MGCN: A Novel Multi-Graph Collaborative Network for Chinese NER

Yingqi Zhang[1], Wenjun Ma[1], and Yuncheng Jiang[1,2(✉)]

[1] School of Computer Science, South China Normal University, Guangzhou 510631, China
{zhangyingqi,jiangyuncheng}@m.scnu.edu.cn, phoenixsam@sina.com
[2] School of Artificial Intelligence, South China Normal University, Foshan 528225, China

Abstract. Named Entity Recognition (NER), one of the most important directions in Natural Language Processing (NLP), is an essential pre-processing step in many downstream NLP tasks. In recent years, most of the existing methods solve Chinese NER tasks by leveraging word lexicons, which has been empirically proven to be useful. Unfortunately, not all word lexicons can improve the performance of the NER. Some self-matched lexical words will either disturb the prediction of character tag, or bring the problem of entity boundaries confusion. Thus, the performance of the NER model will be lowered by such irrelevant lexical words. However, to the best of our knowledge, none of the existing methods can solve these challenges. To address these issues, we present a novel Multi-Graph Collaborative Network (MGCN) for Chinese NER. More specifically, we propose two innovative modules for our methods. Firstly, we build connections among characters to eliminate interferential influences of the noisiness in lexical knowledge. Secondly, by constructing relationship between contextual lexical words, we solve the problem of boundaries confusion. Finally, experimental results on the benchmark Chinese NER datasets show that our methods are not only effective, but also outperform the state-of-the-art (SOTA) results.

Keywords: Chinese NER · Lexical knowledge · Graph neural network

1 Introduction

NER is mainly dedicated to identifying and classifying unstructured texts into predefined semantic categories such as person names, locations, etc. [12,23]. NER not only acts as a standalone tool for information extraction (IE) [1], but also plays an essential role in a variety of NLP applications such as text understanding [25], information retrieval [8], recommendation system [5], etc.

In contrast with NER in English, Chinese NER is relatively difficult because sentences in Chinese are not naturally segmented. Therefore, it is common for Chinese NER to first perform word segmentation by using an existing Chinese Word Segmentation (CWS) system and then use a sequence labeling model based

W. Lu et al. (Eds.): NLPCC 2022, LNAI 13551, pp. 618–630, 2022.
https://doi.org/10.1007/978-3-031-17120-8_48

on word-level to separate sentence [6]. However, it is difficult for the CWS system to correctly segment query sentences, which will result in error propagation. In order to solve this problem, there are some methods resorting to performing Chinese NER directly at the character-level, which has been empirically proven to be effective [14]. However, such methods cannot exploit the lexical knowledge. With this consideration, Zhang et al. [24] proposed the Lattice-LSTM model to exploit explicit word and word sequence information. However, the architecture of this model is too complicated, which causes relatively poor training and inference speeds. Besides the Lattice-LSTM model, Sui et al. [19] proposed a Collaborative Graph Network to use word information by integrating lexical knowledge. Their models use lexical word information to obtain great experimental results, but they neglect the interferential influences of the noisiness in lexical matched words. Therefore, there are still two challenges.

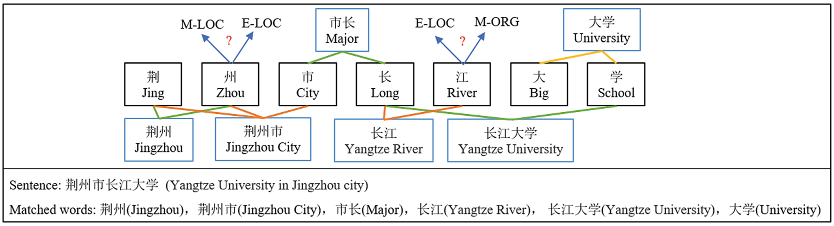

Fig. 1. An example sentence influenced by the noisiness in lexical knowledge

The first challenge is how to eliminate the interferential influences of the noisiness in lexical matched words. As shown in Fig. 1, for the sentence “荆州市长江大学” (there is Yangtze University in Jingzhou city), the words “荆州市” (Jingzhou City) and “荆州” (Jingzhou) are the self-matched words of the character “荆” (Jing) in this sentence. If the word “荆州” (Jingzhou) is recognized as an entity in the sentence, it influences the prediction of the characters “州” (Zhou) and “市” (City), which will influence the meaning of this sentence. In details, the character “州” (Zhou) may be predicted as an “E-LOC” tag, and the character “市” (City) may be incorrectly predicted as a “B-PER” tag. Worst of all, the model may recognize the words “荆州” (Jingzhou), “市长” (Major), and “江大学” (Daxue Jiang) as three entities in this sentence, respectively. But in fact, the meaning of this sentence presents that there is Yangtze University in Jingzhou city. It is not that Daxue Jiang is a major in Jingzhou city. As we can know from above, the matched word “荆州” (Jingzhou) is noisiness for the sentence, which interferes the prediction of the NER models.

The second challenge is how to solve the problem of entity boundaries confusion. For the same example of “荆州市长江大学” (there is Yangtze University in Jingzhou city), the matched words “长江大学” (Yangtze University) and “长江 ” (Yangtze River) are the self-matched words of the character “长” (Long). The matched words “长江大学” (Yangtze University) and “大学” (University) are the self-matched words of the character “大” (Big).

Due to the influences of the matched words “长江” (Yangtze River) and “大学” (University), the model cannot define the boundary of sub-sentence “长江大学” (Yangtze University). That is, the character “江” (River) may be predicted as an “E-LOC” tag, and the character “大” (Big) may be predicted as a “B-ORG” tag. Hence, based on such tags, the model may recognize the words “长江” (Yangtze River) and “大学” (University) as two entities, which is caused by the confusion of entity boundaries. In fact, the word “长江” (Yangtze River) is just a part of the true entity “长江大学” (Yangtze University) in this sentence, as well as the word “大学” (University), and it does not indicate the longest river in China. However, to the best of our knowledge, none of the existing methods can solve this problem. For example, these methods cannot incorporate the words “长江” (Yangtze River) and “大学” (University) into the word “长江大学” (Yangtze University).

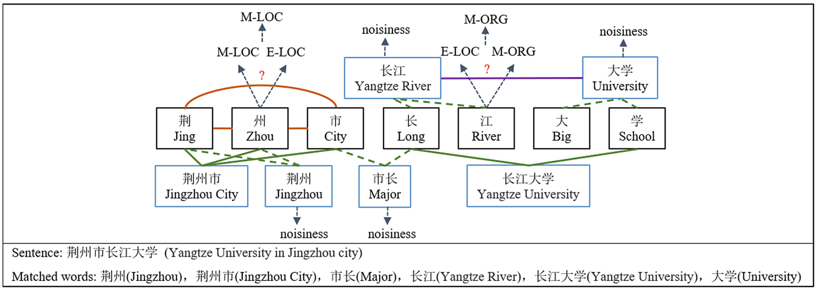

Fig. 2. An example sentence fully exploiting lexical knowledge

In order to solve the above challenges, we think that the connections among characters should be built, and the relationships between the nearest contextual matched words should be fully considered. As shown in Fig. 2, the connections among characters can solve the first challenge. If the character “州” (Zhou) has a connection with its predecessor (the character “荆” (Jing)) or successor (the character “市” (City)), at the same time, the character “荆” (Jing) has a connection with the character “市” (City). Based on graph neural network, these characters are more likely recognized as an entity. In this way, we can eliminate the interferential influences of the matched word “荆州” (Jingzhou). For the second challenge, the relationships among contextual lexical words can overcome the obstacle of boundaries confusion. Specifically, if the word “长江” (Yangtze River) has been seen as an entirety, as well as the word “大学” (University), there is not any connection between the words “长江” (Yangtze River) and “大学” (University), which may cause incorrect prediction. Therefore, we can use the matched word “长江大学” (Yangtze University) to construct a relationship between the matched words “长江” (Yangtze River) and “大学” (University). In this way, the words “长江” (Yangtze River) and “大学” (University) are incorporated into the word “长江大学” (Yangtze University), and the model just recognizes the word "长江大学" (Yangtze University) as an entity. The more detailed explanation is introduced in Sect. 3.

In this vein, we propose a novel multi-graph collaborative network model. Specifically, we construct three word-character interactive graphs in the graph layer for achieving our methods. The first graph is the Constructing graph, which not only builds connections among characters, but also constructs the most decisive self-matched words in a sentence. The second graph is the Associating graph, which exploits the enhanced relationships among characters to build relationships between the contextual words, and connections between the character and its nearest words. The third graph is the Boundary graph, which is designed for confirming the boundaries of named entities, in order to solve the confusion of entity boundaries. Since different graphs have different functions, they will cooperate with each other via a fusion layer.

In summary, our main contributions are as follows: 1) we propose a novel multi-graph collaborative network for Chinese NER tasks; 2) we eliminate the interferential influences of the noisiness in matched words by constructing connections among characters; 3) we solve the problem of entity boundary confusion by building relationships between contextual words; 4) experimental results show that our methods outperform the SOTA results.

2 Related Work

Some previous Chinese NER studies have shown that character-based methods [11], can outperform word-based counterparts [7], due to the error propagation caused by CWS. Recently, some lexical knowledge methods have been widely used to augment character information for Chines NER, which has been empirically proven to be effective [18]. Especially, the Lattice-LSTM model proposed by Zhang et al. [24] not only avoids error propagation, but also models characters and potential words simultaneously.

Transformer-based methods have also been used with lexical enhancement [13]. Especially, Flat-Lattice Transformer proposed by Li et al. [13] can convert the lattice structure into a flat structure consisting of spans. This method not only has an excellent parallelization ability, but also leverages the lattice information. Besides, graph neural networks have been successfully applied to Chinese NER tasks [4,19,26]. Gui et al. [4] has proposed a lexicon-based graph network for Chinese NER. This method treats the named entities as a node classification task, which can avoid error propagation and leverage lexical knowledge.

In this work, we adopt a novel multi-graph collaborative network method that builds three word-character interactive graphs with different functions. This method not only leverages lexical knowledge, but also solves problems caused by noisiness in lexicon words.

3 Methodology

In this section, we first introduce the construction of three word-character interactive graphs. Then, we introduce the structure of our training model for solving Chinese NER tasks.

3.1 The Construction of Graphs

We construct three word-character interactive graphs to achieve our methods. For any graph, the vertices of the graph consist of characters and the lexical words matched by the corresponding characters in the sentence. For example, an input sentence can be represented as $s = \{$荆, 州, 市, 长, 江, 大, 学$\}$. In order to utilize potential words in the sentence, we match all lexical words of every character. All lexical words matched by the corresponding characters can be represented as $l = \{$荆州市, 荆州, 市长, 长江大学, 长江, 大学$\}$. Thus, the vertices of three interactive graphs are denoted as $V = \{$荆$, \ldots,$ 长江大学$, \ldots\}$.

The vertices of three interactive graphs are the same, but the edges of each graph are different. For this, we introduce adjacency matrix to represent the edges of each graph. The values in the adjacency matrix indicate whether there are relations between vertices or not in a graph. Since the functions of different graphs are different, their adjacency matrices are also different.

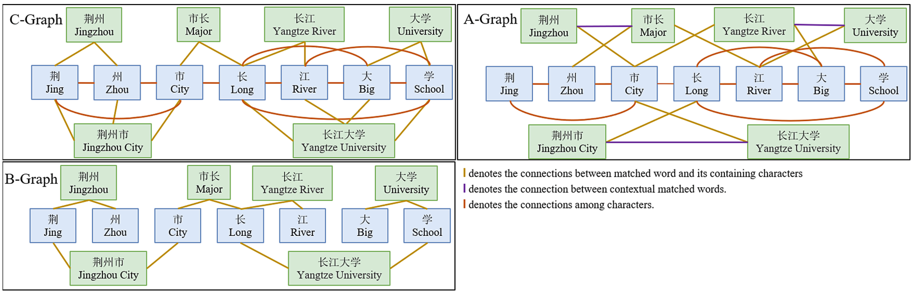

Fig. 3. Three word-character interactive graphs

Associating Graph. Inspired by Sui et al. [19], we build the Associating graph (A-graph). With this graph, we not only build relationships among contextual words, but also connections between character and its nearest words, by using the enhanced relationships among characters. Firstly, we should augment relationships among characters. As shown in Fig. 3, if a lexical word l_i contains character set $C = \{c_1, c_2, \ldots, c_p\}, c_n, c_m \in C$, the (c_n, c_m)-entry of the A-graph corresponding adjacency matrix A^A is assigned a value of 1. To capture the semantic information between the character and its nearest contextual words, if a lexical word l_i matches the nearest character c, $A^A_{l_i c}$ will be assigned a value of 1. Moreover, in order to build relationship between lexical words, if a lexical word l_i is the previous or next context of another lexical word l_j, the (l_i, l_j)-entry of the A-graph corresponding adjacency matrix A^A is assigned a value of 1.

Boundary Graph. The Boundary graph (B-graph) is constructed to use self-matched lexical words to determine the boundaries of entities, in order to eliminate the confusion of entity boundaries. As shown in Fig. 3, if a lexical word l_i contains many characters, we need to leverage its contained first character or last character c. Therefore, the (l_i, c)-entry of the B-graph corresponding adjacency matrix A^B is assigned a value of 1.

Constructing Graph. With this Constructing graph (C-graph), we not only build connections among characters, but also construct the most decisive self-matched words in a sentence. As shown in Fig. 3, if a lexical word l_i contains character set $C = \{c_1, c_2, \ldots, c_p\}, c_n, c_m \in C$, we will assign the (l_i, c_n)-entry of the C-graph corresponding adjacency matrix A^C a value of 1, as well as the (c_n, c_m)-entry.

3.2 The Whole Architecture of Our Model

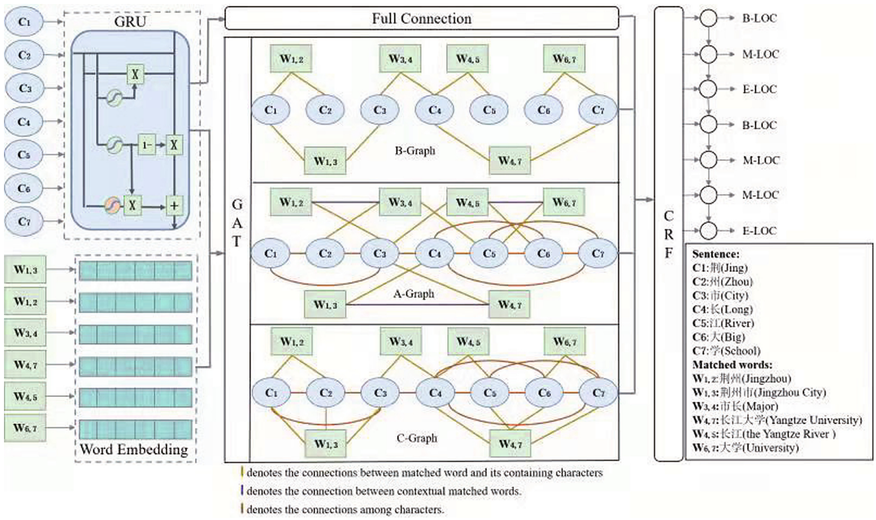

Fig. 4. The architecture of our model

As shown in Fig. 4, the whole architecture of our training model is as follows. First of all, every character in the input sequence is converted into a dense vector. Secondly, we utilize a bidirectional Gated Recurrent Unit to capture contextual information of input sequence, and then fuse it with three word-character interactive graphs, respectively. In the end, the results of the final predictions are obtained through Conditional Random Field (CRF).

Encoding Layer

The input of the training model based on characters is a sentence. The sentence can be denoted as $s = \{c_1, c_2, \cdots, c_n\}$, where c_i is the i-th character in a sentence. By looking up the embedding vector, each character c_i can be represented as a dense vector, which denotes as $\mathbf{x}_i^c$:

$$\mathbf{x}_i^c = e^c(c_i), \tag{1}$$

where e^c is a character embedding lookup table.

Recurrent Neural Network (RNN) is beneficial to capturing contextual information of Chinese sentences. In this paper, we adopt a bidirectional Gated Recurrent Unit (GRU) network. Compared with other RNNs, GRU has a merit in training speed. As shown in Eq. (2), the bidirectional GRU can be applied to the input sentence $\mathbf{x}^c = \{\mathbf{x}_1^c, \mathbf{x}_2^c, \cdots, \mathbf{x}_n^c\}$, and then we can obtain the contextual representation $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h_n}\}$.

$$\mathbf{h}_i = \overrightarrow{GRU}\left(\mathbf{x}_i^c, \overrightarrow{\mathbf{h}}_{i-1}\right) \oplus \overleftarrow{GRU}\left(\mathbf{x}_i^c, \overleftarrow{\mathbf{h}}_{i+1}\right). \tag{2}$$

We utilize lexical knowledge to augment the character representation, in order to enhance performance of our model. All lexical words matched by the corresponding characters, can denoted as $l = \{l_1, l_2, \cdots, l_m\}$. By looking up the pre-trained embedding lookup table, each lexical word is represented as a dense vector, which denotes as $\mathbf{x}_i^l$:

$$\mathbf{x}_i^l = e^l(l_i), \tag{3}$$

where e^l is a lexical embedding looking table.

In the end, output representation of the encoding layer can be shown in Eq. (4).

$$\mathbf{X} = \left[\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_n, \mathbf{x}_1^l, \mathbf{x}_2^l, \cdots, \mathbf{x}_m^l\right]. \tag{4}$$

GATs over These Graphs

Graph Attention Network (GAT) [20] can allow for assigning different importances to different nodes with a neighborhood. In this work, we adopt GAT to model three word-character interactive graphs. In an M-layer GAT, the input representation of j-th layer consists of a set of node features, $\mathbf{NF}^j = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N\}$, together with an adjacency matrix $\mathbf{A}$, $\mathbf{f}_i \in \mathbb{R}^F, \mathbf{A} \in \mathbb{R}^{N\times N}$, where F denotes the dimension of features at j-th layer and N is the number of nodes. The output representation of j-th layer is a new set of node features differing with others $\mathbf{NF}^{(j+1)} = \{\mathbf{f}'_1, \mathbf{f}'_2, \ldots, \mathbf{f}'_N\}$. Every GAT operation with K different and independent attention heads is shown in Eqs. (5) and (6):

$$\mathbf{f}'_i = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \mathbf{f}_j\right), \tag{5}$$

$$\alpha_{ij}^{k} = \frac{\exp\left(\mathrm{LeakyReLU}\left(\mathbf{a}^{\mathrm{T}}\left[\mathbf{W}^{\mathrm{k}}\mathbf{f}_{\mathrm{i}} \| \mathbf{W}^{\mathrm{K}}\mathbf{f}_{\mathrm{j}}\right]\right)\right)}{\Sigma_{k\in\mathscr{N}_i}\exp\left(\mathrm{LeakyReLU}\left(\mathbf{a}^{\mathrm{T}}\left[\mathbf{W}^{\mathrm{k}}\mathbf{f}_{\mathbf{i}} \| \mathbf{W}^{\mathrm{K}}\mathbf{f}_{\mathrm{k}}\right]\right)\right)}, \tag{6}$$

where concatenation operation is denoted as $\|$. The nonlinear activation function is denoted as σ. The adjacent nodes of node i in a graph are denoted as $\mathscr{N}_i$. The attention coefficients are denoted as α_{ij}^k, $\mathbf{W}^k \in \mathbb{R}^{F'\times F}$ and $\mathbf{a} \in \mathbb{R}^{2F'}$. The single-layer feed-forward neural network is denoted as $\mathbf{a} \in \mathbb{R}^{2F'}$. Note that, KF' is the dimension of the output $\mathbf{f}_i'$, and F' is the dimension of the final output features. Finally, the averaging in the last layer will be kept.

$$\mathbf{f}_i^{final} = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathscr{N}_i}\alpha_{ij}^k\mathbf{W}^k\mathbf{f}_j\right). \tag{7}$$

In detail, three independent graph attention networks are built for modeling three different word-character interactive graphs. Three independent graph attention networks can be denoted as $GAT1$, $GAT2$, and $GAT3$, respectively. The same vertex set is shared by three word-character interactive graphs. The input node features of all GAT models are the input representation X, which is shown in Eq. (4). These three GAT models denote different output node features, as shown in Eq. (8):

$$G_K = GAT_K\left(X, A^p\right), \tag{8}$$

where $\mathbf{G}_k \in \mathbb{R}^{F'\times(n+m)}, k \in \{1,2,3\}$, $p \in \{C, A, B\}$, n is the number of characters in the sentence, and m is the number of the lexical words matched by characters in the sentence. We do not need all columns of these matrices, because some columns are interferential for our model. Therefore, the front n columns of these matrices are kept to decode labels.

$$\mathbf{Q}_k = \mathbf{G}_k[:, 0:n], k \in \{1,2,3\}. \tag{9}$$

Fusion Layer

In our model, three graphs have different functions. To utilize the merits of different interactive graphs, we use a fusion layer to fuse three $GATs$. In addition, the input representation of original characters is also needed. Therefore, the input representation of the fusion layer is the contextual representation $\mathbf{H}$ and the output of the $GATs$ $\mathbf{Q}_i, i \in \{1,2,3\}$. The fusion equation is as follows:

$$\mathbf{R} = \mathbf{W}_1\mathbf{H} + \mathbf{W}_2\mathbf{Q}_1 + \mathbf{W}_3\mathbf{Q}_2 + \mathbf{W}_4\mathbf{Q}_3, \tag{10}$$

where $\mathbf{W}_y, y \in \{1,2,3,4\}$, is a trainable matrix. We can obtain a collaborative matrix $\mathbf{R}$ from a fusion layer, which can integrate these different matrices. The matrix $\mathbf{R}$ includes contextual information, the relations among characters and the relationship between nearest words.

Decoding

A standard CRF [9] layer is adopted to capture the dependencies between successive labels. For any input sentence $s = \{c_1, c_2, \cdots, c_n\}$, the input representation of this layer is $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_n\}$ and the probability of a label sequence $y = \{y_1, y_2, \cdots, y_n\}$ is as follows:

$$p(y \mid s) = \frac{\exp\left(\sum_i \left(\mathbf{W}^{y_i}\mathbf{r}_i + \mathbf{T}_{(y_{i-1}, y_i)}\right)\right)}{\sum_{y'} \exp\left(\sum_i \left(\mathbf{W}^{y'_i} r_i + \mathbf{T}_{\left(y'_{i-1}, y'_i\right)}\right)\right)}, \tag{11}$$

where an arbitrary label sequence is denoted as y'_i. A model parameter specific to the i_{th} character in the sentence is denoted as $\mathbf{W}^{y_i}$ and the associating matrix is denoted as $\mathbf{T}$. The first-order Viterbi algorithm [21] is used to find the highest scored label sequence over a character-based input representation. Given a manually labeled training data $\{(s_i, y_i)\}|_{i=1}^{N}$, the optimized model is obtained by using sentence-level log-likelihood loss with L_2 regularization.

$$L = -\sum_{i=1}^{N} \log\left(P\left(y_i \mid s_i\right)\right) + \frac{\lambda}{2}\|\Theta\|^2. \tag{12}$$

As shown in Eq. 12, the L_2 regularization parameter is represented as λ, and the training parameters set is denoted as Θ.

4 Experiment

In this section, we show experimental processes, including tested datasets, evaluation metric (F1) and so on. We test our methods with three datasets. Specifically, our datasets include MSRA [10], Weibo NER [17] and E-commerce [2].

4.1 Overall Performance

Table 1. Main results on all datasets.

Models	MSRA	E-commerce	Weibo-ALL	Weibo-NM
Lattice-LSTM [24]	93.18	-	58.79	62.25
CAN-NER [26]	92.97	-	59.31	62.98
LR-CNN [3]	93.71	-	59.92	66.67
LGN [4]	93.46	-	60.21	64.98
SoftLexicon(LSTM) [15]	93.66	73.59	61.42	62.22
+bichar [15]	94.06	73.88	59.81	64.20
MECT [22]	94.32	72.27	63.30	62.51
LGCN [16]	94.08	-	61.10	65.30
Ours	**94.34**	**76.42**	**64.31**	**71.46**

Evaluation metric is F1 value.

MSRA. There is a larger amount of data in MSRA dataset, compared with other datasets. There is an important task to evaluate the robustness and effectiveness of training model besides the problem about quality and consistency of the annotation. Our methods can keep stronger robustness in front of these problems because we consider the relationships among characters well. The C-graph bridges the gap between characters with some relationships. In addition, the A-graph builds connections between self-matched lexical words. Moreover, the B-graph solves the problem of confusion in entity boundaries. The results in Table 1 show that our methods are effective and useful. Compared with the MECT model [22], our methods not only obtain the slightly better performance, but also do not depend on the quality of structure decomposition dictionary of Chinese character.

Weibo and E-commerce. NER datasets on informal text are more challenging than on formal text because of the shortness and noisiness. Compared with other datasets, there is more shortness and noisiness in Weibo and E-commerce datasets, which may cause poor performance of the neural model. In addition, the connections among entities are not close together. With this consideration, we construct the C-graph. The C-graph can build the close relationships among characters by using lexicon knowledge. Those close relationships are beneficial to the recognition of entities. The results on these two datasets show that our methods are useful, as shown in Table 1. Compared with the MECT model [22], our model can outperform it by 4.15%, 1.01%, 8.95% in F1 score on E-commerce, Weibo-All, and Weibo-NM datasets, respectively.

4.2 Effectiveness

The ablation experiments show the effectiveness of three interactive graphs based on word-character.

Comparison Settings. The specific details of the ablation studies are as follows: 1) GRU+CRF: baseline model. 2) GRU + A: just keep the A-graph. 3) GRU + A + B: without the C-graph.

Table 2. Ablation study

Models	MSRA	E-commerce	Weibo-ALL	Weibo-NM
GRU+CRF	87.80	61.89	48.02	51.87
GRU+A	91.14	68.15	60.43	67.41
GRU+C	90.5	67.04	60.99	66.66
GRU+B	92.41	68.32	62.41	65.34
GRU+A+C	93.85	74.18	63.24	68.63
GRU+A+B	93.64	73.45	63.48	69.37
GRU+C+B	94.07	74.97	64.17	69.89
Complete model	**94.34**	**76.42**	**64.31**	**71.46**

Comparison Results. The results of the ablation study are shown in Table 2. We can know that removing any graph can cause badly poor performance of the model in different datasets. In addition, we see that "GRU+CRF" obtains worse results than others from any dataset, which obviously shows that our methods are effective and useful. Moreover, we know that the relationship between nearest words is beneficial to Chinese NER from "GRU+C+B" model. In conclusion, the statistics of ablation experiments show that each graph is indispensable, but the best performance can be obtained by them together.

5 Conclusion

In this paper, we present a novel multi-graph collaborative network to solve the problem caused by noisiness in lexical knowledge. Specifically, we build connections among characters to eliminate the interferential influences of the noisiness in matched words. Secondly, by constructing relationship between contextual lexical words, we solve the problem of boundaries confusion. In the end, we construct three word-character interactive graphs with different functions. The various experiments show that our methods are effective and useful.

Acknowledgements. The works described in this paper are supported by The National Natural Science Foundation of China under Grant Nos. 61772210 and U1911201; The Project of Science and Technology in Guangzhou in China under Grant No. 202007040006.

References

1. Dai, W., Hua, X., Lv, R., Bo, R., Chen, S.: The solution of Xiaomi AI lab to the 2021 language and intelligence challenge: multi-format information extraction task. In: Natural Language Processing and Chinese Computing, pp. 496–508 (2021)
2. Ding, R., Xie, P., Zhang, X., Lu, W., Li, L., Si, L.: A neural multi-digraph model for Chinese NER with gazetteers. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1462–1467 (2019)
3. Gui, T., Ma, R., Zhang, Q., Zhao, L., Jiang, Y.G., Huang, X.: CNN-based Chinese NER with lexicon rethinking. In: IJCAI, pp. 4982–4988 (2019)
4. Gui, T., et al.: A lexicon-based graph neural network for chinese NER. In: Empirical Methods in Natural Language Processing, pp. 1039–1049 (2019)
5. Hao, B., et al.: Negative feedback aware hybrid sequential neural recommendation model. In: Zhu, X., Zhang, M., Hong, Yu., He, R. (eds.) NLPCC 2020. LNCS (LNAI), vol. 12431, pp. 279–291. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60457-8_23
6. He, H., Sun, X.: F-score driven max margin neural network for named entity recognition in Chinese social media. In: European Chapter of the Association for Computational Linguistics, pp. 713–718 (2017)
7. He, J., Wang, H.: Chinese named entity recognition and word segmentation based on character. In: Third International Joint Conference on Natural Language Processing, IJCNLP, pp. 128–132 (2008)

8. Hersh, W.: Information retrieval. In: Shortliffe, E.H., Cimino, J.J. (eds.) Biomedical Informatics. LGTS, pp. 755–794. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-58721-5_23
9. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning, pp. 282–289 (2001)
10. Levow, G.A.: The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In: SIGHAN Workshop on Chinese Language Processing, pp. 108–117 (2006)
11. Li, H., Hagiwara, M., Li, Q., Ji, H.: 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.,pp. 2532–2536 (2014)
12. Li, H., Xu, H., Qian, L., Zhou, G.: Multi-layer joint learning of Chinese nested named entity recognition based on self-attention mechanism. In: Zhu, X., Zhang, M., Hong, Yu., He, R. (eds.) NLPCC 2020. LNCS (LNAI), vol. 12431, pp. 144–155. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60457-8_12
13. Li, X., Yan, H., Qiu, X., Huang, X.: FLAT: Chinese NER using flat-lattice transformer. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6836–6842 (Jul 2020)
14. Liu, Z., Zhu, C., Zhao, T.: Chinese named entity recognition with a sequence labeling approach: Based on characters, or based on words? In: Advanced Intelligent Computing Theories and Applications, With Aspects of Artificial Intelligence, pp. 634–640 (2010)
15. Ma, R., Peng, M., Zhang, Q., Wei, Z., Huang, X.: 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)
16. Nie, Y., Zhang, Y., Peng, Y., Yang, L.: Borrowing wisdom from world: modeling rich external knowledge for Chinese named entity recognition. Neural Comput. Appl. **34**(6), 4905–4922 (2022)
17. Peng, N., Dredze, M.: Named entity recognition for Chinese social media with jointly trained embeddings. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 548–554 (2015)
18. Song, B., Bao, Z., Wang, Y.Z., Zhang, W., Sun, C.: Incorporating lexicon for named entity recognition of traditional Chinese medicine books. In: Zhu, X., Zhang, M., Hong, Yu., He, R. (eds.) NLPCC 2020. LNCS (LNAI), vol. 12431, pp. 481–489. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60457-8_39
19. Sui, D., Chen, Y., Liu, K., Zhao, J., Liu, S.: Leverage lexical knowledge for Chinese named entity recognition via collaborative graph network. In: EMNLP-IJCNLP, pp. 3830–3840 (2019)
20. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018)
21. Viterbi, A.J.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans. Inf. Theory **13**(2), 260–269 (1967)
22. Wu, S., Song, X., Feng, Z.: MECT: Multi-metadata embedding based cross-transformer for Chinese named entity recognition. In: ACL-IJCNLP, pp. 1529–1539 (2021)

23. Yan, T., Huang, H., Mao, X.-L.: SciNER: a novel scientific named entity recognizing framework. In: Zhu, X., Zhang, M., Hong, Yu., He, R. (eds.) NLPCC 2020. LNCS (LNAI), vol. 12430, pp. 828–839. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60450-9_65
24. Zhang, Y., Yang, J.: Chinese NER using lattice LSTM. In: Gurevych, I., Miyao, Y. (eds.) Association for Computational Linguistics, pp. 1554–1564 (2018)
25. Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q.: Ernie: enhanced language representation with informative entities. arXiv preprint arXiv:1905.07129 (2019)
26. Zhu, Y., Wang, G.: CAN-NER: convolutional attention network for Chinese named entity recognition. In: NAACL, pp. 3384–3393 (2019)