Leveraging Lexical Features for Chinese Named Entity Recognition via Static and Dynamic Weighting

Dong Zhang, Chengying Chi, Xuegang Zhan

***Abstract*—Named entity recognition (NER) is an essential research assignment in natural language processing. Entity boundaries and word segmentation information are crucial determinants that affect the accuracy of Chinese named entity recognition. Researchers generally utilize two methods: word-based method and character-based method. Word-based methods rely heavily on segmentation tools. These methods can cause error propagation problems from word segmentation incorrectly. Character-based methods lack word-level information and word boundaries information, which will cause the problem of missing information. A novel neural entity recognition model fused with self-matched lexical word features is proposed for Chinese NER tasks in response to these problems. In the embedding layer, they were using the attention mechanism to extract the semantic features of word sequences, using the statistical character frequency to extract the syntactic features. Our proposed model is validated in many datasets. The F1 score of 73.9% and 57.65% was obtained on the OntoNotes 4 and Weibo datasets. The F1 score of 90.76% and 95.58% was obtained on the MSRA and Resume datasets, and the state-of-the-art experimental results are obtained on the Resume benchmark dataset.**

***Index Terms*—named entity recognition, sequence tagging, trie tree, attention mechanism**

I. Introduction

NAMED Entity Recognition (NER) is a sub-task of information extraction, which plays a significant role in Natural Language Processing (NLP). It is also the basis of tasks such as machine translation, text classification, information retrieval[1], emotion recognition [2] knowledge reasoning, and entity linking. The entity category includes names of people, places, and institutions. This task consists of two parts: detection of entity boundary and label classification. Detection of entity boundary is essential in NER. For Chinese words without delimiters, the entity boundary ambiguity tends to worse effect than expected. Label classification is usually to classify the entity that recognizes the boundary. Investigating NER as a classification or sequence labeling problem is a continuing concern within NLP.

The model is mainly divided into four types according to the model's decoding process. The most commonly adopted method is to use the Softmax layer [3] or Conditional Random Field (CRF) [4], [5], [6] to decode the best score label sequence. Softmax function to select the optimal result for a given label sequence through a fast greedy selection strategy, output scores are normalized into a probability distribution. The second approach is to predict the entity boundary in each sentence through PointerNet, at the same time, in order to obtain the probability of different entities according to the positional information [7]. Another way is to enumerate the sub-segments of text and then classify the labels [8]. This process can identify the nested entities in the sentence, but when the sentence is long, many negative samples can lead to higher processing costs. The last decoding method is to encode the tokens information by the Seq2Seq model to predict each token [9]. This method utilized an end-to-end calculation process to partially solved the problem of nested entities.

There are two standard approaches to previous works for Chinese NER: word-based method and character-based method. The word-based way performs word segmentation first, and then uses the result after segmentation as the input of the neural network. Word-based methods are the NER pipeline, and the NER task's performance relies heavily on the accuracy of segmentation since name entities are the critical source of Out of Vocabulary (OOV) in segmentation. The incorrectly segmented entity boundaries lead to task errors, and the pipeline task can suffer the potential issue of error propagation. As shown in Fig. 1, we construct the self-match word sets by matching the word sequences "南京市长江大桥(Nanjing Yangtze River Bridge)". Two different results for "南京市/长江大桥(Nanjing City/Yangtze River Bridge)" and "南京/市长/江大桥(Nanjing/major/Daqiao Jiang)" produced different meanings of the sentence. Character-based methods avoid using word segmentation in the input layer, which can effectively prevent the error propagation problem caused by Chinese word segmentation errors. However, character-based methods have two disadvantages. First, there is no word-level information in the embedding layer, which will cause a lack of latent word information in the model. The information contained in Chinese words and characters is different, and word-level information can also provide adequate prior knowledge for the model. Second, the sentence's word boundaries are ambiguous, and the model needs to be obtained through network learning during the training process.

Chinese NER is a challenging problem because there are no delimiting white spaces between words. This paper proposed a neural entity recognition approach to leverage lexical features using self-matched words and the nearest contextual

Manuscript received August 25, 2020; revised January 7, 2021. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant No.61672138.

D. Zhang is a master candidate in the School of Computer Science and Software Engineering at the University of Science and Technology LiaoNing, Anshan, China (e-mail: zhang_dong126@126.com).

C. Chi, the corresponding author, is a Professor in the School of Computer Science and Software Engineering at the University of Science and Technology LiaoNing, Anshan, China (e-mail: chichengying@ustl.edu.cn).

X. Zhan is an Associate Professor in the School of Computer Science and Software Engineering at the University of Science and Technology LiaoNing, Anshan, China (e-mail: zhanxg@ustl.edu.cn).

Fig. 1. Example of word segmentation.

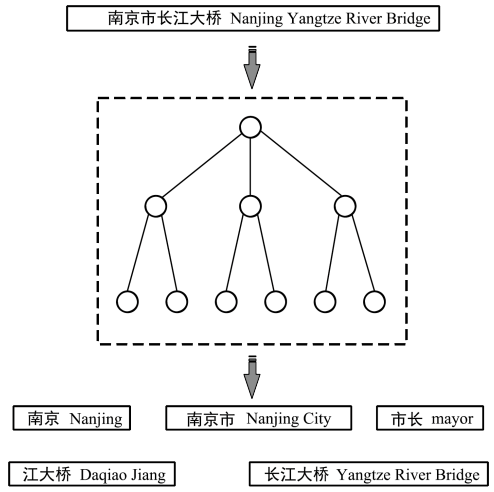

Fig. 2. The process of matching self-matched word.

lexical words. Based on the character-based methods, we also leveraged external data by word embedding lexicon information. The self-matched words of a character are the lexical word that contains this character. We used a trie (a letter-tree) to integrate self-matched lexical words. Our pre-trained embedding vector is similar to what is described in Sui [9]. The process of matching the self-matched word is shown in Fig. 2 Compared with character-based methods, our model explicitly leveraged prior knowledge. Compared with the word-based methods, our model alleviated the error propagation problem caused by wrong word segmentation. At the same time, our model improved the data at the input layer, has a more lightweight representation, and can easily transplant various models.

The main contributions of this work are summarized as follows:

1) We used the dynamic and static weighting methods to extract the self-matched lexicon words information, and they merged the lexicon information with the character features. According to the number of self-matched lexicon words, statistical methods are used to statically extract the syntactic features of self-matched words in different word sequences. According to the self-matched words embedding matrix, the attention mechanism is used to dynamically extract the semantic features of matching words in different sentences.

2) Our model's improvement at the word embedding layer, using both character-level features and word-level features, can be swiftly and effortlessly ported to different models.

3) Our proposed model obtained the state-of-the-art results on the Resume benchmark dataset and excellent result on the MSRA benchmark dataset. Our source code is published at https://github.com/zhangdddong/ChineseNER_DS.

II. RELATED WORK

Typically, NER is conducted as a sequence labeling task. Traditional machine learning methods mainly use Hidden Markov Model (HMM) [10] and Conditional Random Field (CRF) [4]. With the progress in deep learning for NLP, many researchers use a deep neural network to finish NER tasks. Liu *et al.* [11] pointed out that the neural network is far more effective and used the high dimensional word vectors as the input to the LSTM-CRF model, and also used a Character-aware Neural Language Model to extract character information in the word sequences. Using the character information, this model obtained the best result of F1 score 91.71 percent on the CoNLL2003 English dataset. Žukov-Gregorič *et al.* [12] employed multiple independent bidirectional LSTM units and proposed Parallel Recurrent Neural Networks. The novel model had fewer parameters and achieved excellent results on the CoNLL2003 English dataset. The CRF is usually used as the last layer of the neural network to learn the label sequence's potential relationship. Although it can improve the accuracy of model prediction, it dramatically affects the running speed. Cui *et al.* [13] used the NCRF++ Neural Sequence Labeling Toolkit [14] to extract character-level features, and proposed a Label Attention Network to replace the CRF. The best results had been achieved in the multi-sequence labeling task. Compared with CRF, this model can improve decoding significantly. The supervised learning method requires many label data, as well as the potentially high cost of annotating data for those methods. Clark *et al.* [15] used a training dataset mixed with a labeled dataset and unlabeled dataset to improve the sentence decoding of bidirectional LSTM. Furthermore, he proposed a semi-supervised learning model based on Cross-View Training. At the same time, some researchers had applied Reinforcement Learning to NER's field and achieved excellent results [16].

The task of Chinses NER is difficult because there are no natural spaces between Chinese sentences. When solving the problem of Chinese NER, Chinese Word Segmentation (CWS) becomes more and more critical on performance ([17], [18]). Cao *et al.* [19] proposed a novel Transfer Learning framework by making full use of word boundaries information when solving Chinese NER's problem. In order to allow the neural network to learn more characteristics of the entity's boundaries, he used a self-attention mechanism to capture the dependencies between adjacent label sequences, and this architecture achieved the best result on the WeiboNER and SighanNER dataset. Wu *et al.* [20] used the CNN-LSTM-CRF model for Chinese NER and used data enhancement techniques that automatically generate pseudo-tags to train a shared neural network that combines word segmentation and entity recognition. Rationalist commonly uses gazetteers, but using this method directly has limitations in the NER tasks when the recognition field expands. Ding *et al.* [21] used Graph Neural Networks to capture lexical words features in named entity recognition system automatically and reduced the ambiguity of the language of fusing lexical words information. Graph neural networks are also widely used in Chinese NER. Gui *et al.* [22] attached character features, word potential relationships, and sentence semantic to Graph Neural Networks to obtain the best results in four Chinese datasets. Combined with the character-based model and the word-based model, Zhang *et al.* [23] decoded input character sequence and lexical words composed of all characters, and he proposed the Lattice-LSTM model. Experiments

showed that entirely using global semantic information can significantly improve model performance. Sui *et al.* [27] expressed the semantic information in the Collaborative Graph Network, which solved the Lattice-LSTM model's shortcomings, and this model made processing speed six times faster. Peng *et al.* [28] proposed a more lightweight model and alleviated information loss caused by fuzzy input word sequences.

The attention mechanism was initially being applied in the field of image classification [29]. As some researchers applied it to machine translation [30], [31], some began to apply attention mechanisms to the NER task. Mengge *et al.* [32] combined Transformer Structure and dictionary information, and used positional encoding representations and lexical information to express the semantic features of word sequences by the Lattice Structure. Mengge investigated the lattice-aware self-attention coupled with position information to explore sufficient word information in the lattice structure. Li *et al.* [33] proposed a FLAT model by changing the position code, which converts the lattice structure into a flat structure consisting of spans. Dong *et al.* [34] applied the self-attention mechanism on the span-level semantic representation to improve NER performance and achieved the state-of-the-art results. Jin *et al.* [35] proposed a novel character-level Gated Convolutional Recurrent neural network with attention called GCRA for the Chinese NER task. This model achieved excellent results. By combining the attention mechanism and the dictionary information, the dynamic lexical information can be quickly and accurately used, so that the model can obtain better prior information in the embedding layer. Our model tried a novel structure based on the combination of the attention mechanism and lexical information and added the syntactic information could more fully express the semantic representation of the input word sequences.

III. MODELS

We used bidirectional LSTM as the basic structure for Chinese NER. We were using dynamic weighting and static weighting to extract the features of self-matched lexical words fully. Then, the self-matched lexical word features and character features are fused at the word embedding layer. In this chapter, we introduced the specific detail of the model that fused the matching word features. The overall structure of the model of fused self-matched lexical word features is divided into three layers: Fusion Layer, Network Layer, and Decoding Layer. The primary function of the Fusion Layer is feature fusion. First, the characters are mapped into word vectors of limited dimensions through a neural language model. Then, the self-matched lexical word sets of the input word sequences are calculated based on the characters' representation in different sentences. Finally, through dynamic weighting and static weighting to extract self-matched word sets features, and use the information after features fusion as the output of the Fusion Layer. The Network Layer mainly contains the neural network's arithmetic unit, which is the necessary part of model components. The primary function of the Decoding Layer is to decode the information of output by the neural network. Our model structure is shown in Fig. 3.

A. Formulation

In the Chinese NER task, we describe a sentence $X = \{c_1, c_2, \ldots, c_n\}$, where c_i is the ith character in the sentence, and n represents the sentence's length. Correspondingly, we indicated the result's label sequences are $Y = \{y_1, y_2, \ldots, y_n\}$ where y_i represents the label corresponding to the ith character. The self-matched lexical word sets for each sentence are $W = \{w_1, w_2, \ldots, w_m\}$ where w_j is the jth matching word in the self-matched word sets, and m is the number of matching words in the sentence. If there is no matching word for each character in the sentence, the self-matched word sets can be empty. We used $f^d(\bullet)$ to denote the features function that extracted the matching word features through dynamic weighting and used $f^s(\bullet)$ to denote the features function that extracts the matching word features through static weighting. Then the objective function can be defined as $f_\theta : [X; f^d(W); f^s(W)] \Rightarrow Y$. The rest of this chapter, on this basis, developed the model derivation process more carefully.

B. Fusion Layer

In this part, we introduced the Fusion Layer of the model. The Fusion Layer's main idea is to obtain adequate prior knowledge by extracting more features. Each character in the input character sequences $X = \{c_1, c_2, \ldots, c_n\}$ used the essential vector feature calculated by the embedding lookup table as $c_i^{emb} = e^c(c_i)$, where e^c is the character embedding lookup table. Furthermore c_i^{emb} is the vector representation corresponding to the ith character and $c_i^{emb} \in R^{charD}$. $charD$ is the dimension of the character vector. The sequence of matching words corresponding to the sentence $W = \{w_1, w_2, \ldots, w_m\}$ used the basic vector features calculated by the embedding lookup table as $w_j^{emb} = e^w(w_j)$, where e^w is the self-matched word embedding lookup table. Furthermore, w_j^{emb} is the vector representation corresponding to the jth matching word in a sentence, and $w_j^{emb} \in R^{wordD}$, $wordD$ is the dimension of the matching word vector. To better merge the matching word features, we used dynamic weighting and static weighting to extract semantic features and syntactic features, respectively.

The dynamic weighting method uses the attention mechanism to integrate the information between the input characters and the self-matched lexical words. The attention mechanism can effectively extract the semantic information of the matched words dynamically. The embedding set on self-matched words is $W^{emb} = \{w_1^{emb}, w_2^{emb}, \ldots, w_m^{emb}\}$, where m is matching words in the word sequences. The attention weight score is:

$$Att_k = \sum_{j=1}^{m} a_j w_j^{emb} \tag{1}$$

where Att_k represents the self-matched word attention vector corresponding to the kth sentence. This method extracted sentence-level matching word features by calculating the matching word's weight in the word sequences. Previously, we planned to dynamically increase the weight of matching word sets for each character, but considering efficiency issues, we chose this solution. The a_j represents the attention weight vector obtained by the first self-matched

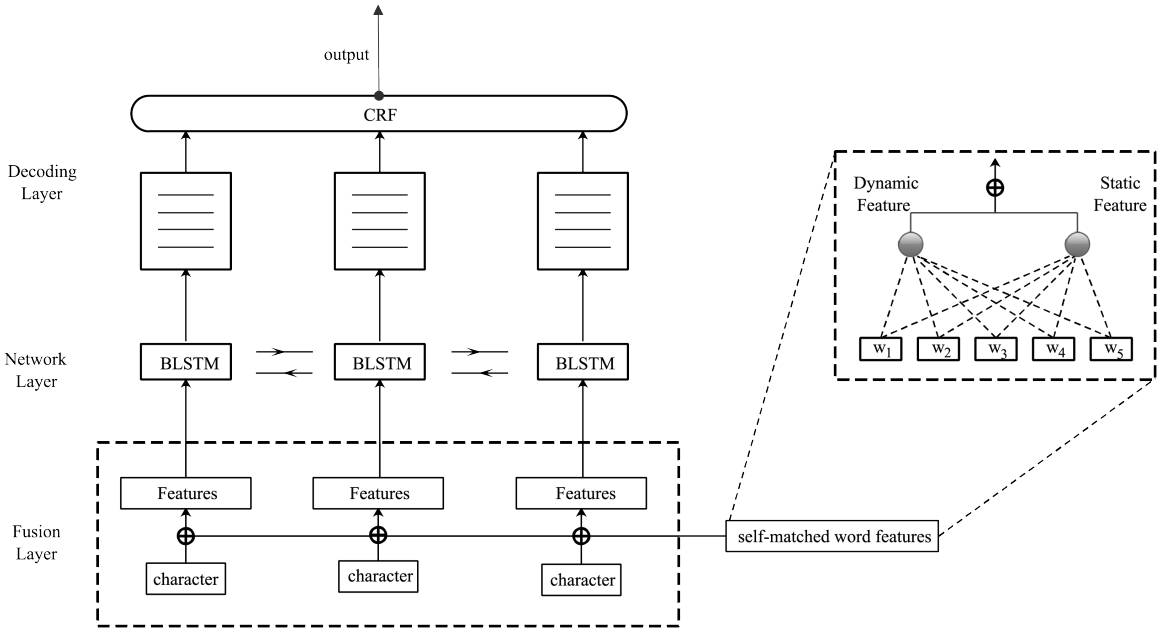

Fig. 3. Model structure.

words through the attention mechanism. The representation for a_j is:

$$a_j = \tanh(a_j{}') \tag{2}$$

$$a_j{}' = softmax(score(q, w_j^{emb})) \tag{3}$$

where $score(q, \mathrm{w}_j^{emb})$ is the attention score of the jth self-matched word, the output result represents the scores of different self-matched words in the actual sentence, q is a weight tensor whose dimension is the same as the dimension of the word vector of the self-matched word, and it will follow the training of the model is continuously update and iterative. The function for $score(q, e_j^{emb})$ is:

$$score(q, w_j^{emb}) = \tanh(w_j^{emb}) \odot q \tag{4}$$

where $\odot$ represents tensor contraction, and the weight tensor q is used to reduce the dimension of the word vector matrix.

The static weighting method used statistical methods to count the quantitative information between the input characters and the self-matched words. And then combine the static word vectors to extract the characteristic information of the self-matched words effectively. This method's process speed is faster than the dynamic weighting method, and we think this method can extract the surface features of self-matched words in word sequences. Form the matching words of sentences to the quantified sets $\{w_1^{emb}, w_2^{emb}, \ldots, w_m^{emb}\}$, the static features of matching is calculated by:

$$STA_k = \frac{1}{|W|} \sum_{w^{emb} \in W} w^{emb} \tag{5}$$

where STA_k represents the static extraction vector corresponding to the kth sentence, $|W|$ is the number of matching words in the self-matched word sets. w^{emb} is the word vector corresponding to the self-matched words. This statistical method believes that in the sets of self-matched words, the more critical in these sets if the same matching word appears more often. This method can quickly calculate the static features in the combination of self-matched words.

Then, perform a matrix connection operation on the character word vector, the dynamic weighted features of the matched word, and the static weighted features to obtain the fusion layer's output. Through the matrix connection operation, it can ensure that the extracted features are not lost. Finally, take the Fusion Layer's output as the input of the Network Layer, and use I_{fusion} to represent the output of the Fusion Layer. The value of I_{fusion} is calculated by :

$$I_{fusion} = [W^{emb}; Att; STA] \tag{6}$$

C. Network Layer

The output after the Fusion Layer can be used as the input of the Network Layer. Most NER models use bidirectional LSTM as the Network Layer of the model, so the Network Layer used in the experiments is mainly bidirectional LSTM. The final representation for I_{fusion} is:

$$I_{fusion} = \{x_1, x_2, \ldots, x_n\} \tag{7}$$

where x_i represents the ith fused features vector, each character corresponds to a high-dimensional feature vector, and n represents the sentence length. The fused features I_{fusion} is used as the input of the bidirectional LSTM network. The bidirectional LSTM captured global sequence information from right to left and from left to right. The Bidirectional LSTM hidden layer's output sequence can be expressed by vector $(h_1, h_2, \ldots, h_n)$ and nearly one-to-one correspondence between input vectors and output sequence. $(h_1, h_2, \ldots, h_n)$ can be divided into two partitions: $(\overrightarrow{h_1}, \overrightarrow{h_1}, \ldots, \overrightarrow{h_n})$ and $(\overleftarrow{h_1}, \overleftarrow{h_1}, \ldots, \overleftarrow{h_n})$ in the left-to-right and right-to-left directions. Therefore, the hidden layer of the bidirectional LSTM function is:

$$h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}] \tag{8}$$

D. Decoding Layer

NER is a typical sequence labeling task, which identifies entities by marking entity boundaries and entity categories.

TABLE I
STATISTICS OF DATASETS

Dataset	Train	Dev	Test	Entity Type
MSRA	13619k	-	1085k	3
Resume	1069k	118k	132k	7
OntoNotes 4	492k	203k	210k	4
Weibo	201k	100k	102k	4

A CRF decoding follows the output result $(h_1, h_2, \ldots, h_n)$ of the neural network. The CRF decoding probability corresponding to each real tag $Y = \{y_1, y_2, \ldots, y_n\}$ is:

$$p(y|s) = \frac{\exp(\sum_i (W^i_{CRF} h_i + b^{(i-1,i)}_{CRF}))}{\sum_{y'} \exp(\sum_i (W^i_{CRF} h_i + b^{(i-1,i)}_{CRF}))} \quad (9)$$

where y represents the sequence label, and y' represents any label in the label sequence. W^i_{CRF} corresponds to the ith parameter, $b^{(i-1,i)}_{CRF}$ represents the paranoia between $i-1$ and i. We used the standard Vertibi algorithm from decoding. The loss function is:

$$loss = \sum_{i=1}^{n} \log(p(y_i|s_i)) \quad (10)$$

IV. EXPERIMENT

A. experiment settings

1) Datasets: For the experiment more convincing, the datasets we used include the MSRA dataset and the Resume dataset. Statistics of the datasets are shown in Table I.

- **MSRA** [36] is a standard news field dataset, provided by Microsoft Research Asia. The dataset only contains the training set and the testing set. The schema method of the dataset uses the BIO annotation method. It mainly contains three types of entities: persons, places, and institutions.
- **Resume** [23] is a financial resume dataset, the resumes of 1027 financial practitioners from Sina Finance. The seven types of entities included in the dataset are cities, educational institutions, place names, personal names, organization names, proper nouns, professional backgrounds, and job titles. The training set contains 13,438 entities, the validation set contains 1497 entities, and the testing set contains 1630 entities.
- **OntoNotes 4** [24] comes from Newswire, Broadcast News, Broadcast Conversation, Web text, and the primary entities used in the experiment: LOC, PER, ORG, and GPE. We also refer to this paper [25] for more information about the data split.
- **Weibo** dataset [26]: The samples consist of 1890 Weibo messages, including four entity types: Location, Person, Organization, and Geo-political entity.

2) Hyper-Parameters: Table II shows the hyper-parameters of our model. The character embedding dimension size is set to 50, the self-matched lexical word vector embedding is set to 300, and the word2vec is trained on Chinses Giga-word of the corpus. The hidden size of bidirectional LSTM models is set to 200. It is worth noting that on the Resume dataset, the optimizer we use is SGD, and the learning rate is set to 0.015. On the MSRA dataset, we set the learning rate to 0.001, and the optimizer used is Adam. To overcome overfitting, we set the gradient clipping to 5.0. We also used the dropout method to improve the generalization ability of neural network learning features.

TABLE II
HYPER-PARAMETER VALUES

Parameters	Value
char embedding	50
lexicon embedding	100
dropout	0.5
gradient clip	5.0
bilstm-layers	1
bilstm hidden units	200
learning rate	0.001

3) Evaluation Metrics: The evaluation metrics of the experiments use precision, recall, and F1 score. In the Chinese NER task, the positive cases are entities, and the negative cases are non-entities. The precision is the proportion of correct words in all predicted entity words; recall measures the model's generalization ability. For the precision and the recall are inversely proportional, and the F1 score is the harmonic average of them. The metrics functions are:

$$precision = \frac{TP}{TP + FP} \quad (11)$$

$$recall = \frac{TP}{TP + FN} \quad (12)$$

$$F1 = 2 \times \frac{precision * recall}{precision + recall} \quad (13)$$

where the TP is the true positive, the FP is the false positive, the TN is the true negative, and the FN is the false negative.

B. Experiment Results

Using the MSRA dataset and Resume dataset to verify the effectiveness of our model. At the same time, to observe the impact of the model's dynamic weighting and static weighing on the experimental effects, we conducted experiments separately. In order to express the experimental results more clearly, we introduce the model's name as follows:

- **Character + Static-features:** It represents a model that combines character features and static features and uses statistical matching word frequency to extract their static features. Bidirectional LSTM is used in the neural network layer, and the input layer uses only character-based representation vectors.
- **Character + Dynamic-features:** It represents the fusion model of character features and dynamic features and uses the attention mechanism to extract the dynamic features of the matching words of the model.
- **DS-attention:** Our proposed model combines the features extracted by static weighting and dynamic weighting, then fusion with character vector features. The method of feature fusion is concatenating operation.

1) MSRA: Due to the lack of a validation set, we use the testing set to display the experimental results. Results on MSRA dataset are shown in Table III.

Chen *et al.* 2006 used CRF and maximum entropy models for Chinese NER and finally achieved an F1 score of 86.20%.

TABLE III
RESULTS ON MSRA

Models	Precision	Recall	F1
Chen *et al.* 2006 [37]	91.22	81.71	86.20
Zhou *et al.* 2006 [38]	88.94	84.20	86.51
Zhou *et al.* 2013 [39]	**91.86**	88.75	90.28
Dong *et al.* 2016 [40]	91.28	90.62	**90.95**
Character + Dynamic-features	89.26	89.41	89.34
Character + Static-features	89.67	89.55	89.61
DS-attention	**90.12**	**91.42**	**90.76**

Zhou *et al.* 2006 used a character-level CRF to segment the sentences, and the results of the segmentation contained a small number of simple entities. Then, using three word-level CRF models, the Person, Places, and Organization in the results are identified. Finally, they use a conventional method to mark entities that are not recognized by the CRF. This method that was combining rules with statistical methods achieved 86.51% of the F1 score. The traditional Chinese NER model cannot use global character features. Zhou *et al.* 2013 made full use of a string of continuous characters globally, and this method identified and categorized Chinese NER via a global linear model and gave 90.28% on F1 score. Their model also used 10 features for Chinese and 31 context features to ensure the model's performance. Dong *et al.* 2016 utilized a bidirectional LSTM-CRF structure that integrates both character-level and radical-level representations. Their model represented the Chinese radical-level information in BLSTM-CRF architecture without using standard features and achieved 90.59% on the F1 score. Our model based on the combination of dynamic weighting and static weighting is 4.25% higher than the best model using CRF. Compared with Zhou *et al.* 2013, our model has increased the F1 score by 0.48% without using any artificial features and contextual features. Compared with Dong *et al.* 2016, the construction process of radical-level features is more complicated. Our model integrates the input information without construct radical-level features or utilizes a lot of external resources.

Table III shows the result comparing Character + Dynamic-features, Character + Static-features, and our model. Compared with the method that only uses dynamic weighting, the DS-attention model's F1 score has increased from 89.34% to 90.76%, and the performance is improved by 1.42%. This experimental phenomenon shows that the features of the self-matched words can be well extracted through the attention mechanism and merged with the character information. Compared with the static model, the DS-attention model's F1 score has increased from 89.61% to 90.76%, an increase of 1.15%. The statically weighted word frequency information also has an impact on the performance of the model. The method of using static weighting is 0.27% more effective than using dynamic weighting. This phenomenon shows that the word frequency information extracted by static weighting will learn more knowledge on the MSRA dataset. Since the DS-attention model integrates dynamic semantic information and static word frequency information, the model has a higher recall. Results of different entities on the MSRA dataset are shown in Table IV. In the table, the three models have lower evaluation indicators for the Organization. The F1 score of dynamic weighting and static weighting reached 79.15% and 79.79%, respectively. Moreover, the F1 score of the DS-attention model after feature fusion increased to 82.83%. At the same time, in the Person entity, compared with the static model, the F1 score of the DS-attention model increased by 0.81% and compared with the dynamic model, the F1 score of the DS-attention model increased by 1.02%. In place name entities, the F1 score of the DS-attention model increased by 0.51% compared with the static model, and the F1 score of the DS-attention model increased by 1.41% compared with the dynamic model. Experimental results show that the model after feature fusion has a significant performance improvement for different entities such as person names, place names, and organization names.

TABLE IV
DIFFERENT ENTITY RESULTS ON MSRA

Models	Type	Precision	Recall	F1
Character + Dynamic-features	P	94.07	92.39	93.22
Character + Dynamic-features	L	93.25	92.76	93.00
Character + Dynamic-features	O	77.94	80.40	79.15
Character + Static-features	P	94.50	92.39	93.43
Character + Static-features	L	91.90	93.93	92.90
Character + Static-features	O	80.99	78.63	79.79
DS-attention	P	94.45	94.03	94.24
DS-attention	L	92.96	93.87	93.41
DS-attention	O	81.27	84.45	82.83

TABLE V
ACCURACY ON MSRA

Models	Accuracy
Character + Dynamic-features	98.38
Character + Static-features	98.50
DS-attention	**98.62**

Accuracy on the MSRA dataset is shown in Table V. On the MSRA dataset, the accuracy of using static weighting reaches 98.50%, which is 0.12% higher than using only dynamic weighting. The combination of dynamic weighting and static weighting has an accuracy of 98.62%, which is 0.12% higher than the static weighting method and 0.24% higher than the dynamic weighting method.

We verified the portability of our method. On the MSRA dataset, the Bi-LSTM and ID-CNN models were used as the baseline model, which established that the dynamic weighting and static weighting methods could flexibly transplant various models and significantly improve the performance of the model. Results are shown in Table VI. Among them, on the MSRA dataset, the F1 score using the Bi-LSTM model reached 87.30%. Adding dynamic weighting and static weighting, the F1 score of the model increased by 3.46%, and the precision and recall have significantly improved. The F1 score on the MSRA dataset using the ID-CNN model reached 89.51%. Considering the ID-CNN model as the baseline model, the model's F1 score increased by 2.29% after using static weighting and dynamic weighting. Simultaneously, the model's precision increased by 3.68%, and the recall of the model increased by 0.98%. The experimental results show that the dynamic weighting and static weighting methods can better adapt to multiple models while improving its performance.

TABLE VI
RESULTS OF DIFFERENT BASELINE MODELS ON MSRA

Models	Precision	Recall	F1
Bi-LSTM + CRF	85.75	88.91	87.30
DS-attention + Bi-LSTM + CRF	**90.12**	**91.42**	**90.76**
ID-CNN + CRF	90.28	88.75	89.51
DS-attention + ID-CNN + CRF	**93.96**	**89.73**	**91.80**

TABLE VII
RESULTS ON RESUME

Models	Precision	Recall	F1
Zhang *et al.* 2018 [23]	94.81	94.11	94.46
Zhu *et al.* 2019 [41]	95.05	94.28	94.94
Gui *et al.* 2019 [42]	95.37	94.84	95.11
Character + Dynamic-features	95.56	95.24	95.40
Character + Static-features	**97.50**	94.91	94.85
DS-attention	95.37	**95.80**	**95.58**

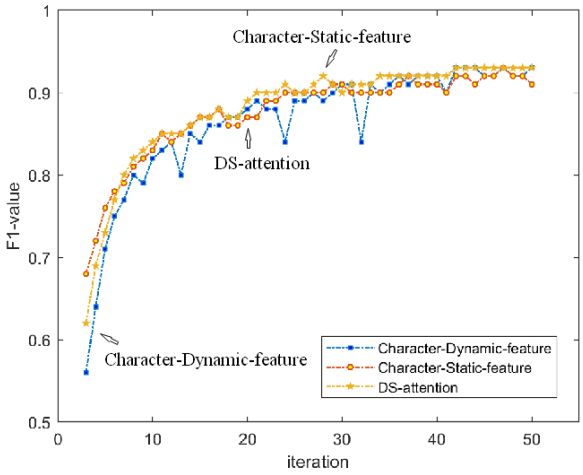

Fig. 4. F1 score against training iteration number

2) Resume: Results on the Resume dataset are shown in Table VII. In the table, Zhang *et al.* 2018 (Lattice model) mainly used string as the input of the neural network and used the self-match lexical words in the neural network layer. He used the self-matched words features as part of the network, which leads to the degree of structural dependence on the neural network during the model training process extensive. The last layer of the Lattice model uses CRF decoding, and a result of 94.4% is achieved on the testing dataset. The F1 score of our model on the testing set dataset reached 95.58%. Compared with the lattice model, the F1 score increased by 1.12%. The precision of our model is 0.56% higher than that of the Lattice model. At the same time, our model has stronger generalization ability, with a recall is increased by 1.69%. Compared with the Lattice model, our model improves the input layer, the model has better mobility and can avoid the dependence on the neural network structure. Zhu *et al.* 2019 proposed the Convolutional Attention Network on Chinese NER. This model used a character-based convolutional neural network with a local-attention layer and a gated recurrent unit with a global self-attention layer to capture the information from adjacent characters and sentence contexts. Compared with Zhu *et al.* 2019, our model has a 0.64% increase in the F1 score, and our model performs well. At the same time, our proposed models are more lightweight and have a 1.52% improvement in recall than Zhu *et al.* 2019. Gui *et al.* 2019 used a rethinking mechanism based on the convolutional neural network. It also used a combination of character features and self-matched features in the neural network. This model can utilize all the characters and potential words that match the word sequences in parallel. Besides, the rethinking mechanism can address the word conflict by feeding back the high-level features to refine the networks. Compared with Gui *et al.* 2019, our model has a 0.47% increase in the F1 score. Moreover, our model has better generalization ability, and the recall is increased by 0.96%.

To allow the model to learn more prior knowledge, we used both dynamic weighting and static weighting to extract self-matched word features. At the same time, we compared the dynamic weighting and static weighting methods. On the Resume dataset, we carried out dynamic weighting and static weighting experiments to observe the impact of dynamic weighting and static weighting on the experimental effect. Compared with the models of other authors in Table VII, both dynamic weighting and static weighting improve the model's performance to a certain extent because the dynamic weighting and static respectively extract the prior information in the text. Experimental results show that after static weighting and dynamic weighting, the model's precision is reduced, but the model's generalization ability is robust, and the robustness is improved. According to the particular mathematical properties of the attention mechanism, we believe that dynamic weighting extracts the semantic information in the word sequences, so the various indicators of the model are slightly improved. Static weighting is static features extracted by counting the number of self-matched words in a sentence. In this way, word frequency information in syntactic information is extracted. The word frequency information extracted through static statistics is volatile, and the precision has been significantly improved, but the recall of the model is greatly reduced. The fused model improves the instability of the static weighting method. Compared with the static weighting, the recall of the fused model is increased by 0.89%. Simultaneously, compared with the dynamic weighting, the F1 score of the fused model is increased by 0.18%, and recall is increased by 0.56%. We conducted a series of trials in which the static statistical method effectively extracts the word frequency features in the self-matched words, improves the model's performance, and makes the model more generalized. Fig. 4 shows the F1 score of different models on the Resume dataset. From the first 50 iterations, the F1 score of the DS-attention model has a steady upward trend.

At the same time, we tested the accuracy of different models on the Resume dataset. Accuracy on the Resume dataset is shown in Table VIII. On the Resume test dataset, the accuracy of using dynamic weighting reaches 97.62%, which is 0.12% higher than the accuracy of using only static weighting. The method of combining dynamic weighting and static weighting has an accuracy of 97.78%, which is 0.16% higher than the dynamic weighting method and 0.28% higher than the dynamic weighting method.

Results of different baseline models on the Resume dataset

TABLE VIII
ACCURACY ON RESUME

Models	Accuracy
Character + Dynamic-features	97.62
Character + Static-features	97.50
DS-attention	**97.78**

TABLE IX
RESULTS OF DIFFERENT BASELINE MODELS ON RESUME

Models	Precision	Recall	F1
Bi-LSTM + CRF	87.95	90.43	89.17
DS-attention + Bi-LSTM + CRF	**95.37**	**95.80**	**95.58**
ID-CNN + CRF	90.09	93.13	91.58
DS-attention + ID-CNN + CRF	**96.13**	**96.24**	**96.18**

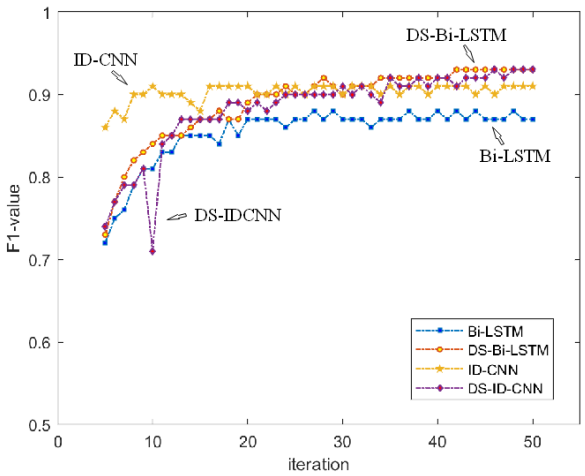

Fig. 5. F1 score of various baseline against training iteration number.

are shown in Table IX. On the Resume dataset, the F1 score of the Bi-LSTM baseline model was 89.17%. Based on the Bi-LSTM model, the F1 score using the combination of dynamic weighting and static weighting increased by 6.41%, and the recall increased by 5.37%. The F1 score using the ID-CNN model has achieved 91.58%. Based on the ID-CNN model, the F1 score using the combination of dynamic weighting and static weighting has increased by 4.6%, and the recall increased by 3.11%. Fig. 5 shows the F1 score of our models against the number of training iterations. In the early stage of iteration, adding dynamic weighting and static weighting methods has a more significant performance improvement, and this method has excellent robustness.

3) OntoNotes 4: Results on the OntoNotes4 dataset are shown in Table X. Based on the semi-supervised NER model, the training data generated contains much noise. Wang et al. proposed a factored probabilistic sequence model, which uses bilingual corpus to improve the noise problem of training data and enhance cross-language and intra-document consistency. The experimental results show that the bilingual CRF model performs better than the monolingual CRF model, and the F1 score is 74.32%. Che et al. enhanced the entity tag representation by Integer Linear Program, and the F1 score reached 75.02%. Yang et al. extracted neural features; the F1 value reached 68.75%. Integrating neural features and discrete indicator features, the F1 score of the model

TABLE X
RESULTS ON ONTONOTES 4

Models	Precision	Recall	F1
Wang *et al.* 2013 [43]	76.43	72.32	74.32
Che *et al.* 2013 [44]	77.71	72.51	75.02
Yang *et al.* 2016 [45]	65.59	71.84	68.57
Yang *et al.* 2016* [45]	72.98	80.15	**76.40**
Lattice LSTM 2018 [23]	76.35	71.56	73.88
Character + Dynamic-features	77.23	70.49	73.71
Character + Static-features	76.84	70.38	73.47
DS-attention	77.39	70.82	**73.96**

TABLE XI
RESULTS OF DIFFERENT BASELINE MODELS ON ONTONOTES 4

Models	Precision	Recall	F1
Bi-LSTM + CRF	77.21	64.33	70.18
DS-attention + Bi-LSTM + CRF	**77.39**	**70.82**	**73.96**
ID-CNN + CRF	78.18	66.35	71.78
DS-attention + ID-CNN + CRF	**77.97**	**71.64**	**74.67**

reached 76.40%. The F1 score of the DS-attention model on the OntoNotes 4 dataset is 73.96%. The F1 score of the bilingual corpus model of Wang et al. is 0.63% higher than that of the DS-attention model. However, the bilingual corpus model requires much bilingualism, and resource data costs are high. The joint model of Yang et al. increases the F1 score of the DS-attention model by 2.44%, but neural features and discrete indicator features require many computing resources, and the DS-attention model has a faster calculation speed. On the OntoNotes 4 dataset, compared with the model using dynamic weighting, the F1 score of the DS-attention model increased by 0.25% and compared with the model using static weighting, the F1 score of the DS-attention model increased by 0.49%.

Results of different baseline model are shown in Table XI. On the OntoNotes 4 dataset, the F1 score of the Bi-LSTM baseline model was 70.18%. Based on BiLSTM, the F1 score using the combination of dynamic weighting and static weighting increased by 3.78%, and the recall increased by 6.49%. The F1 score using the ID-CNN model has achieved 71.78%. Based on ID-CNN, the F1 score using the combination of dynamic weighting and static weighting has increased by 4.6%, and the recall has increased by 2.89%. Adding dynamic weighting and static weighting methods has a more significant performance improvement, and the method has better portability and robustness.

4) Weibo: Results on the Weibo dataset are shown in Table XII. Peng 2015 used a jointly learn embedding technology to apply on the NER system, and the F1 score reached 56.05%. Aiming at the frequent use of linguistic features in NER tasks, Peng 2016 jointly trained word segmentation and NER models, and the F1 score on the Chinese social media dataset reached 58.99%. He 2017a applied the semi-supervised learning model to the social media field dataset, and the F1 score reached 84.82%. The F1 score of He 2017b's cross-domain semi-supervised learning method reached 58.23%. The F1 score of the DS-attention on the Weibo dataset is 57.65%. The F1 score of the DS-attention is 1.6% higher than the Peng and 2.38% higher than the F1 score of the He2017a. Compared with the model

TABLE XII
RESULTS ON WEIBO

Models	Precision	Recall	F1
Peng *et al.* 2015 [46]	51.96	61.05	56.05
Peng *et al.* 2016 [47]	55.28	62.97	**58.99**
He *et al.* 2017a [48]	50.60	59.32	54.82
He *et al.* 2017b [49]	54.50	62.17	58.23
Character + Dynamic-features	54.08	56.49	55.70
Character + Static-features	54.74	55.61	55.17
DS-attention	56.72	58.62	**57.65**

TABLE XIII
RESULTS OF DIFFERENT BASELINE MODELS ON WEIBO

Models	Precision	Recall	F1
Bi-LSTM + CRF	51.62	56.38	53.89
DS-attention + Bi-LSTM + CRF	**56.72**	**58.62**	**57.65**
ID-CNN + CRF	53.64	57.72	55.60
DS-attention + ID-CNN + CRF	**55.94**	**60.31**	**58.04**

using dynamic weighting on the Weibo dataset, the F1 score of the DS-attention model is increased by 1.95%. Compared with the model using static weighting, the F1 score of the DS-attention model is increased by 2.48%; using matching word features can significantly improve the performance of social media NER tasks in vertical fields.

Results on Weibo dataset are shown in Table XIII. On the Weibo dataset, the F1 score of the Bi-LSTM baseline model reached 53.89%. Based on BiLSTM, the F1 value using the combination of dynamic weighting and static weighting increased by 3.76%, and the recall increased by 5.37%. The F1 score using the ID-CNN model has achieved 55.60%. Based on ID-CNN, the F1 score using the combination of dynamic weighting and static weighting has increased by 2.44%, and the recall has increased by 2.56%.

V. CONCLUSION

We proposed a weighted method to solve the Chinese NER task, in which the attention mechanism is used for dynamic weighting, and the statistical word frequency is used for static weighing. The experimental results show that good results have been achieved in the MSRA dataset and the Resume dataset. This method can effectively avoid the shortcomings of character-based and word-based methods. However, we have observed in experiments that this method has two shortcomings: First, external dictionaries are limited resources. Second, the model's static weighting method does not consider the boundary information of the self-matched words, which will cause the problem of missing word boundary information. In future work, we will optimize the model for these two problems.

REFERENCES

[1] Bracewell, David B., Fuji Ren, and Shingo Kuroiwa, "A Low Cost Machine Translation Method for Cross-Lingual Information Retrieval" Engineering Letters, vol. 16, no. 1, pp160-165, 2008.

[2] Kai Zheng, ZhiGuang Xia, Yi Zhang, Xuan Xu, and Yaqin Fu, "Speech Emotion Recognition based on Multi-Level Residual Convolutional Neural Networks" Engineering Letters, vol. 28 no. 2, pp559-565, 2020.

[3] H. James, "Named entity recognition with long short-term memory," in *HLT-NAACL 2003*, pp. 172-175.

[4] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *the Proc. 18th Int. Conf. Mach. Learn. (ICML) 2001*, pp. 282-289.

[5] N. Peng and M. Dredze, "Improving named entity recognition for Chinese Social media with word segmentation representation Learning," in *the Proc. 54th Annu. Meeting Assoc. Comput. Linguistics, (ACL) 2016*, pp. 149-155.

[6] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *the Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol. (NAACL-HLT) 2016*, pp. 260-270.

[7] X. Li, J. Feng, Y. Meng, Q. Han, F. Wu, and J. Li, "A unified MRC framework for named entity recognition," in *the Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, (ACL) 2020*, pp. 5849-5859.

[8] K. Dixit and Y. Al-Onaizan, "Span-level model for relation extraction," in *the Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, (ACL) 2019*, pp. 5308-5314.

[9] J. Straková, M. Straka and J. Hajic, "Neural architectures for nested NER through linearization," in *the Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, (ACL) 2019*, pp. 5326-5331.

[10] G. Zhou and J. Su, "Named entity recognition using an HMM-based chunk tagger," in *the Proc. 40th Annu. Meeting Assoc. Comput. Linguistics 2002*, pp. 473-480.

[11] L. Liu et al., "Empower sequence labeling with task-aware neural language model," in *the Proc. 32nd AAAI Conf. Artif. Intell. (AAAI) 2018*, pp. 5253-5260.

[12] A. Žukov-Gregorič, Y. Bachrach and S. Coope, "Named entity recognition with parallel recurrent neural networks," in *the Proc. 56th Annu. Meeting Assoc. Comput. Linguistics, (ACL) 2018*, pp. 69-74.

[13] L. Cui and Y. Zhang, "Hierarchically-refined label attention network for sequence labeling," *arxiv*, submitted for publication.

[14] J. Yang and Y. Zhang, "An open-source neural sequence labeling toolkit," in *the Proc. Proceedings of ACL 2018, System Demonstrations 2018*, pp. 74-79.

[15] K. Clark, M. Luong, C. D. Manning and Q. Le, "Semi-supervised sequence modeling with cross-view training" in *the Proc. Proceedings of ACL 2018, System Demonstrations 2018*, pp. 1914-1925.

[16] H. Peng, M. Chang and W. Yih, "Maximum margin reward networks for learning from explicit and implicit supervision" in *the Proc. Proceedings of EMNLP 2017, Empirical Methods 2017*, pp. 2368-2378.

[17] N. Zhang, F. Li, G. Xu, W. Zhang and H. Yu, "Chinese NER using dynamic meta-embeddings" *IEEE Access*, pp. 64450-64459.

[18] N. Zhang, G. Xu, Z. Zhang and F. Li, "MIFM: multi-granularity information fusion model for Chinese named entity recognitio" *IEEE Access*, pp. 181648-181655.

[19] P. Cao, Y. Chen, K. Liu, J. Zhao and S. Liu, "Adversarial transfer learning for Chinese named entity recognition with self-attention mechanism," in *the Proc. Proceedings of EMNLP 2018, Empirical Methods 2018*, pp. 182-192.

[20] F. Wu, J. Liu, C. Wu, Y. Huang and X. Xie, "Neural Chinese named entity recognition via CNN-LSTM-CRF and joint training with word segmentation," in *the Proc. The World Wide Web Conference 2019*, pp. 3342-3348.

[21] R. Ding, P. Xie, X. Zhang, W. Lu, L. Li and L. Si, "A neural multi-digraph model for Chinese NER with gazetteers," in *the Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, (ACL) 2019*, pp. 1462-1467.

[22] T. Gui, Y. Zou, Q. Zhang, M. Peng, J. Fu, Z. Wei and X. Huang, "A lexicon-based graph neural network for Chinese NER," in *Proc. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) 2019*, pp. 1040-1050.

[23] Y. Zhang and J. Yang, "Chinese NER using lattice LSTM," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (ACL) 2018*, pp. 1554-1564.

[24] W. Ralph et al. "OntoNotes Release 4.0"in *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium 2011*, pp. 201-204.

[25] L. John, A. McCallum, and F. Pereira."Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.", Proceedings of the Eighteenth International Conference on Machine Learning. 2001, pp. 282-289.

[26] P. Nanyun, and M. Dredze. "Improving Named Entity Recognition for Chinese Social Media with Word Segmentation Representation Learning." *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2016.*, pp. 149-155.

[27] D. Sui, Y. Chen, K. Liu, J. Zhao and S. Liu, "Leverage lexical knowledge for Chinese named entity recognition via collaborative graph network," in *Proc. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) 2019*, pp. 3830-3840.

[28] R. Ma, M. Peng, Q. Zhang, Z. Wei and X. Huang, "Simplify the usage of lexicon in Chinese NER," in *Proc. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL) 2020*, pp. 5951-5960.
[29] M. Volodymyr, N. Heess and A. Graves, "Recurrent models of visual attention," in *Proc. NIPS 2014*, pp. 5951-5960.
[30] D. Bahdanau, K. Cho and Y. Bengio , Neural machine translation by jointly learning to align and translate in *arxiv* submitted for publication.
[31] T. Luong, H. Pham and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. EMNLP 2015*, pp. 1412-1421.
[32] T. Xue et al., "Effective approaches to attention-based neural machine translation," in *arxiv* submitted for publication.
[33] X. Li, H. Yan, X. Qiu and X. Huang, "FLAT: Chinese NER using flat-lattice transformer," in *the Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, (ACL) 2020*, pp. 6836-6842.
[34] X. Dong, X. Xin and P. Guo, "Chinese NER by span-level self-attention," in *the Proc. 2019 15th International Conference on Computational Intelligence and Security (CIS) 2019*, pp. 68-72.
[35] Y. Jin, J. Xie, W. Guo, C. Luo, D. Wu and R. Wang, "LSTM-CRF neural network with gated self attention for Chinese NER," in *IEEE Access 2019*, pp. 136694-136703.
[36] G.-A. Levow, "The third international Chinese language processing bakeoff: Word segmentation and named entity recognition," in *Proc. 5th Workshop Chin. Lang. Process. 2006*, pp. 108-117.
[37] A. Chen, F. Peng, R. Shan, and G. G.-Z. Sun, "Chinese named entity recognition with conditional probabilistic models," in *Proc. 5th Workshop Chin. Lang. Process. 2006*, pp. 173-176.
[38] J. Zhou, L. He, X. Dai, and J. Chen, "Chinese named entity recognition with a multi-phase model," in *Proc. 5th Workshop Chin. Lang. Process. 2006*, pp. 213-216.
[39] J. Zhou, W. Qu, and F. Zhang, "Chinese named entity recognition via joint identification and categorization," in *Proc. Chin. J. Electron. 2013*, pp. 225-230.
[40] C. Dong, J. Zhang, C. Zong, M. Hattori, and H. Di, "Character-based LSTM-CRF with radical-level features for Chinese named entity recognition," in *Proc. Natural Lang. Understand. Intell. Appl., 5th CCF Conf. Natural Lang. Process. Chin. Comput. (NLPCC), 24th Int. Conf. Comput. Process. Oriental Lang. (ICCPOL). 2016*, pp. 239–250.
[41] Y. Zhu and G. Wang, "CAN-NER: convolutional attention network for Chinese named entity recognition," in *Proc. NAACL. 2019*, pp. 3384–3393.
[42] G. Tao et al., "CNN-Based Chinese NER with lexicon rethinking," in *Proc. IJCAI 2019*, pp. 4982-4988.
[43] W. Mengqiu, W. Che, and C. D. Manning. "Effective Bilingual Constraints for Semi-Supervised Learning of Named Entity Recognizers." *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence. 2013.*, pp. 919-925.
[44] W. Che, M. Wang, C. D. Manning, and T. Liu. "Named entity recognition with bilingual constraints." *In HLT-NAACL. 2013.*, pp. 52–62.
[45] J. Yang, Z. Teng, M. Zhang, and Y. Zhang. "Combining discrete and neural features for sequence labeling" *International Conference on Intelligent Text Processing and Computational Linguistics. 2016.*, pp. 140-154.
[46] N. Peng and M. Dredze. "Named entity recognition for Chinese social media with jointly trained embeddings." *In EMNLP. 2015.*, pp. 548–554.
[47] N. Peng and M. Dredze. "Improving named entity recognition for Chinese social media with word segmentation representation learning." *In ACL. 2016.*, pp. 149–155.
[48] H. He and X. Sun. "F-score driven max margin neural network for named entity recognition in Chinese social media." *In EACL. 2017a.*, pp. 713–718.
[49] H. He and X. Sun. "A unified model for cross-domain and semi-supervised named entity recognition in Chinese social media." *In AAAI. 2017b.*, pp. 3216–3222.