MCSN: Multi-graph Collaborative Semantic Network for Chinese NER

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

[1] School of Computer Science, South China Normal University,
Guangzhou 510631, China
{zhangyingqi,gwen,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) is not only one of the most important directions in Natural Language Processing (NLP), but also plays an essential pre-processing role in many downstream NLP tasks. In recent years, most of the existing methods solve Chinese NER tasks by leveraging word lexicon, which have been empirically proven to be effective. However, these methods that depend on lexical knowledge too much tend to be confused by lexicon words, which leads to recognizing false entities. In addition, the lexicon is just a method to augment performance for the NER models, but cannot provide the dependency information of every Chinese word in a sentence, which causes relatively poor results in complex text. In order to solve these issues, this paper proposes a Multi-graph Collaborative Semantic Network (MCSN) fusing the dependency information of Chinese words. We build the dependency relationships of Chinese words by leveraging Graph Attention Network. With the dependency relationships of Chinese words, MCSN not only overcomes the shortages of lexicon, but also better captures the semantic information of Chinese words. Experimental results on some Chinese benchmarking datasets show that our methods are not only effective, but also outperform the state-of-the-art (SOTA) results. Especially in the Weibo-NM dataset, our methods can outperform it more than 9.34% in F1 score, in contrast with the SOTA models.

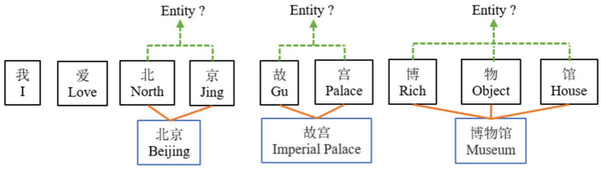
Keywords: Chinese NER · Dependency information · Graph attention network

1 Introduction

NER is one of the most important directions of NLP, which is designed for identifying and classifying unstructured texts into predefined semantic categories such as person names, organizations, etc. [9,11]. Moreover, NER plays an essential role in a variety of NLP applications such as information extraction (IE) [2], text understanding [25], information retrieval [17], knowledge graph [1,23], recommendation system [22] etc.

G. Memmi et al. (Eds.): KSEM 2022, LNAI 13368, pp. 322–334, 2022.
https://doi.org/10.1007/978-3-031-10983-6_25

In contrast with an English sentence, there is no space between characters in a Chinese sentence as word delimiters. Therefore, it is common for Chinese NER to first perform word segmentation by using an existing Chinese Word Segmentation (CWS) system and then apply a sequence labeling model based on word-level to segmented sentence [6,21]. 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 [13]. However, such methods cannot exploit lexical knowledge. With this consideration, Zhang et al. [24] proposed the Lattice-LSTM model to exploit explicit word and word sequence information. Besides, Li et al. [10] presented a Flat-Lattice Transformer, which converts the lattice structure into a flat structure consisting of spans. These methods can increase the performance of NER models by leveraging lexical knowledge, but tend to neglect the dependency relationships of Chinese words, which leads to recognizing false entities.



Sentence: 我爱北京故宫博物馆 (I love the Palace Museum in Beijing)

Matched words: 北京(Beijing)，故宫(Imperial Palace)，博物馆(Museum)

Fig. 1. A sentence just leveraging lexical knowledge

As shown in Fig. 1, for the sentence “我爱北京故宫博物馆” (I love the Palace Museum in Beijing), the matched word “北京” (Beijing) can better increase the relation between character “北” (North) and “京” (Jing), so the model is likely to recognize these two characters as an entirety, as well as other matched words. As a result, the model will recognize the word “北京” (Beijing), “故宫” (Imperial Palace) and “博物馆” (Museum) as an entity, respectively. But in fact, there is only one entity in this sentence, which is the word “北京故宫博物馆” (the Palace Museum in Beijing). To the best of our knowledge, the existing NER models cannot recognize the word “北京故宫博物馆” (the Palace Museum in Beijing) as an entity, due to lack of this word in the lexicon. In addition, it is impossible for the lexicon to contain all lexical matched words you want.

In order to solve this issue, this paper presents a MCSN model fusing dependency information of Chinese words. As shown in Fig. 2, we can use the dependency information of the word “北京” (Beijing). Obviously, the word “北京” (Beijing) in this sentence is a component that acts as attributive, as well as the word “故宫” (Imperial Palace). Because the word “博物馆” (Museum) is modified by both the words “北京” (Beijing) and “故宫” (Imperial Palace), we can build relationships among them by using their dependency information. With the relationships among these words, the word “北京故宫博物馆” (the

Palace Museum in Beijing) is seen as an entirety. Finally, the word “北京故宫博物馆” (the Palace Museum in Beijing) will be recognized as an entity.

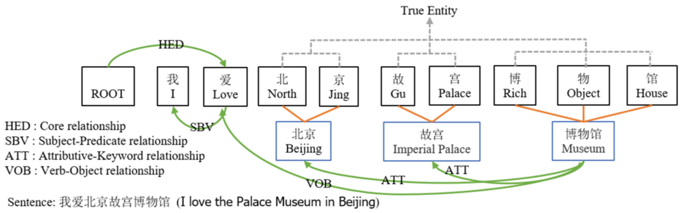

True Entity
HED
ROOT
我
I
爱
Love
北
North
京
Jing
故
Gu
宫
Palace
博
Rich
物
Object
馆
House
SBV
北京
Beijing
故宫
Imperial Palace
博物馆
Museum
ATT
ATT
VOB
HED : Core relationship
SBV : Subject-Predicate relationship
ATT : Attributive-Keyword relationship
VOB : Verb-Object relationship
Sentence: 我爱北京故宫博物馆 (I love the Palace Museum in Beijing)

Fig. 2. A sentence fusing the dependency relationship of Chinese words

In order to achieve our methods, we construct three word-character interactive graphs. Specifically, The first graph is the Relation graph, and its function is to build the dependency relationships among Chinese words. The second graph is the Containing graph, which is designed for capturing contextual information in a sentence. 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:

- We propose a multi-graph collaborative semantic network for Chinese NER tasks.
- To the best of our knowledge, we are the first neural approach to NER that models the dependency information of Chinese words with multi-graph structure.
- Experimental results show that our methods are not only effective, but also outperform the SOTA models.

2 Related Work

There are two main types of enhanced performances in Chinese NER.

Lexical Knowledge. In recent years, many studies focus on character-based model by leveraging lexical knowledge. A representative method is the Lattice-LSTM model proposed by Zhang et al. [24], which not only avoids error propagation, but also models characters and potential words simultaneously. Moreover, Transformer-based methods have been used with lexical enhancement. Flat-Lattice Transformer proposed by Li et al. [10] can convert the lattice structure into a flat structure consisting of spans. This method has an excellent parallelization ability. Based on Flat-Lattice Transformer, Wu et al. [20] presents a novel Multi-metadata Embedding to improve the performance of the NER model by considering structural information about Chinese characters.

Glyph-Structural Methods. Compared with traditional neural network methods, graph neural networks can consider the relations among characters better, which has been proven by many excellent methods [5,16]. A lexicon-based graph network is proposed by Gui et al. [5]. This method treats the named entities as a node classification task, which can avoid error propagation and leverage lexical knowledge. In addition, Ding et al. [3] proposed a neural multi-digraph model for Chinese NER with Gazetteers.

Although the enhanced methods of those two main types can obtain great experimental results in Chinese NER datasets, they neglect one of the most important points in Chinese sentences, the dependency information among Chinese words. Due to the shortness of the lexicon and the lack of dependency information among Chinese words, these models may not only recognize false entities, but also cause relatively poor results in complex Chinese text. In this work, we propose a MCSN model fusing the dependency information of Chinese words. With the dependency relationships of Chinese words, MCSN not only overcomes the shortages of the lexicon, but also better captures the semantic information of Chinese words. Experimental results show that our methods are effective.

3 Methodology

In this section, we first introduce the construction of three word-character graphs to integrate lexical knowledge and the dependency information of Chinese words into sentences, and then introduce the structure of our training model.

3.1 The Construction of Graphs

In order to achieve our methods, we construct three word-character interactive graphs. The vertices of three interactive graphs are different, and the edges of each graph are different. For this, we introduce an 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.

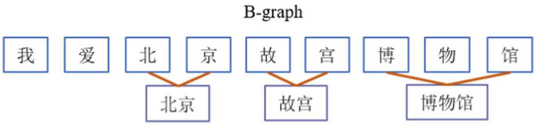


Fig. 3. The word-character Boundary graph

Boundary Graph. For the word-character Boundary graph (B-graph), the vertices of the B-graph consist of characters and lexical matched words of the corresponding character in a sentence. Take the sentence in Fig. 1 as an example. The sentence can be represented as $s = \{$我, 爱, 北, 京, 故, 宫, 博, 物, 馆$\}$. In order to utilize potential words in the sentence, we match all lexical words of every character. All matched words can be represented as $l = \{$北京, 故宫, 博物馆$\}$. Thus, the vertices of the B-graph is denoted as $V = \{$我, $\ldots$, 故宫, $\ldots\}$. As shown in Fig. 3, if a lexical matched word l_i contains many characters, we need to leverage its contained first character or last character c. The (l_i, c)-entry of the B-graph corresponding adjacency matrix A^B is assigned a value of 1.

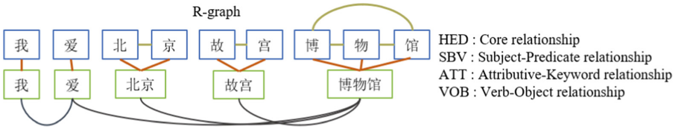

R-graph
我
爱
北
京
故
宫
博
物
馆
我
爱
北京
故宫
博物馆
HED : Core relationship
SBV : Subject-Predicate relationship
ATT : Attributive-Keyword relationship
VOB : Verb-Object relationship

Fig. 4. The word-character Relation graph

Relation Graph. Taking the same sentence as an example, the vertex set of the word-character Relation graph (R-graph) consists of characters and words separated by the dependency parsing tool[1]. The sentence can be represented as $s = \{$我, 爱, $\ldots$, 宫, 博, 物, 馆$\}$. The words separated by the dependency parsing tool, can be represented as $w = \{$我, 爱, 北京, 故宫, 博物馆$\}$. The dependency information of these words is as $f = \{\mathrm{SBV}, \mathrm{HED}, \mathrm{ATT}, \mathrm{ATT}, \mathrm{VOB}\}$, which can be replace as the set $index = \{2, 0, 5, 5, 2\}$ (the value in set $index$ represents that the current node has a connection with the value-th node). With this R-graph, we build not only connections among characters, but also the dependency relationships among separated words in a sentence. As shown in Fig. 4, if a separated word w_n contains character set $C = \{c_1, c_2, \ldots, c_p\}, c_i, c_j \in C$, we will assign the (w_n, c_i)-entry of the R-graph corresponding adjacency matrix A^R a value of 1, as well as the (c_i, c_j)-entry. Moreover, if a separated word w_n has a relation with another separated word w_m, the (w_n, w_m)-entry of the R-graph corresponding adjacency matrix A^R is assigned a value of 1.

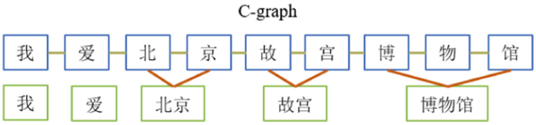


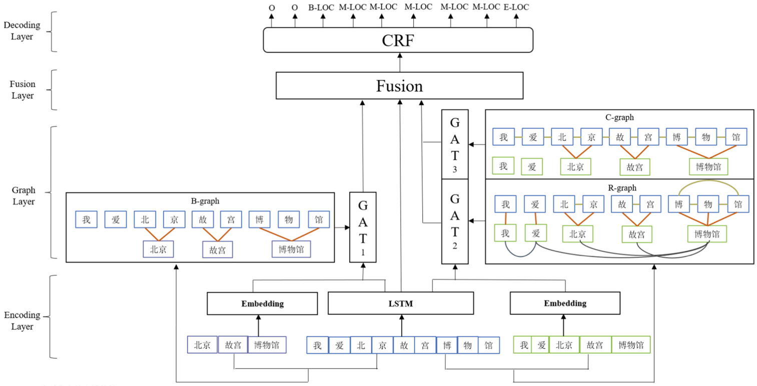
Fig. 5. The word-character Containing graph

Containing Graph. The vertices of the Containing graph (C-graph) consist of characters and words separated by the dependency parsing tool, as well as

[1] https://github.com/baidu/DDParser.

the R-graph. With this graph, the contextual information in the sentence can be captured by our model. As shown in Fig. 5, if a separated word w_n contains more than one character, we need to leverage its contained first character or last character i. Therefore, the (w_n, i)-entry of the C-graph corresponding adjacency matrix A^C is assigned a value of 1. Moreover, in the original sentence, if a character i has a predecessor or successor j, we will assign "$A^C_{ij} = 1$".

3.2 The Whole Architecture of Our Model



Sentence: 我爱北京故宫博物馆 (I love the Palace Museum in Beijing)
Matched word: 北京(Beijing), 故宫(Imperial Palace), 博物馆 (Museum)
Separated words: 我(I) , 爱(Love),北京(Beijing), 故宫(Imperial Palace), 博物馆 (Museum)

Fig. 6. The architecture of our model

The whole architecture of our model is shown in Fig. 6. For a sentence, we use the lexicon to obtain all matched words, and use the dependency parsing tool to obtain separated words. Then, those words are converted into dense vectors. Moreover, we first utilize a bidirectional LSTM model to capture contextual information of the input sequence, and then merge it with matched words and separated words, respectively. Furthermore, we fuse the merged results with three word-character interactive graphs in the graph layer, respectively. In order to fully use the merits of different graphs, a fusion layer is used for fusing these graphs. In the end, the results of the final predictions are obtained through the Conditional Random Field (CRF).

Encoding

For a Chinese NER model, the input of the training model based on characters is seen as $s = \{c_1, c_2, \cdots, c_n\}$, where c_i is the i-th character in a sentence. Each character c_i can be represented by looking up the embedding vector:

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

where e^c denotes the character embedding vector table.

Due to features of Long-Short Term Memory (LSTM) [7], LSTM can better capture contextual information of Chinese sentences, which has been empirically proven to be useful. In this paper, a bidirectional LSTM is applied to $\mathbf{x}^c = \{\mathbf{x}_1^c, \mathbf{x}_2^c, \cdots, \mathbf{x}_n^c\}$ to obtain the left-to-right and right-to-left LSTM hidden states. As shown in Eq. (2), the hidden vector representations of input sentence are denoted as $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h_n}\}$.

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

Lexical knowledge can augment the character representation to enhance performance of the NER model. All lexical words matched by each character in the sentence are denoted as $l = \{l_1, l_2, \cdots, l_m\}$. By looking up the pre-trained embedding lookup table, each lexical word l_i is represented as a dense vector.

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

where e^w is a lexical embedding looking table.

In order to use the dependency relationships of Chinese words, we can obtain the separated words $w = \{w_1, w_2, \cdots, w_k\}$, by using dependency parsing tool. By looking up word embedding from a pre-train word embedding matrix, each separated word w_i can be represented as a dense vector.

$$\mathbf{x}_i^w = e^w\left(w_i\right), \tag{4}$$

where e^w is a lexical embedding looking table.

Graph Attention Networks over These Graphs
To give a graph architecture, meaningful outputs can be produced by graph neural network, or node representation can be learned through graph neural network [15]. Compared with other architectures of graph networks, we believe that Graph Attention Network (GAT) [18] is the most suitable for Chinese NER tasks, because GAT can allow for assigning different importances to different nodes with a neighborhood. Therefore, three word-character interactive graphs are modeled by GAT. Specifically, in a M-layer GAT, the input representation of j-th layer consists of a set of node features, $\mathbf{NF}^j = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N\}$. In addition, an adjacency matrix $\mathbf{A}$ is needed, $\mathbf{f}_i \in \mathbb{R}^F, \mathbf{A} \in \mathbb{R}^{N \times N}$, 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).

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

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

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, $\mathbf{W}^k \in \mathbb{R}^{F' \times F}$. The single-layer feed-forward neural network is denoted as $\mathbf{a} \in \mathbb{R}^{2F'}$. Note that, KF' is as the dimension of the output $\mathbf{f}_i'$. In the end, we will keep the averaging in the last layer, and F' is the dimension of the final output features we need.

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

Specifically, three independent graph attention networks are built for modeling three different word-character interactive graphs. Due to the different vertices of these three graphs, we denote these three independent graph attention networks as GAT_1, GAT_2, and GAT_3, respectively. The B-graph can capture the boundary information of the entity by using lexical knowledge. Therefore, the vertex set of B-graph can be denoted as $X_B = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_n, \mathbf{x}_1^l, \mathbf{x}_2^l, \cdots, \mathbf{x}_m^l\}$. The output node features of GAT_1 model is denoted as $\mathbf{G}_1$.

$$\mathbf{G}_1 = GAT_1\left(X_B, A^B\right), \tag{8}$$

where $\mathbf{G}_1 \in \mathbb{R}^{F' \times (n+m)}$, n is the number of characters in the sentence, and m is the number of the lexical words matched by characters in the sentence. For the R-graph and the C-graph, these two graphs are shared the same vertex set $X = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_n, \mathbf{x}_1^w, \mathbf{x}_2^w, \cdots, \mathbf{x}_k^w\}$. The output node features of GAT_2 and GAT_3 model are denoted as $\mathbf{G}_2$ and $\mathbf{G}_3$, respectively.

$$\mathbf{G}_2 = GAT_2\left(X, A^R\right), \tag{9}$$

$$\mathbf{G}_3 = GAT_3\left(X, A^C\right), \tag{10}$$

where $\mathbf{G}_y \in \mathbb{R}^{F' \times (n+k)}, y \in \{2,3\}$, n is the number of characters in the sentence, and k is the number of the words separated by dependency parsing tool. For the output node feature $\mathbf{G}_i, i \in \{1,2,3\}$, we keep the first n columns of these matrices, since only character representations are used to decode labels.

$$\mathbf{Q}_i = \mathbf{G}_i[:, 0:n], i \in \{1,2,3\}. \tag{11}$$

Fusion

In order to use the merits of these interactive graphs, a fusion layer is adapted to fuse three graphs. Moreover, the contextual information of the input sentence is beneficial to Chinese NER. The input of the fusion layer is the contextual representation $\mathbf{H}$ and the output of the graph layer $\mathbf{Q_i}, i \in \{1,2,3\}$. The fusion equation is introduced below:

$$\mathbf{R} = \mathbf{W}_1\mathbf{H} + \mathbf{W}_2\mathbf{Q}_1 + \mathbf{W}_3\mathbf{Q}_2 + \mathbf{W}_4\mathbf{Q}_3, \tag{12}$$

where $\mathbf{W}_y, y \in \{1,2,3,4\}$, is a trainable matrice. In the end, a collaborative matrix $\mathbf{R}$ can be obtained from a fusion layer. The matrix $\mathbf{R}$ integrate different functions of these different graphs.

Decoding

A standard CRF [8] layer is adopted to capture the dependencies between successive labels. For any input sentence $s = \{c_1, c_2, \cdots, c_n\}$, we obtain the input representation of CRF layer $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_n\}$ from the fusion layer. Given the predicted tag sequence $y = \{y_1, y_2, \cdots, y_n\}$, the probability of the tag sequence y can be computed by:

$$p(y \mid s) = \frac{\exp\left(\sum_i \left(\mathbf{W}_{CRF}^{y_i}\mathbf{r}_i + \mathbf{b}_{CRF}^{(y_{i-1},y_i)}\right)\right)}{\sum_{y'} \exp\left(\sum_i \left(\mathbf{W}_{CRF}^{y'_i}\mathbf{r}_i + \mathbf{b}_{CRF}^{(y'_{i-1},y'_i)}\right)\right)}, \tag{13}$$

where y' denotes an arbitrary label sequence, $\mathbf{W}_{CRF}^{y_i}$ and $\mathbf{b}_{CRF}^{(y_{i-1},y_i)}$ are trainable parameters. The first-order Viterbi algorithm [19] 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.

$$L = \sum_{i=1}^{N} \log\left(P\left(y_i \mid s_i\right)\right) + \frac{\lambda}{2}\|\Theta\|^2. \tag{14}$$

As shown in Eq. 14, 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 metrics (P, R, F1) and so on. Specifically, our datasets include Weibo NER [13], OntoNotes [14] and E-commerce [3].

4.1 Overall Performance

Weibo and E-commerce. Chinese NER datasets on informal text are more challenging, due to the shortness and noisiness of the informal text. Compared with others, there are many informal texts in Weibo and E-commerce datasets, which may cause poor performance of the models. Moreover, the connections among entities are not close together. The R-graph can build closer relationships among characters by using the dependency information of Chinese words. Those close relationships are beneficial to the recognization of entities. The experimental results show that our methods are useful, as shown in Table 1. Compared with the MECT model [20], our model can outperform it by 9.34%, 1.82% and 3.33% in F1 score on Weibo-NM, Weibo-ALL and E-commerce, respectively.

Table 1. Main results on Weibo and E-commerce.

Models	Weibo-NE	Weibo-NM	Weibo-ALL	E-commerce
Lattice-LSTM [24]	53.04	62.25	58.79	–
CAN-NER [26]	55.38	62.98	59.31	–
LR-CNN [4]	57.14	66.67	59.92	–
LGN [5]	55.34	64.98	60.21	–
SoftLexicon (LSTM) [12]	59.08	62.22	61.42	73.59
+bichar [12]	58.12	64.20	59.81	73.88
MECT [20]	**61.91**	62.51	63.30	72.27
Ours	59.56	**71.85**	**65.12**	**75.60**

Table 2. Main results on OntoNotes.

Models	P	R	F1
Lattice-LSTM [24]	76.35	71.56	73.88
CAN-NER [26]	75.05	72.29	73.64
LR-CNN [4]	76.40	72.60	74.45
LGN [5]	76.13	73.68	74.89
SoftLexicon (LSTM) [12]	77.28	74.07	75.64
+bichar	77.13	75.22	76.16
MECT [20]	77.57	**76.27**	76.92
Ours	**79.89**	74.42	**77.06**

OntoNotes. There is a problem with the quality and consistency of the annotation in OntoNotes due to language ambiguity, which can lead to confusion in entity boundaries. The B-graph is capable of solving this problem. The results of the OntoNotes dataset indicate that our methods are effective. Detailed information about the main results can be known in Table 2. The MECT is proposed by Wu et al. [20], which is the SOTA model on OntoNotes dataset. Compared with the SOTA model, our methods gain 2.32%, 0.14% in Precision score and F1 score, respectively.

4.2 Effectiveness

The ablation experiments show the effectiveness of three interactive graphs.

Settings. The details of the ablation studies are as follows: 1) LSTM: we just use the LSTM model for training. 2) LSTM + B: we keep the LSTM and the B-graph for training. 3) LSTM + R + C: without the B-graph, etc.

Results. The results of the ablation study are shown in Table 3. We know that removing any graph can cause poor performance of the model in different

Table 3. Ablation study

Models	Weibo-NE	Weibo-NM	Weibo-ALL	Ontonotes
LSTM	46.51	54.98	52.14	61.89
LSTM + B	54.36	65.85	60.94	73.34
LSTM + R	55.77	66.56	61.78	74.86
LSTM + C	54.59	64.71	59.69	72.54
LSTM + B + R	57.89	70.01	63.36	76.10
LSTM + B + C	57.52	69.85	62.43	75.61
LSTM + R + C	56.97	67.38	62.16	75.96
Complete model	**59.56**	**71.85**	**65.12**	**77.06**

datasets. Specifically, the models with the R-graph can obtain better performances than others, which shows that the dependency information of Chinese words is beneficial to NER. However, the C-graph performs poorly without cooperating with other graphs. We guess that the contextual information of the sentence captured by "C-graph" is not enough to recognize the entities in informal text. 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 propose a Multi-graph Collaborative Semantic Network fusing the dependency information of Chinese words. The core of our model is three word-character interactive graphs. Specifically, The first graph is the Relation graph, and its function is to build the dependency relationships among Chinese words. The second graph is Containing graph, which is designed for capturing contextual information in a sentence. The third graph is the Boundary graph, which is designed for confirming the boundaries of named entities. With those three graphs, our model not only overcomes the shortages of lexicon, but also better captures the semantic information of Chinese words. Experimental results show that our methods are not only effective, but also outperform the SOTA results.

Acknowledgements. The works described in this paper are supported by The National Natural Science Foundation of China under Grant Nos. 61772210 and U1911201; Guangdong Province Universities Pearl River Scholar Funded Scheme (2018); The Project of Science and Technology in Guangzhou in China under Grant Nos.202007040006; Key Projects of the National Social Science Foundation of China under Grant Nos. 19ZDA041.

References

1. Chen, H., Yin, C., Fan, X., Qiao, L., Rong, W., Xiong, Z.: Learning path recommendation for MOOC platforms based on a knowledge graph. In: KSEM, pp. 600–611 (2021)
2. Cheng, D., Song, H., He, X., Xu, B.: Joint entity and relation extraction for long text. In: KSEM, pp. 152–162 (2021)
3. Ding, R., Xie, P., Zhang, X., Lu, W., Li, L., Si, L.: A neural multi-digraph model for Chinese NER with gazetteers. In: ACL, pp. 1462–1467 (2019)
4. 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)
5. Gui, T., et al.: A lexicon-based graph neural network for Chinese NER. In: EMNLP, pp. 1039–1049 (2019)
6. He, H., Sun, X.: F-score driven max margin neural network for named entity recognition in Chinese social media. In: EACL, pp. 713–718 (2017)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
8. 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)
9. Li, Q., Huang, Z., Dou, Y., Zhang, Z.: A framework of data augmentation while active learning for Chinese named entity recognition. In: KSEM, pp. 88–100 (2021)
10. Li, X., Yan, H., Qiu, X., Huang, X.: FLAT: Chinese NER using flat-lattice transformer. In: ACL, pp. 6836–6842 (2020)
11. Liu, P., Guo, Y., Wang, F., Li, G.: Chinese named entity recognition: the state of the art. Neurocomputing **473**, 37–53 (2022)
12. Ma, R., Peng, M., Zhang, Q., Wei, Z., Huang, X.: Simplify the usage of lexicon in Chinese NER. In: ACL, pp. 5951–5960 (2020)
13. Peng, N., Dredze, M.: Named entity recognition for Chinese social media with jointly trained embeddings. In: Proceedings of the EMNLP, pp. 548–554 (2015)
14. Pradhan, S., Ramshaw, L., Marcus, M., Palmer, M., Weischedel, R., Xue, N.: Conll-2011 shared task: modeling unrestricted coreference in ontonotes. In: Computational Natural Language Learning, pp. 1–27 (2011)
15. Qiu, H., Zheng, Q., Msahli, M., Memmi, G., Qiu, M., Lu, J.: Topological graph convolutional network-based urban traffic flow and density prediction. IEEE Trans. Intell. Transp. Syst. **22**(7), 4560–4569 (2021)
16. 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)
17. Tamine, L., Goeuriot, L.: Semantic information retrieval on medical texts: research challenges, survey, and open issues. ACM Comput. Surv. 146:1–146:38 (2022)
18. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018)
19. Viterbi, A.J.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans. Inf. Theory **13**(2), 260–269 (1967)
20. 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)

21. Yang, J., Teng, Z., Zhang, M., Zhang, Y.: Combining discrete and neural features for sequence labeling. In: International Conference on Intelligent Text Processing and Computational Linguistics, pp. 140–154 (2016)
22. Zhang, F., Li, R., Xu, K., Xu, H.: Similarity-based heterogeneous graph attention network for knowledge-enhanced recommendation. In: KSEM, pp. 488–499 (2021)
23. Zhang, Y., Gao, T., Lu, J., Cheng, Z., Xiao, G.: Adaptive entity alignment for cross-lingual knowledge graph. In: KSEM, pp. 474–487 (2021)
24. Zhang, Y., Yang, J.: Chinese NER using lattice LSTM. In: ACL, 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)