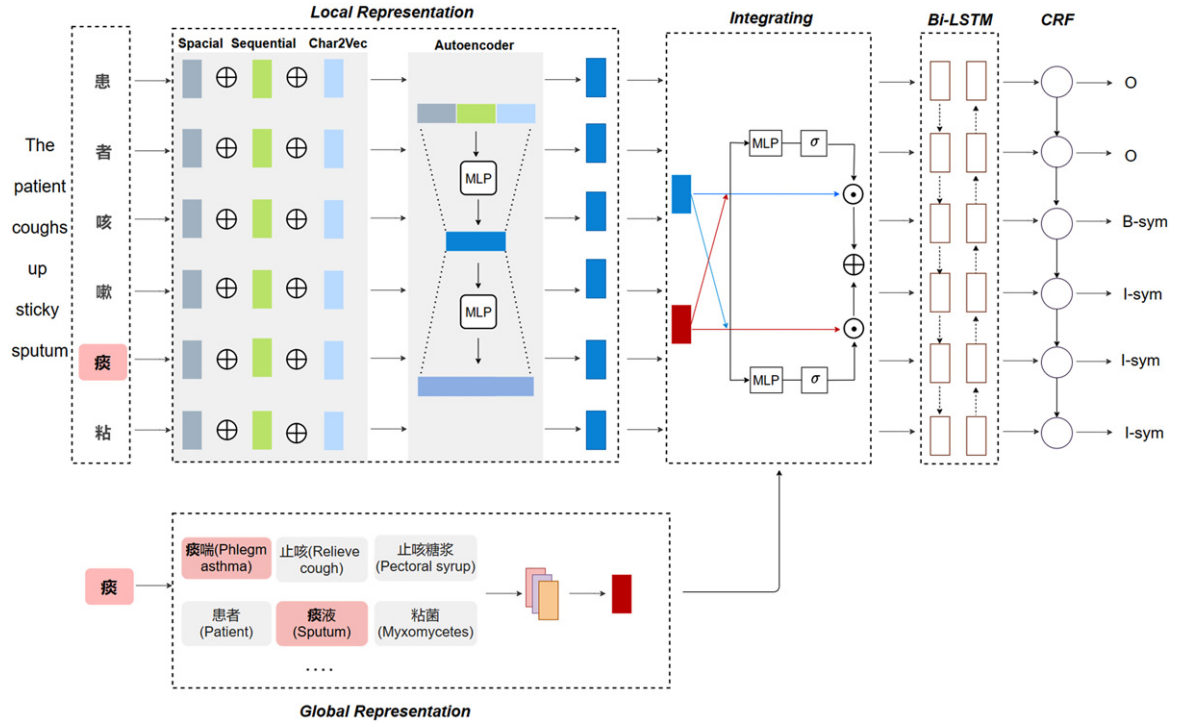

Fig. 2. The architecture of our model.

extraction based on Bi-LSTM, and 5) tags decoding base on CRF.

3.1. Local character representation

For each character c_i, We obtain the spacial representation x_i^s , the sequential representation x_i^o and the regular Char2vec representation x_i^w respectively, and then fuse them by autoencoder mechanism.

3.1.1. Spacial representation

First, we regard a character c_i as a two-dimensional image. From the perspective of spatial processing, we can obtain its spacial representation by the image feature encoder. Referring to the practice of the model proposed by Meng et al.[25], we converted c_i character into six 8-bit gray images corresponding to six different fonts $c_i^{image1}, c_i^{image2}, \ldots, c_i^{image6}$, where c_i^{imagej} is the j th font with 12×12 pixels. We concatenate different image pixel matrices to obtain the total structure image $c_i^{image} \in \mathbb{R}^{12\times12\times6}$:

$$c_i^{image} = Concat[c_i^{image1}; c_i^{image2}; \ldots; c_i^{image6}] \quad (1)$$

Where $Concat[.]$ is concatenation operation.

Then, we used a convolution operation $Conv1$ with 5×5 convolution kernel size and 384 output channels to obtain hidden vector $h_i^{image} \in \mathbb{R}^{8\times8\times384}$:

$$h_i^{image} = Conv1(x_i^{image}) \quad (2)$$

Next, we put h_i^{image} into a image max-pooling operation $Maxpooling1$ with template size of 4×4, and a convolution operation $Conv2$ with 1×1 kernel size and d_s output channels, to obtain hidden layer vector $h_i^s \in \mathbb{R}^{2\times2\times d_s}$.

$$h_i^s = Conv2(Maxpooling1(h_i^{image})) \quad (3)$$

We put h_i^s into the group convolution operation $GroupConv$ with convolution kernel size of 2, and perform the dimension conversion operation $Reshape$ to obtain the final spacial representation $x_i^s \in \mathbb{R}^{d_s}$ of the character.

$$x_i^s = Reshape(GroupConv(h_i^s)) \quad (4)$$

3.1.2. Sequential representation

As mentioned in the introduction, radical components in a character have strong function to indicate

entity, which may not be fully reflected only by the spacial representation. Therefore, we need to further emphasize the role of the radicals as independent components in a character. We divide a character into radicals and the other components, and then use the sequence feature encoder to extract the sequential representation of the character.

First, we split the character c_i into K parts $\boldsymbol{O_i} = \{o_i^1, o_i^2, \cdots, o_i^K\}$,according to the split structure of Chinese characters in the Xinhua dictionary. If the number of components is less than K, the vacant position is filled with "<PAD>". We randomly embed each of the character components using embedding operation E^r:

$$e_i^k = E^r(o_i^k) \tag{5}$$

The obtained random embedding sequence of characters $\{e_i^1, e_i^2, \cdots, e_i^K\}$ is sent to convolution operation $Conv3$ with convolution kernel size of 3 to obtain the character hidden vector sequence $h_i^l \in \mathbb{R}^{d_b}$:

$$h_i^l = Conv3(e_i^1, e_i^2, \cdots, e_i^K) \tag{6}$$

we apply a max-pooling operation $Maxpooling2$ to the hidden vector, and then send it to a full connection operation F_c to obtain the sequential local representation $x_i^o \in \mathbb{R}^{d_o}$ of the character.

$$x_i^o = F_c(Maxpooling2(h_i^l)) \tag{7}$$

3.1.3. Char2Vec representation

Character-based Chinese NER models usually use Char2-Vec vector trained on large-scale corpus as the initial embedding of characters [26], which contains the context local information of characters. In our model, we also use the pre-trained character embedding lookup table E^c to get Char2Vec vectors, which is trained by Word2Vec model on a massive Chinese corpus Gigaword[1]. We map the character c_i to its Char2Vec representation $x_i^c \in \mathbb{R}^{d_c}$:

$$x_i^c = E^c(c_i) \tag{8}$$

3.1.4. Local representation fusion

After obtaining the three local vectors of a character, it is necessary to fuse them to obtain the comprehensive local representation. Because of the great differences in the three representations, we use the autoencoding mechanism to fuse them. Specifically, we concatenate x_i^s, x_i^o and x_i^c to obtain the initial splicing vector $x_i^d \in \mathbb{R}^{d_s+d_o+d_c}$:

$$x_i^d = Concat[x_i^s; x_i^o; x_i^c] \tag{9}$$

Then, we performs the following two linear transformations and activation functions on x_i^d to obtain the hidden vector $x_i^l \in \mathbb{R}^{d_t}$:

$$x_i^l = Relu(Linear(Tanh(Linear(x_i^d)))) \tag{10}$$

where Linear(.) is a linear transformation, Tanh(.) is Tanh activation function and Relu(.) is Relu activation function. We obtain the reconstructed vector $\widetilde{x_i^d}$ by reconstructing the hidden vector x_i^l:

$$\widetilde{x_i^d} = Linear(Relu(Linear(x_i^l))) \tag{11}$$

We sum up the mean squared error between between x_i^d and $\widetilde{x_i^d}$ of each character in the sentence to get the fusion loss of the sentence.

$$Loss_f = \|\widetilde{x_i^d} - x_i^d\|^2 \tag{12}$$

This loss will be added to the total loss of the model. By this reconstruction process, the middle layer obtains the compressed features to fit the entity recognition through the following sequence annotation task. We take the hidden vector x_i^l of the middle layer as the final local representation of character.

3.2. Global character representation

In this module, we obtain the global representation of characters by utilizing the domain word vectors which commonly contains entity information. The global representation of a character is decided by four situations that the character appears in domain words. We perform the following steps:

Firstly, we use the medical segmentation tool pkuseg[2], an open source segmentation Toolkit, to segment the collected medical corpuses[3], and train word embedding on the segmented corpuses by skipgram model [27]. We set the window size of skipgram to 4, and remove the words less than 5 times. Then we get a medical domain dictionary D and a word embedding lookup table E^d for each domain word.

Next, we make the character c_i as a query to search domain words in dictionary D. If a word w in D contains the character c_i. we assign w to one of the four

[1] https://www.flyai.com/m/gigaword_chn.all.a2b.bi.ite50.vec

[2] https://github.com/lancopku/pkuseg-python

[3] https://github.com/senjinwang/Chinese_medical_NLP

sets $B(c_i)$,$M(c_i)$,$E(c_i)$,$S(c_i)$ according to the different positions of c_i in w, Specifically, according to c_i appears at the beginning, middle, or end of the domain word, the word w is classified into the set $B(c_i)$,$M(c_i)$,$E(c_i)$ respectively, if c_i is same as w (that is, the character is an independent word),the word w is classified into the word set $S(c_i)$.

Then we count the weight q_i^w that represents how important the word w to the character c_i:

$$q_i^w = \frac{m}{M} \tag{13}$$

where m is the frequency of w containing c_i, and M is the frequency of all words containing c_i in the training dataset. We multiply the word vector of each word in the set $B(c_i)$ by the weight q_i^w and add them up to get the beginning global vector $x_i^g(B)$ of the character c_i:

$$x_i^g(B) = \sum_{w \in B(c_i)} q_i^w E^d(w) \tag{14}$$

Similarly, we can obtain the middle global vector $x_i^g(M)$, the end global vector $x_i^g(E)$, and the single global vector $x_i^g(S)$ of c_i. In the end, we combined the four vectors of character c_i to obtain the global representation $x_i^g \in \mathbb{R}^{d_g}$.

$$x_i^g = Concat[x_i^g(B); x_i^g(M); x_i^g(E); x_i^g(S)] \tag{15}$$

3.3. Local and global representation integration

After obtaining the local and global representation of a character, we use gating-mechanism to control their contributions to the comprehensive representation, since one of the representations may have information redundancy compared to the other. The calculation formula is as follows:

$$g_i^l = \sigma(W_l x_i^l + U_l x_i^g + b_l) \tag{16}$$

$$g_i^g = \sigma(W_g x_i^l + U_g x_i^g + b_g) \tag{17}$$

Where, g_i^l and g_i^g is the local and global gating to control the proportion of local and global representation respectively by adaptive learning of the model. W_l, U_l, W_l, U_g, b_l and b_g are learnable parameters. σ denotes the sigmoid function, which is employed to regulate the weight of the gating mechanism within a specific range.

Finally, the local and global representation is multiplied by its corresponding gating and concatenated to be the comprehensive representation $x_i^c \in \mathbb{R}^{d_l+d_g}$.

$$x_i^c = Concat[g_i^l \odot x_i^l; g_i^g \odot x_i^g] \tag{18}$$

3.4. Context representation extraction

Character-based NER is a continuous tokenization task with strong constraint relations between neighboring characters. Therefore, the contextual information of the characters in the sentence sequence should also be considered. We feed the sequence of comprehensive character representations into the Bi-LSTM network to extract the context representation $h_i^c \in \mathbb{R}^{d_h}$:

$$h_i = Concat[\overrightarrow{h_i}; \overleftarrow{h_i}] \tag{19}$$

$$\overrightarrow{h_i} = \overrightarrow{LSTM}(x_i^c) \tag{20}$$

$$\overleftarrow{h_i} = \overleftarrow{LSTM}(x_i^c) \tag{21}$$

Where $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ denote the hidden layer vectors obtained by the forward sequence encoding and the reverse sequence encoding of the sequence, respectively.

3.5. Tags decoding

In the tags output stage, we use CRF as the decoder. CRF will affect the results of the current tag based on the results of the previous tag. Specifically, CRF is composed of an emission matrix and a transition matrix. The emission matrix $M \in \mathbb{R}^{n \times tags}$ records the probability of each tag, and $M_{i,j}$ denotes the probability of the ith word being launched (predicted) to the jth entity tag. n is the number of characters in a sentence, and tags are the number of entity tags. The transition matrix $T \in \mathbb{R}^{tags \times tags}$ is used to simulate the relationship between adjacent tags to be learned in the CRF layer,where $T_{i,j}$ denotes the probability of the jth tag transferring to the ith tag. It is a learnable parameter matrix that can help us to model the transfer relationship between tags explicitly and improve the accuracy of named entity recognition. We use $\mathbf{H}$ to represent the hidden vector matrix of the input sequence, and send it to CRF to find the tag sequence with the largest probability by minimizing the negative maximum likelihood function. The formula is as follows:

$$M = \sigma(W_t \mathbf{H} + \mathbf{b_t}) \tag{22}$$

$$\phi(S, y) = \sum_{i=1}^{n} M_{i,y_i} + \sum_{i=1}^{n-1} T_{y_i, y_{i-1}} \tag{23}$$

$$p(y|S) = \frac{e^{\phi(S,y)}}{\sum_{y' \in Y} e^{\phi(S,y')}} \tag{24}$$

Where $\phi(S, y)$ is the sum of the emission probability between the observation sequence and the tag sequence and the transfer score of the tag sequence, S denotes the observation sequence, y is the real sequence tags. $W_t \in \mathbb{R}^{d_h \times tags}$ and $b_t \in \mathbb{R}^{n \times tags}$ are the parameters of the linear layer, and Y denotes the set of valid tag sequences. We calculate the negative log likelihood function as the label classification loss of the sentence:

$$Loss_{cls} = -\log p(y|S) \tag{25}$$

Finally, we add the classification loss and the fusion loss of characters in the sentence to get the sentence loss, and sum up the loss of all sentences in the training set to get the total loss of the model.

$$Loss_{total} = Loss_{cls} + \lambda Loss_f \tag{26}$$

4. Experiments

We first introduce the datasets, experimental parameters setting and evaluation metrics. Then we present the experimental results of the proposed method and comparison model on the two public datasets, we conduct ablation and parameters experiments on the dataset, Finally, the results and analysis are given. Our model is implemented using PyTorch and requires a minimum 8GB of graphics memory for model training. With a relatively small parameter count of 11.4 million, our model is more compact compared to other existing models. For training, we utilize a GTX 3090 GPU with 24GB of memory on a Linux platform. It takes approximately 1025 seconds to complete one epoch and around 23 hours to finish the entire training process. During the training process, we continuously evaluate the model's F1 score on the validation set after each epoch. The model parameters achieving the highest F1 score on the validation set are saved, and this saved model is further evaluated on the test set. Once the model is trained and saved, it can efficiently extract specific entities from input sentences and provide corresponding labels. It is important to note that if the sentence length exceeds 512 characters, proper segmentation is required to ensure the model's optimal performance.

Table 1
Statistics of datasets

Datasets	statics	Train	Dev	Test
IMCS21	Sentence	54252	6712	6796
	Entity	35033	4398	4309
CMeEE	Sentence	15205	2564	2496
	Entity	58119	9554	9523

4.1. Datasets

To evaluate the performance of our model, we conduct experiments on two medical datasets. One is IMCS21 provided by the Chinese Conference on Computational Linguistics (CCL), which includes more than 60K sentences, and contains five entity types, including symptom, drug name, drug type, examination and operation. Another is CMeEE from the Chinese Biomedical Language Understanding Evaluation, which contains more than 20K sentences. The dataset contains nine categories of medical entities such as common pediatric diseases, body parts, clinical manifestations, medical procedures and so on. The IMCS21 dataset contains a significant number of short sentences with sparse entity distributions, while the CMeEE datasets mostly consist of longer sentences with denser entity distributions, where entities tend to be longer than those in the former dataset. This presents a greater challenge for entity recognition in the latter dataset. These two datasets are publicly available[4], and details are shown in Table 1.

4.2. Evaluation metrics

To evaluate the performance of the model, the evaluation metrics adopted in this paper is the precision rate (P), the recall rate (R) and f1-score (F1):

$$P = \frac{N}{M} \times 100\% \tag{27}$$

$$R = \frac{M}{Z} \times 100\% \tag{28}$$

$$F_1 = \frac{2 \times P \times R}{P + R} \times 100\% \tag{29}$$

Where N is the number of entities correctly predicted by the model. A correctly predicted entity means that both its boundaries and type should be correctly predicted. M is the number of the predicted entity by the model, and Z is the total number of labeled entities in the test data.

[4]https://tianchi.aliyun.com/dataset/95414

Table 2
Experimental parameter settings

Parameter	Value
Spacial representation dimension	128
Sequential representation dimension	128
Char2Vec representation dimension	50
Local representation dimension	200
Global representation dimension	200
Bi-LSTM hidden dimension	400
Learning rate	0.0015
Learning decay rate	0.05
Training batch size	32
Dropout rate	0.5

4.3. Experimental parameter setting

In our model, we adopt Adam optimizer with the learning rate of 0.0015 and the decay step of 0.05. The hyper-parameter λ is set to 0.1 by experimental comparison. Details of parameters settings are shown in Table 2:

4.4. Compared methods

BiLSTM-CRF [28]: The model uses Bi-LSTM to encode character sequences and input hidden vectors to CRF layer for decoding to obtain sequence labels.

Lattice [3]: The model encodes all the characters in the sentence and words recognized by the dictionary, so as to integrate potential word information into the character-based LSTM model.

WC-LSTM [29]: The model is an improved Lattice-based method. The dimension of word embedding is fixed and word embedding is only fused into the last character in the word.

IDCNN [30]: It is iterated dilated convolutional neural networks, which adds a division width to the convolution kernel of CNN to increase the receptive field, so that each convolutional output contains a larger range of information. It can alleviate the disadvantage of CNN with weak feature extraction ability on the input of long sequences.

LR-CNN [19]: The model continuously extracts n-gram information through multi-level CNN, weights different words weights through hierarchical attention, and finally extracts features by fusing attention.

LGN [31]: The model adopts the lexicon-based graph neural network. Each character is taken as a node, and the edge is formed by the matched words. Based on the interaction of multiple graphs between characters, potential words, and the whole sentence, the problem of word ambiguity in NER can be effectively solved.

SoftLexicon [4]: The model improves the way of introducing word information based on Lattice. By integrating lexical information into character representation, it can be transferred to different sequence annotation frameworks, and it is easy to merge with the pretraining model.

FLAT [32]: The model flattens the Lattice structure from a directed acyclic graph to a flat Transformer structure, which takes advantage of long-distance dependencies of Transformer and designs the clever position code to improve the performance of NER.

MECT [26]: The model exploits multiple data embedding, integrates radical, character and word information through cross transformer network, and uses random attention to further improve performance.

5. Result analysis

5.1. Overall results

Experimental results of the different models on CmeEE and IMCS21 are given in Table 3. We can observe that: 1) Our model achieves the best performance on the two datasets. Compared with MECT, which has the best performance among the baselines, the F1 value of our model improves by 1.04% in CMeEE and 0.62% in IMCS21. Our model achieved the highest F1 score on both the text-long and entity-dense CMeEE dataset as well as the text-short and entity-sparse IMCS21 dataset. This indicates that our model has the capability to effectively process texts of varying lengths and densities of entities. Based on this observation, we can speculate that our model exhibits a certain degree of generalization capability. 2) On the whole, the performance of BiLSTM-CRF and WC-LSTM are the lowest. It maybe owing to that BiLSTM-CRF uses only the contextual information of characters, omitting important semantic of words. WC-LSTM has a low performance because its shortest word-first strategy is not suitable for these medical datasets with long entities. CNN-based models such as IDCNN and LR-CNN achieve lower F1 values than MECT and FLAT because they are limited in learning global semantic features. 3) Our model, MECT and Lattice+Glyce outperformed LGN, SoftLexicon, and FLAT on the whole, which shows that the external glyph or radical information is helpful for CMNER. 4) All the models have better results on IMCS21 than CMeEE. While our model can enhance the accuracy of entity boundaries and categories, it struggles to

Table 3
Experimental results of different models on CMeEE and IMCS21

Model	CMeEE			IMCS21		
	P	R	F1	P	R	F1
WC-LSTM	55.63	49.00	52.10	90.97	91.23	91.10
BiLSTM-CRF	58.07	55.01	56.50	90.55	91.44	90.99
IDCNN	58.57	55.81	57.16	90.24	93.39	91.79
LR-CNN	59.91	58.53	59.21	89.99	93.32	91.63
LGN	58.16	58.66	58.41	90.45	93.46	91.93
Lattice	59.34	56.94	58.11	90.40	93.53	91.94
Lattice+Glyce	59.62	60.49	60.05	91.38	92.83	92.10
SoftLexicon	59.96	58.97	59.46	91.09	92.27	91.67
FLAT	56.67	**64.01**	60.12	89.97	93.81	91.85
MECT	59.50	62.49	60.96	90.93	93.41	92.02
Our model	**60.93**	63.15	**62.02**	**91.40**	**93.92**	**92.64**

process complex, lengthy texts and entity-dense medical datasets. In the case of CMeEE datasets with longer sentence and entity-dense, the model identifies multiple continuous entities as a single entity, leading to a lower recall rate and fewer predicted positive entities.

Table 4
Ablation results of the local and global representation

Model	P	R	F1
w/o local	53.89	55.32	53.58
w/o global	59.48	54.63	56.96
w/o spacial&sequential	60.28	62.10	61.17
w/o spacial	61.16	61.44	61.30
w/o sequential	59.95	62.49	61.35
Our model	**60.93**	**63.15**	**62.02**

5.2. Ablation study of local and global representation

We performed the following ablation experiments to verify the effect of the local and global character representation.

w/o local: remove the local character representation from our model.

w/o global: remove the global character representation from our model.

w/o spacial&sequential: remove the spacial and sequential representation in the local character representation.

w/o spacial: only remove the spacial representation in the local character representation.

w/o sequential: only remove the sequential representation in the local character representation.

Experimental results are shown in Table 4. It can be seen that the removal of local or global representation leads to a huge decrease of the F1 value by 8.44% and 5.06%, respectively, which verifies that both local and global representation plays an important role in guaranteeing the performance of our model. What's more, the spacial and sequential ways are helpful to represent the local information of characters completely, and support the model to gain the best performance.

5.3. Experiment analysis of local representation fusion

5.3.1. Comparison of fusion methods

In order to verify the effectiveness of the proposed local fusion method, we conducted experiments on CMeEE with the other local feature fusion methods.

$\mathbf{Fusion_{CAT}}$: concatenate the Char2Vec, sequential and spacial representation directly as the local representation.

$\mathbf{Fusion_{MLP}}$: put the Char2Vec, sequential and spacial representation to three linear layers respectively and add up the output vectors of the linear layers as the final local representation.

The experimental results are shown in Table 5. It can be seen that the autoencoding fusion of our model gets the best result, and the F1 value is 0.51% and 1.67% higher than other methods. $\mathbf{Fusion_{MLP}}$ has the highest R value and lowest P value, which may be because the direct concatenating can use the three local vectors to identify entities comprehensively, but introduces redundant information at the same time to misidentify entities. In contrast, the autoencoding approach can better fuse the three representations, thus fulfilling the accuracy and recall of entity recognition better at the same time.

Table 5
Experiment results of different fusion methods

Model	P	R	F1
$Fusion_{CAT}$	59.80	**63.33**	61.51
$Fusion_{MLP}$	60.82	59.67	60.24
Our model	**60.93**	63.15	**62.02**

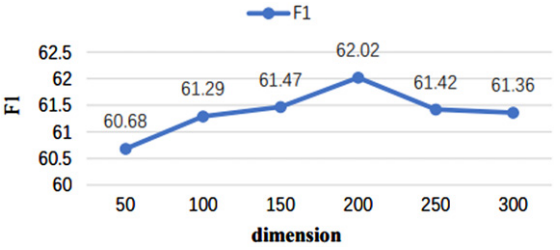

Fig. 3. The effect of fusion vector dimension.

5.3.2. *Effect of hidden vector dimension*

We further explored the effect of the hidden vector dimension in the autoencoder. We perform experiments with the dimension of the hidden vector x_i^l from 50 to 250 on CMeEE. From the results shown in Figure 3 , we can find that the performance of our model is best when the dimension is about 200. The performance decreases if the hidden vector dimension is too low, due to the insufficient representing ability of the hidden vector.

5.4. *Experimental analysis of representation integration*

In order to verify the effectiveness of the local and global representation gating methods on our model, we also conducted experiments on the CMeEE dataset with three other methods.

Integration$_{\mathbf{ADD}}$: add the local and global representations together and then feed it into Bi-LSTM for encoding.

Integration$_{\mathbf{CAT}}$: concatenate the local and global representations directly and feed it into BiLSTM for encoding.

Integration$_{\mathbf{GAT_ADD}}$: multiply the local and global representation by their corresponding gating and add them up, then sent the summation into BiLSTM for encoding.

The experimental results are shown in Table 6. We can see that $Integration_{ADD}$ is not as effective as $Integration_{CAT}$, due to the latter method may save local and global information more completely. Our model uses the gating mechanism to process the local and global representations, and then concatenate them together, which can select the most important information, so as to obtain the optimal performance.

Table 6
Experiment results of different integration methods

Model	P	R	F1
$Integration_{ADD}$	53.89	55.32	53.58
$Integration_{CAT}$	59.31	53.71	56.37
$Integration_{GAT\_ADD}$	60.28	62.10	61.17
Our model	**60.93**	**63.15**	**62.02**

Table 7
Analysis of entity recognition errors

Model	CMeEE			IMCS21		
Error types	BE	EE	TE	BE	EE	TE
LR-CNN	2021	2238	1936	223	218	46
SoftLexicon	2165	2406	1932	212	210	47
FLAT	2039	2384	1887	206	217	51
Our model	**1788**	**2012**	**1864**	**165**	**165**	**39**

5.5. *Error analysis*

Table 7 shows the number of recognizing errors of different models on the two datasets, including the beginning boundary error (BE), the end boundary error (EE), and the entity type error (TE). Compared with the other models, the number of BE, EE and TE of our model are reduced obviously. It shows that the model in this paper significantly improves the performance of medical entity boundary and entity type recognition.

5.6. *Case study*

Figure 4 lists several instances from CMeEE. We can see from the first sentence that LR-CNN and SoftLexicon are unable to recognize the second entity that follows the first, demonstrating the limitations of their recognition capabilities for numerous consecutive entities. For the second sentence, the type of the entity "病毒RNA蛋白复合体" (viral RNA protein complex)" is predicted incorrectly by LR-CNN, which may owe to that the model uses CNN to obtain the context representation, while CNN cannot extract the remote information of characters sufficiently. The boundary of this entity is recognized incorrectly by SoftLexicon. The reason may be that the model introduces too much word information to harm judging the character boundary. Our model can effectively recognize the boundaries and types

Example (Entity tags are marked)	LR-CNN		SoftLexicon		Our Model	
	prediction	Comparison with the real result	prediction	Comparison with the real result	prediction	Comparison with the real result
1. [乙肝疫苗]$_{Dru}$[接种]$_{Pro}$3 剂次. [脊灰疫苗]$_{Dru}$[口服]$_{Pro}$4 剂次。 1. English: Three [doses]$_{Pro}$ of [hepatitis B vaccine]$_{Dru}$ and four [doses]$_{Pro}$ of [oral polio vaccine]$_{Dru}$.	[乙肝疫苗]$_{Dru}$ [脊灰疫苗]$_{Dru}$	**Lack of :** [接种]$_{Pro}$ [口服]$_{Pro}$	[乙肝疫苗]$_{Dru}$ [脊灰疫苗]$_{Dru}$	**Lack of :** [接种]$_{Pro}$ [口服]$_{Pro}$	[乙肝疫苗]$_{Dru}$ [接种]$_{Pro}$ [脊灰疫苗]$_{Dru}$ [口服]$_{Pro}$	Correct
2. 与 [病毒RNA蛋白复合体]$_{Mic}$ 结合后可抑制[病毒]$_{Mic}$转录。 2. English: It can inhibit [viral]$_{Mic}$ transcription after binding to [viral RNA protein complex]$_{Mic}$	[病毒 RNA 蛋白复合体]$_{Bod}$ [病毒]$_{Mic}$	**Type error:** *Bod*	[病毒 RNA 蛋白]$_{Mic}$ [病毒]$_{Mic}$	**Boundary error** complex is the end of entity	[病毒 RNA 蛋白复合体]$_{Mic}$ [病毒]$_{Mic}$	Correct

Fig. 4. Case analysis.

of medical entities owing to its reasonable utilization of the local character information and the global word information. Furthermore, in this particular case, the longer entity includes the two characters present in the shorter entity. If both entities have been labeled correctly in the training corpus, some high-performing models can accurately recognize them. However, if a model is trained solely on the entity "病毒(virus)," the characters "病毒(virus)" in the longer entity "病毒RNA蛋白复合体(viral RNA protein complex)" may be erroneously identified as a separate entity.

6. Conclusion

In this paper, we propose a novel character representation enhanced model to improve the performance of NER in medical Chinese text. According to the observation that both certain radicals inside the character and certain domain words containing the character imply medical entity meaning, we propose to combine these local and global representations by autoencoding and gating mechanism, therefore to improve the medical entity-aware ability of characters. The proposed model demonstrates satisfactory results for medical named entity recognition, as confirmed by the ablation study, which highlights the usefulness and rationalization of the proposed modules. However, real industrial settings often face challenges such as high labeling costs and insufficient generalization. To address this, hybrid systems combining the rules-based model with our model are typically used. We'll investigate this further in the future.

Acknowledgement

This work is supported by Yunnan provincial major science and technology special plan projects (Grant NO.202202AD080004, NO.202202AE090008).

References

[1] C. Chen and F. Kong, Enhancing entity boundary detection for better chinese named entity recognition, in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (Volume **2**: Short Papers), pp. 20–25, 2021.

[2] Q. Zhang, M. Wu, P. Lv, M. Zhang and H. Yang, Research on named entity recognition of chinese electronic medical records based on multi-head attention mechanism and character-word information fusion, *Journal of Intelligent & Fuzzy Systems*, no. Preprint, pp. 1–12, 2022.

[3] Y. Zhang and J. Yang, Chinese ner using lattice lstm, in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (Volume **1**: Long Papers), pp. 1554–1564, 2018.

[4] R. Ma, M. Peng, Q. Zhang, Z. Wei and X.-J. Huang, Simplify the usage of lexicon in chinese ner, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5951–5960, 2020.

[5] W. Liu, X. Fu, Y. Zhang and W. Xiao, Lexicon enhanced chinese sequence labeling using bert adapter, in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (Volume **1**: Long Papers), pp. 5847–5858, 2021.

[6] J. Lafferty, A. McCallum and F.C. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, 2001.
[7] A. McCallum and W. Li, Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons, 2003.
[8] B. Settles, Biomedical named entity recognition using conditional random fields and rich feature sets, in *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications* (NLPBA/BioNLP), pp. 107–110, 2004.
[9] Z. Ju, J. Wang and F. Zhu, Named entity recognition from biomedical text using svm, in *2011 5th international conference on bioinformatics and biomedical engineering*, pp. 1–4, IEEE, 2011.
[10] B. Tang, H. Cao, Y. Wu, M. Jiang and H. Xu, Recognizing clinical entities in hospital discharge summaries using structural support vector machines with word representation features, in *BMC Medical Informatics and Decision Making* **13** (2013), 1–10, BioMed Central.
[11] K. Liu, Q. Hu, J. Liu and C. Xing, Named entity recognition in chinese electronic medical records based on crf, in *2017 14th Web Information Systems and Applications Conference (WISA)*, pp. 105–110, IEEE, 2017.
[12] B.A. Goldstein, A.M. Navar, M.J. Pencina and J. Ioannidis, Opportunities and challenges in developing risk prediction models with electronic health records data: a systematic review, *Journal of the American Medical Informatics Association* **24**(1) (2017), 198–208.
[13] L. Yao, H. Liu, Y. Liu, X. Li and M.W. Anwar, Biomedical named entity recognition based on deep neutral network, *Int J Hybrid Inf Technol* **8**(8) (2015), 279–288.
[14] L. Li, L. Jin, Y. Jiang and D. Huang, Recognizing biomedical named entities based on the sentence vector/twin word embeddings conditioned bidirectional lstm, in *Chinese computational linguistics and natural language processing based on naturally annotated big data*, pp. 165–176, Springer, 2016.
[15] D. Zeng, C. Sun, L. Lin and B. Liu, Lstm-crf for drug-named entity recognition, *Entropy* **19**(6) (2017), 283.
[16] L. Luo, Z. Yang, P. Yang, Y. Zhang, L. Wang, H. Lin and J. Wang, An attention-based bilstm-crf approach to documentlevel chemical named entity recognition, *Bioinformatics* **34**(8) (2018), 1381–1388.
[17] Z. Liu, C. Zhu and T. Zhao, Chinese named entity recognition with a sequence labeling approach: based on characters, or based on words? in *International Conference on Intelligent Computing*, pp. 634–640, Springer, 2010.
[18] H. Li, M. Hagiwara, Q. Li and H. Ji, Comparison of the impact of word segmentation on name tagging for chinese and japanese, in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pp. 2532–2536, 2014.
[19] T. Gui, R. Ma, Q. Zhang, L. Zhao, Y.-G. Jiang and X. Huang, Cnn-based chinese ner with lexicon rethinking., in *ijcai*, pp. 4982–4988, 2019.
[20] H. Yan, B. Deng, X. Li and X. Qiu, Tener: adapting transformer encoder for named entity recognition, *arXiv preprint arXiv:1911.04474*, 2019.
[21] J. Shi, M. Sun, Z. Sun, M. Li, Y. Gu and W. Zhang, Multilevel semantic fusion network for chinese medical named entity recognition, *Journal of Biomedical Informatics* **133** (2022), 104144.
[22] B. Lyu, L. Chen and K. Yu, Glyph enhanced chinese character pre-training for lexical sememe prediction, in *Findings of the Association for Computational Linguistics: EMNLP*, 2021, pp. 4549–4555, 2021.
[23] Q. He, L. Wu, Y. Yin and H. Cai, Knowledge-graph augmented word representations for named entity recognition, in *Proceedings of the AAAI Conference on Artificial Intelligence* **34** (2020), 7919–7926.
[24] 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* **98** (2019), 103289.
[25] Y. Meng, W. Wu, F. Wang, X. Li, P. Nie, F. Yin, M. Li, Q. Han, X. Sun and J. Li, Glyce: Glyph-vectors for chinese character representations, *Advances in Neural Information Processing Systems* **32** (2019).
[26] S. Wu, X. Song and Z. Feng, Mect: Multi-metadata embedding based cross-transformer for chinese named entity recognition, *arXiv preprint arXiv:2107.05418*, 2021.
[27] T. Mikolov, K. Chen, G. Corrado and J. Dean, Efficient estimation of word representations in vector space, *arXiv preprint arXiv:1301.3781*, 2013.
[28] Z. Huang, W. Xu and K. Yu, Bidirectional lstm-crf models for sequence tagging, *arXiv preprint arXiv:1508.01991*, 2015.
[29] W. Liu, T. Xu, Q. Xu, J. Song and Y. Zu, An encoding strategy based word-character lstm for chinese ner, in *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), pp. 2379–2389, 2019.
[30] E. Strubell, P. Verga, D. Belanger and A. McCallum, Fast and accurate entity recognition with iterated dilated convolutions, *arXiv preprint arXiv:1702.02098*, 2017.
[31] T. Gui, Y. Zou, Q. Zhang, M. Peng, J. Fu, Z. Wei and X.-J. Huang, A lexicon-based graph neural network for chinese ner, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1040–1050, 2019.
[32] L. Xiaonan, Y. Hang and Q. Xipeng, Flat: Chinese ner using flat-lattice transformer [c], in *Association for Computational Linguistics*, 2020.