Leverage Lexical Knowledge for Chinese Named Entity Recognition via Collaborative Graph Network

Dianbo Sui[1,2], **Yubo Chen**[1], **Kang Liu**[1,2], **Jun Zhao**[1,2], **Shengping Liu**[3]
[1] National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing, 100190, China
[2] University of Chinese Academy of Sciences, Beijing, 100049, China
[3] Beijing Unisound Information Technology Co., Ltd, Beijing, 100028, China
{dianbo.sui, yubo.chen, kliu, jzhao}@nlpr.ia.ac.cn
liushengping@unisound.com

Abstract

The lack of word boundaries information has been seen as one of the main obstacles to develop a high performance Chinese named entity recognition (NER) system. Fortunately, the automatically constructed lexicon contains rich word boundaries information and word semantic information. However, integrating lexical knowledge in Chinese NER tasks still faces challenges when it comes to self-matched lexical words as well as the nearest contextual lexical words. We present a Collaborative Graph Network to solve these challenges. Experiments on various datasets show that our model not only outperforms the state-of-the-art (SOTA) results, but also achieves a speed that is six to fifteen times faster than that of the SOTA model.[1]

1 Introduction

Named entity recognition (NER) aims to locate and classify certain occurrences of words or expressions in unstructured text into predefined semantic categories such as the person names, locations, organizations, etc. NER is an essential pre-processing step for many natural language processing (NLP) applications, such as relation extraction (Bunescu and Mooney, 2005), event extraction (Chen et al., 2015), question answering (Mollá et al., 2006) etc. In English NER, LSTM-CRF models (Lample et al., 2016; Ma and Hovy, 2016; Chiu and Nichols, 2016; Liu et al., 2018) leveraging word-level representations and character-level representations achieve the state-of-the-art results.

In this paper, we focus on Chinese NER. Compared with English, Chinese has no obvious word boundaries. Since without word boundaries information, it is intuitive to use character information only for Chinese NER (He and Wang, 2008; Liu et al., 2010; Li et al., 2014), although such methods could result in the disregard of word information. However, word information is very useful in Chinese NER, because word boundaries are usually the same as named entity boundaries. For example, as shown in Figure 1, the boundaries of the word "北京机场" (Beijing airport) are the same as the boundaries of the named entity " 北京机场" (Beijing airport). Therefore, making full use of word information would help to improve the Chinese NER performance.

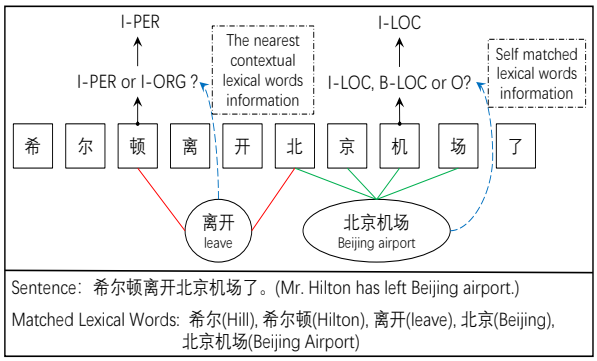

Figure 1: An example sentence integrating the nearest contextual lexical words (red line) and self-matched lexical words (green line)

There are three main ways to incorporate word information in NER. The first one is the pipeline method. The way of pipeline method is to apply Chinese Word Segmentation (CWS) first, and then to use a word-based NER model. However, the pipeline method suffers from error propagation, since the error of CWS may affect the performance of NER. The second one is to learn CWS and NER tasks jointly (Xu et al., 2013; Peng and Dredze, 2016; Cao et al., 2018; Wu et al., 2019). However, the joint models must rely on CWS annotation datasets, which are costly and are annotated under many diverse segmentation criteria (Chen

[1]The code is available at https://github.com/DianboWork/Graph4CNER

et al., 2017). The third one is to leverage an automatically constructed lexicon, which is pre-trained on large automatically segmented texts. Lexical knowledge includes boundaries and semantic information. Boundaries information is provided by the lexicon word itself, and semantic information is provided by pre-trained word embeddings (Bengio et al., 2003; Mikolov et al., 2013). Compared with joint methods, a lexicon is easy to obtain and additional annotation CWS datasets are not required. Recently, Zhang and Yang (2018) propose a lattice LSTM to integrate lexical knowledge in NER. However, integrating lexical knowledge into sentences still faces two challenges.

The first challenge is to integrate self-matched lexical words. A self-matched lexical word of a character is the lexical word that contains this character. For instance, "北京**机**场" (Beijing Airport) and "**机**场" (Airport) are the self-matched words of the character "**机**" (airplane). "离开" (leave) is not the self-matched word of the character "机" (airplane), since " 机" (airplane) is not contained in the word " 离开" (leave). The lexical knowledge of self-matched word is useful in Chinese NER. For example, as shown in Figure 1, the boundaries and semantic knowledge of the self-matched word "北京机场" (Beijing Airport) can help the character "机"(airplane) to predict an "I-LOC" tag, instead of "O" or "B-LOC" tags. However, due to the limits of the word-character lattice, the lattice LSTM (Zhang and Yang, 2018) fails to integrate the self-matched word "北京机场" (Beijing Airport) into the character "机" (airplane).

The second challenge is to integrate the nearest contextual lexical words directly. The nearest contextual lexical word of a character is the word that matches the nearest past or future subsequence in the given sentence of this character. For instance, the lexical word "离开" (leave) is the nearest contextual word of the character "顿" (-ton), since the word matches the nearest future subsequence "离开" of the character, while "北京" (Beijing) is not the nearest contextual lexical word of this character. The nearest contextual lexical words are beneficial for Chinese NER. For example, as shown in Figure 1, by directly using the semantic knowledge of the nearest contextual words "离开" (leave), an "I-PER" tag can be predicted instead of an "I-ORG" tag, since "希尔顿" (Hilton Hotels) cannot be taken as the subject of the verb "离开" (leave). However, a lattice model (Zhang and Yang, 2018) only implicitly integrate the knowledge of the nearest contextual lexical words via the previous hidden state. The information of the nearest contextual lexical word may be disturbed by other information.

To solve the above challenges, we propose a character-based Collaborative Graph Network, including an encoding layer, a graph layer, a fusion layer and a decoding layer. Specifically, there are three word-character interactive graphs in the graph layer. The first one is the Containing graph (C-graph), which is designed for integrating self-matched lexical words. It models the connection between characters and self-matched lexical words. The second one is the Transition graph (T-graph), which builds the direct connection between characters and the nearest contextual matched words. It helps to handle the challenge of integrating the nearest contextual words directly. The third one is the Lattice graph (L-graph), which is inspired by the lattice LSTM (Zhang and Yang, 2018). L-graph captures partial information of self-matched lexical words and the nearest contextual lexical words implicitly by multiple hops. These graphs are built without external NLP tools, which can avoid error propagation problem. Besides, these graphs complement each other nicely and a fusion layer is designed for collaboration between these graphs.

We test our model with various Chinese NER datasets. our model not only significantly outperforms the existing state-of-the-art (SOTA) model but also is six to fifteen times faster than the speed of the SOTA model.

In summary, our main contributions are as follows:

- We propose a Collaborative Graph Network to integrate lexical knowledge directly and efficiently for Chinese NER.
- To solve the challenges of integrating self-matched lexical words and the nearest contextual lexical words, we propose three word-character interactive graphs. These interactive graphs can capture different lexical knowledge and are built without external NLP tools.
- We achieve the state-of-the-art results in various popular Chinese NER datasets, and our model achieves a 6-15x speedup over the existing SOTA model.

2 Related Work

NER. There is rich literature on NER. This includes statistic methods, such as SVM (Isozaki and Kazawa, 2002), HMMs (Bikel et al., 1997) and CRF (Lafferty et al., 2001), suffering from feature engineering. There are also a number of recent neural network approaches applied to NER, such as (Collobert et al., 2011; Huang et al., 2015; Lample et al., 2016; Ma and Hovy, 2016; Chiu and Nichols, 2016; Liu et al., 2018; Akbik et al., 2018; Jie et al., 2019; Akbik et al., 2019). Compared with English, Chinese is not featured with obvious word boundaries, but it is important to leverage word boundaries and semantic information in Chinese NER. Many works use word segmentation information as extra features for Chinese NER, such as (Peng and Dredze, 2015; He and Sun, 2017a; Zhu and Wang, 2019). Peng and Dredze (2016), Cao et al. (2018) and Wu et al. (2019) propose joint models to train NER together with CWS. Our work is inspired by lattice LSTM (Zhang and Yang, 2018), which can integrate lexicon in NER.

Graph convolutional networks. There are a number of recent graph convolutional network (GCN) architectures (Kipf and Welling, 2017; Hamilton et al., 2017; Veličković et al., 2018; Qu et al., 2019) for learning over graphs. Our work is closely related to the graph attention networks (GAT), introduced by Veličković et al. (2018), leveraging masked self-attention layers to assign different importance to neighbouring nodes. In recent years, there is more and more literature about the application of GCN in NLP (Bastings et al., 2017; Marcheggiani and Titov, 2017; Zhang et al., 2018; Yao et al., 2019; Wang et al., 2018; Mishra et al., 2019; Cao et al., 2019; Zhang et al., 2019). Cetoli et al. (2017) use GCN to investigate the role of the dependency tree in English named entity recognition. However, most of the works (Bastings et al., 2017; Marcheggiani and Titov, 2017; Cetoli et al., 2017; Zhang et al., 2018) heavily rely on the dependency tree to construct a single graph, which suffer from error propagation. To capture different semantic and boundaries information, we propose a Collaborative Graph Network consisting of three automatically constructed graphs, which can avoid error propagation problem naturally. To our best knowledge, we are the first to introduce GAT and automatically constructed semantic graphs to Chinese NER tasks.

3 Approach

In this section, we first introduce the construction of graphs to integrate self-matched lexical words and the nearest contextual lexical words into sentences. We then introduce the architecture of Collaborative Graph Network as a core for solving Chinese NER tasks.

3.1 The Construction of Graphs

To integrate self-matched lexical words and the nearest contextual lexical words, we propose three word-character interactive graphs. The first is the word-character Containing graph (C-graph), which is to assist the character to capture the boundaries and semantic information of self-matched lexical words. The second is the word-character Transition graph (T-graph). The function of T-graph is to assist the character to capture the semantic information of the nearest contextual lexical words. The third is the Lattice graph (L-graph). Zhang and Yang (2018) propose a lattice structure, nested in the LSTM (Hochreiter and Schmidhuber, 1997), to integrate lexical knowledge. We free the lattice structure from the LSTM and adopt it as the third graph.

These three graphs share the same vertex set, but the edge sets of the three graphs are completely different. The vertex set is made up of the characters in the sentence and the matched lexical words, for example, as shown in Figure 1, the vertex set is V={希, 尔,..., 了, 希尔, 希尔顿, ..., 北京机场}. To represent the edge set, adjacency matrix needs to be introduced. The elements of the adjacency matrix indicate whether pairs of vertices are adjacent or not in the graph. Since the edge sets of the three graphs are totally different, the adjacency matrices of these three graphs are introduced below:

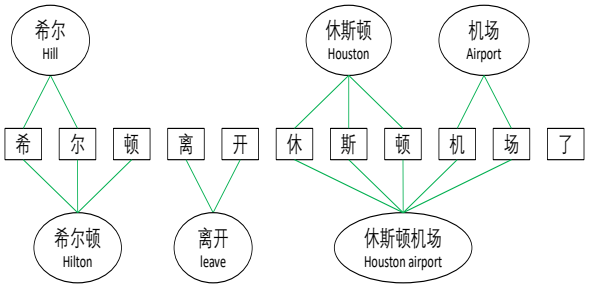

Figure 2: Word-Character Containing graph

Word-Character Containing graph

With the C-graph, the characters in the sentence can capture the boundaries and semantic information of self-matched lexical words. As shown in Figure 2, if a lexical word i contains a character j, the (i, j)-entry of the C-graph corresponding adjacency matrix $\mathbf{A}^C$ will be assigned a value of 1.

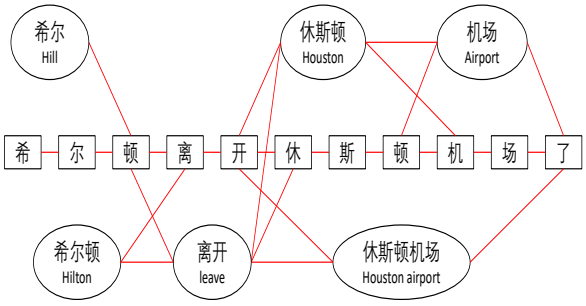

Figure 3: Word-Character Transition graph

Word-Character Transition graph

The T-graph is to assist the character to capture the semantic information of the nearest contextual lexical words. As shown in Figure 3, if a lexical word i or a character m matches the nearest preceding or following subsequence of a character j, the (i, j) or (m, j)-entry of the T-graph corresponding adjacency matrix $\mathbf{A}^T$ will be assigned a value of 1. Moreover, for capturing the context relation between lexical words, if a lexical word i is the preceding or following context of another lexical word k, we will assign "$\mathbf{A}^T_{ik} = 1$". Note that the T-graph is the same with the word cutting graph which is used in Chinese Word Segmentation.

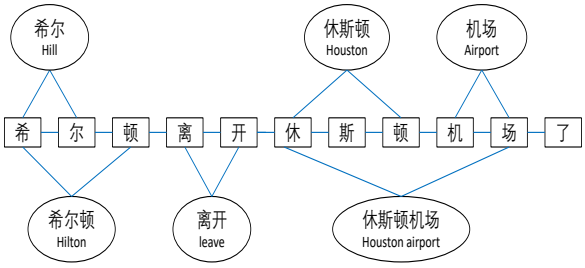

Figure 4: Word-Character Lattice graph

Word-Character Lattice graph

Zhang and Yang (2018) propose a lattice structure LSTM to exploit lexical knowledge for Chinese NER. A lattice structure can capture the information of the nearest contextual lexical words implicitly and capture some information of self-matched lexical words. As shown in Figure 4, if a character m is the nearest preceding or following character of a character j, the (m, j)-entry of the L-graph corresponding adjacency matrix $\mathbf{A}^L$ will be assigned a value of 1. Moreover, if a character j matches the lexical word i first character or end character, we will assign "$\mathbf{A}^L_{ij} = 1$".

3.2 Model

A character-based Collaborative Graph Network includes an encoding layer, a graph layer, a fusion layer, and a decoding layer. The encoding layer is to capture contextual information of the sentence and to represent the semantic information of lexical words. The graph layer is based on GAT (Veličković et al., 2018) for modeling over three word-character interactive graphs. A fusion layer is used for fusing different lexical knowledge captured by these three graphs. Finally, a standard CRF (Lafferty et al., 2001) model is used for decoding labels.

Encoding

The input of the model is a sentence and all lexical words that match consecutive subsequences of the sentence. We denote the sentence as $s = \{c_1, c_2, ..., c_n\}$, where c_i is the i-th character, and denote the matched lexical words as $l = \{l_1, l_2, ..., l_m\}$. By looking up the embedding vector from a pre-train character embedding matrix, each character c_i is represented as a vector, which denotes as $\mathbf{x}_i$.

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

e^c is a character embedding lookup table.

To capture contextual information, A bidirectional LSTM (Hochreiter and Schmidhuber, 1997) is applied to $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$. By concatenating the left-to-right and right-to-left LSTM hidden states, we obtain the contextual representation $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n\}$.

$$\overrightarrow{\mathbf{h}_i} = \overrightarrow{LSTM}(\mathbf{x}_i, \overrightarrow{\mathbf{h}_{i-1}}) \tag{2}$$

$$\overleftarrow{\mathbf{h}_i} = \overleftarrow{LSTM}(\mathbf{x}_i, \overleftarrow{\mathbf{h}_{i+1}}) \tag{3}$$

$$\mathbf{h}_i = \overrightarrow{\mathbf{h}_i} \oplus \overleftarrow{\mathbf{h}_i} \tag{4}$$

To represent the semantic information of lexical words, we look up word embeddings from a

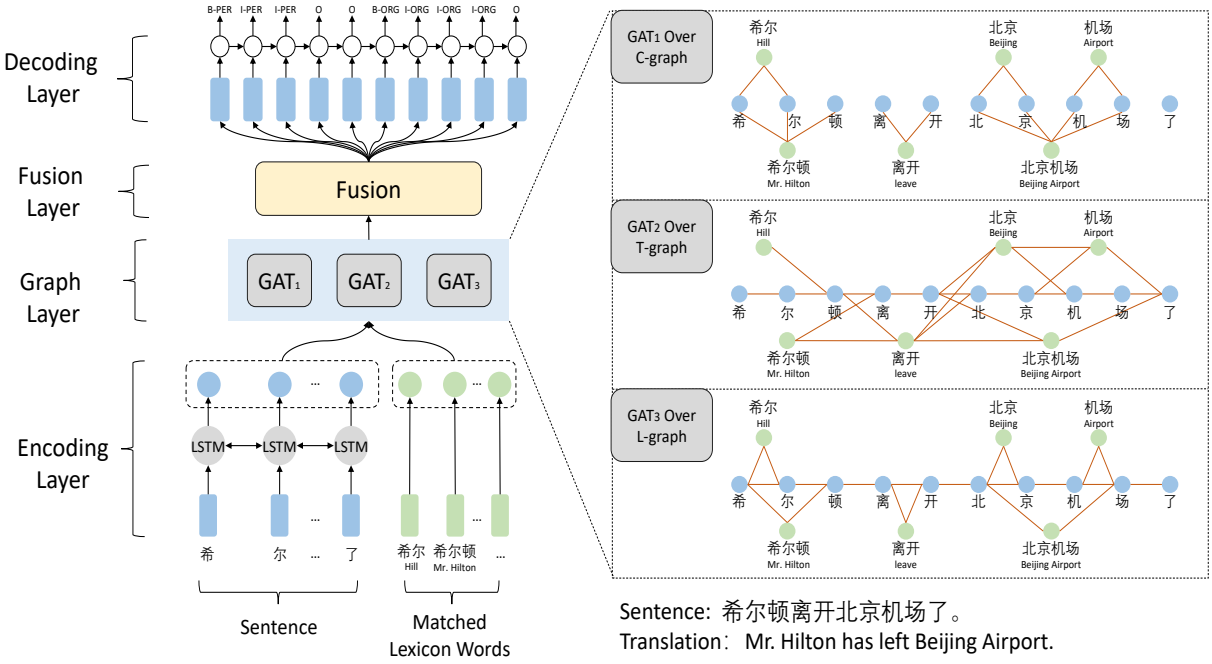

Figure 5: Main architecture of a Collaborative Graph Network for integrating lexical knowledge in Chinese NER. The left side shows the overall architecture, including an encoding layer, a graph layer, a fusion layer, and a decoding layer. On the right side, we show the details of graph attention networks over three word-character interactive graphs. We use blue to denote the characters in the sentence and use green to denote the matched lexicon words.

pre-train word embedding matrix, and each lexical words l_i is represented as a semantic vector, which denotes as $\mathbf{wv}_i$.

$$\mathbf{wv}_i = e^w(l_i) \tag{5}$$

e^w is a word embedding lookup table. We concatenate the contextual representation and the word embeddings as the output of this layer, denoting it as $\mathbf{Node}_f$.

$$\mathbf{Node}_f = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n, \mathbf{wv}_1, \mathbf{wv}_2, ..., \mathbf{wv}_m] \tag{6}$$

Graph Attention Networks over Word-Character Interactive Graphs

We use Graph Attention Networks (GAT) to model over three interactive graphs. In an M-layer GAT, the input of j-th layer is a set of node features, $\mathbf{NF}^j = \{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_N\}$, together with an adjacency matrix $\mathbf{A}$, $\mathbf{f}_i \in \mathbb{R}^F$, $\mathbf{A} \in \mathbb{R}^{N \times N}$, where N denotes the number of the nodes and F is the the dimension of features at j-th layer. The output of j-th layer is a new set of node features, $\mathbf{NF}^{(j+1)} = \{\mathbf{f}'_1, \mathbf{f}'_2, ..., \mathbf{f}'_N\}$. A GAT operation with K independent attention head can be written as :

$$\mathbf{f}'_i = \Big\Vert_{k=1}^{K} \sigma\Big(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \mathbf{f}_j\Big) \tag{7}$$

$$\alpha_{ij}^k = \frac{\exp(\mathrm{LeakyReLU}(\mathbf{a}^{\mathrm{T}}[\mathbf{W}^{\mathrm{k}}\mathbf{f}_{\mathrm{i}} \| \mathbf{W}^{\mathrm{K}}\mathbf{f}_{\mathrm{j}}]))}{\Sigma_{k \in \mathcal{N}_i} \exp(\mathrm{LeakyReLU}(\mathbf{a}^{\mathrm{T}}[\mathbf{W}^{\mathrm{k}}\mathbf{f}_{\mathrm{i}} \| \mathbf{W}^{\mathrm{K}}\mathbf{f}_{\mathrm{k}}]))} \tag{8}$$

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

$$\mathbf{f}_i^{final} = \sigma\Big(\frac{1}{K} \sum_{k=1}^{K} \sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \mathbf{f}_j\Big) \tag{9}$$

To model three totally different word-character interactive graphs, We build three independent graph attention networks, which are denoted as GAT_1, GAT_2, and GAT_3. Since three word-character interactive graphs share the same vertex set, the input node features of all GAT are matrix $\mathbf{Node}_f$, which is shown in Equation 6. The output node features are denoted as $\mathbf{G}_1$, $\mathbf{G}_2$ and $\mathbf{G}_3$,

$$\mathbf{G}_1 = GAT_1(\mathbf{Node}_f, A^C) \tag{10}$$

$$\mathbf{G}_2 = GAT_2(\mathbf{Node}_f, A^T) \tag{11}$$

$$\mathbf{G}_3 = GAT_3(\mathbf{Node}_f, A^L) \tag{12}$$

Extra Resource	Models	Named Entity			Named Mention			Overall
		P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	F1(%)
Automatic word seg	Peng and Dredze (2015)	74.78	39.81	51.96	71.92	53.03	61.05	56.05
Word Seg Data	Peng and Dredze (2016)	66.67	47.22	55.28	74.48	54.55	62.97	58.99
Automatic word seg	He and Sun (2017a)	66.93	40.67	50.60	66.46	53.57	59.32	54.82
Other data	He and Sun (2017b)	61.68	48.82	54.50	74.13	53.54	62.17	58.32
Word Seg Data	Cao et al. (2018)	59.51	50.00	54.43	71.43	47.90	57.53	58.70
Automatic word seg	Zhu and Wang (2019)	-	-	55.38	-	-	62.98	59.31
Lexicon	Zhang and Yang (2018)	-	-	53.04	-	-	62.25	58.79
Lexicon	**Ours**	**67.31**	**48.61**	**56.45**	**75.15**	**62.63**	**68.32**	**63.09**

Table 1: Main results on Weibo NER

where $\mathbf{G}_k \in \mathbb{R}^{F' \times (n+m)}$, $k \in \{1, 2, 3\}$. We keep the first n columns of these matrices and discard the last m columns, because only character representations are used to decode labels.

$$\mathbf{Q}_k = \mathbf{G}_k[:, 0{:}n],\ k \in \{1, 2, 3\} \quad (13)$$

Fusion Layer

A fusion layer is used to fuse different lexical knowledge captured by word-character interactive graphs. 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 equation of the fusion layer is introduced below:

$$\mathbf{R} = \mathbf{W}_1\mathbf{H} + \mathbf{W}_2\mathbf{Q}_1 + \mathbf{W}_3\mathbf{Q}_2 + \mathbf{W}_4\mathbf{Q}_3 \quad (14)$$

where $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{W}_3$ and $\mathbf{W}_4$ are trainable matrices. Via a fusion layer, we obtain a matrix $\mathbf{R}$, $\mathbf{R} \in \mathbb{R}^{F' \times n}$, which is a new sentence representation integrating the contextual information as well as the lexical knowledge of self-matched lexical words and the nearest contextual lexical words.

Decoding and Training

We use a standard CRF (Lafferty et al., 2001) layer to capture the dependencies between successive labels. Given a sentence $s = \{c_1, c_2, ..., c_n\}$, the input of the CRF layer is $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_n\}$, and the probability of the ground-truth tag sequence $y = \{y_1, y_2, ..., y_n\}$ is

$$p(y|s) = \frac{\exp(\sum_i (\mathbf{W}^{y_i}\mathbf{r}_i + \mathbf{T}_{(y_{i-1}, y_i)}))}{\sum_{y'} \exp(\sum_i (\mathbf{W}^{y'_i} r_i + \mathbf{T}_{(y'_{i-1}, y'_i)}))} \quad (15)$$

Here y' is an arbitrary label sequence, $\mathbf{W}^{y_i}$ is used for modeling emission potential for the i-th character in the sentence, and $\mathbf{T}$ is the transition matrix storing the score of transferring from one tag to another. Viterbi algorithm (Viterbi, 1967) is used to get the label sequence with the highest score. Given a manually annotated training data $\{(s_i, y_i)\}|_{i=1}^N$, we optimize the model by minimizing the negative log-likelihood loss with L_2 regularization. The loss function is defined as:

$$L = -\sum_{i=1}^{N} log(P(y_i|s_i)) + \frac{\lambda}{2}\|\Theta\|^2 \quad (16)$$

where λ denotes the L_2 regularization parameter and Θ is the all trainable parameters set

Resource	Models	P(%)	R(%)	F1(%)
Gold Seg	Che et al. (2013)	77.71	72.51	75.02
	Wang et al. (2013)	76.43	72.32	74.32
	Yang et al. (2016)	65.59	71.84	68.57
	Yang et al. (2016)	72.98	80.15	76.40
	Zhu and Wang (2019)	75.05	72.29	73.64
Lexicon	Zhang and Yang (2018)	76.35	71.56	73.88
	Ours	**75.06**	**74.52**	**74.79**

Table 2: Main results on OntoNotes. Gold seg means gold-standard segmentation, which is not available in the real world.

4 Experiments

In this section, we carry out extensive experiments to investigate the effectiveness of the Collaborative Graph Network.

4.1 Datasets

We evaluate our model on Weibo NER (Peng and Dredze, 2015; He and Sun, 2017a), OntoNotes 4 (Weischedel et al., 2011), and MSRA (Levow, 2006), where Weibo NER is in social domain, OntoNotes and MSRA are in the news domain. On Weibo NER, we use the same training, development and test split as Peng and Dredze (2015). On OntoNotes, we use the same data split as Che et al. (2013). Since the MSRA dataset does not have a development set, we randomly select 10% samples from the training set as the development set.

Models	P(%)	R(%)	F1(%)
Chen et al. (2006)	91.22	81.71	86.20
Zhang et al. (2006)	92.20	90.18	91.18
Zhou et al. (2013)	91.86	88.75	90.28
Lu et al. (2016)	-	-	87.94
Dong et al. (2016)	91.28	90.62	90.95
Cao et al. (2018)	91.73	89.58	90.64
Zhu and Wang (2019)	93.53	92.42	92.97
Zhang and Yang (2018)	93.57	92.79	93.18
Ours	**94.01**	**92.93**	**93.47**

Table 3: Main results on MSRA

	Dataset	Ours(s)	Lattice(s)	Speedup
Training	MSRA	344	13723	×**15**
	OntoNotes	188	2561	×**13**
	Weibo NER	64	458	×**7**
Testing	MSRA	52	344	×**6**
	OntoNotes	27.1	386	×**14**
	Weibo NER	2.2	23	×**10**

Table 4: The performance of models in training and testing time. Time is measured in seconds. Lattice means the lattice LSTM (Zhang and Yang, 2018).

4.2 Experimental Settings

In our experiments, We use the same character embeddings as Zhang and Yang (2018), which is pretrained on Chinese Giga-Word. We use the lexicon provided by Li et al. (2018), including 1.3 million Chinese words. We set the dimensionality of LSTM hidden states to 300 and set the initial learning rate to 0.001. Since the scale of each dataset varies, we set different training batch size for different datasets. Specifically, we set batch sizes of MSRA, OntoNotes and Weibo NER as 64, 20 and 10. We use stochastic gradient Descent (SGD) algorithm to optimize parameters in OntoNotes and WeiboNER, and use Adam (Kingma and Ba, 2014) algorithm to optimize parameters in MSRA. We stop the training when we find the best result in the development set.

4.3 Overall Performance

Weibo NER. Table 1 shows the results on Weibo NER. Zhu and Wang (2019) propose a Convolutional Attention Network using segmentation information, which is the existing state-of-the-art (SOTA) model. Our model outperforms SOTA model by 3.78%, 1.07% and 5.34% in F1 score on Overall, Named Entity, and Nominal Mention. Zhang and Yang (2018) propose a lattice LSTM to integrate lexical knowledge. Our model outperforms the lattice LSTM by 4.3%, 3.41% and 6.07% in F1 score on Overall, Named Entity, and Nominal Mention.

OntoNotes. Table 2 shows the results on OntoNotes. Compared with lattice LSTM (Zhang and Yang, 2018), Our model gains a 0.91% improvement in F1 score. Compared with the best result (Yang et al., 2016), our model doesn't rely on gold-standard segmentation, which is not available in the real world. Note that our model even outperforms the model proposed by (Wang et al., 2013; Yang et al., 2016; Zhu and Wang, 2019), which uses the information of gold-standard segmentation.

MSRA. Results on the MSRA dataset are shown in Table 3. By leveraging hand crafted features (Chen et al., 2006; Zhang et al., 2006; Zhou et al., 2013) and character embeddings (Lu et al., 2016), statistical models achieve good results on MSRA dataset. Dong et al. (2016) integrate LSTM-CRF with radical features and Zhang and Yang (2018) propose a lattice LSTM to integrate lexical knowledge. Our model outperforms the lattice LSTM by 0.29% in F1 score on MSRA datasets.

Speed. As an essential preprocessing NLP tool, NER tasks require high speeds of both training and testing. Since aligning word-character lattice structure for batch training is usually non-trivial, the lattice LSTM (Zhang and Yang, 2018) suffers from slow speeds in training and testing. However, both LSTM and GAT in our model can compute efficiently by batch training.

For fair comparison, both the lattice LSTM and our model are implemented under PyTorch[2]. By using a single NVIDIA GeForce GTX 1080 Ti GPU, We randomly select 10 training and testing epoch as samples. The average time of training and testing is shown in Table 4. Our model can achieve a 6-15x speedup over the lattice LSTM.

4.4 Effectiveness of Three Word-Character Interactive Graphs

We conduct ablation experiments to demonstrate the effectiveness of these three word-character interactive graphs.

Comparison Setting. We design ablation studies as follow: 1) w/o C: without word-character Containing graph(C-graph). 2) w/o T: without word-character Transition graph (T-graph). 3) w/o L: without word-character Lattice graph (L-graph). 4)w/o C & T: without C-graph and T-graph, only keep L-graph. 5)w/o C & L : without C-graph and L-graph, only keep T-graph. 6) w/o

[2] https://pytorch.org/

Case1	Sentence	…//@西安电子科技大学:#早安… …// @Xidian University:#good morning…	Case2	Sentence	腾讯联想联合发起电脑清理日… Tencent and Lenovo jointly launched a computer cleaning day…
	Matched lexical word	… 西安电子科技大学(XidianUniversity)，西安(Xi’ an)， 电子科技大学(UESTC)，安电(An Dian)，电子科技(Electronics Technology)， 早安(Good Morning)…		Matched lexical word	腾讯(Tencent), 联想(Lenovo), 联合(Joint), 发起(Launch), 电脑(Computer), 清理(Clean)…
	Sentence with gold label	…// (O)@(O)西(B-ORG) 安(I-ORG) 电(I-ORG) 子(I-ORG) 科(I-ORG)技(I-ORG) 大(I-ORG)学(I-ORG):(O)#(O) 早(O)安(O) …		Sentence with gold label	腾(B-ORG)讯(I-ORG)联(B-ORG) 想(I-ORG) 联(O)合(O) 发(O)起(O)电(O)脑(O)清(O)理(O)日(O)…
	w/o C-graph predicted label	…// (O)@(O) 西(B-LOC) 安(I-LOC) 电(B-ORG) 子(I-ORG) 科(I-ORG)技(I-ORG) 大(I-ORG) 学(I-ORG):(O)#(O)早(O)安(O) …		w/o T-graph predicted label	腾(B-ORG)讯(I-ORG)联(O)想(O)联(O)合(O) 发(O) 起(O)电(O)脑(O)清(O)理(O)日(O)…
	with C-graph predicted label	…// (O)@(O)西(B-ORG) 安(I-ORG) 电(I-ORG) 子(I-ORG) 科(I-ORG)技(I-ORG) 大(I-ORG)学(I-ORG):(O)#(O)早(O)安(O)…		with T-graph predicted label	腾(B-ORG)讯(I-ORG)联(B-ORG) 想(I-ORG) 联(O) 合(O) 发(O) 起(O)电(O)脑(O)清(O) 理(O)日(O)…

Table 6: Case study. w/o C-graph predicted label means without C-graph predicted label, and w/o T-graph predicted label means without T-graph predicted label. We use green to denote the correct labels and use red to denote the wrong labels.

Models	Dataset		
	OntoNotes	Weibo NER	MSRA
Complete model	**74.79**	**63.09**	**93.47**
w/o C	72.24	60.75	93.35
w/o T	71.57	60.94	93.02
w/o L	72.87	60.69	93.21
w/o C & T	70.53	58.51	92.72
w/o C & L	65.81	58.65	91.98
w/o T & L	71.41	58.72	92.80
BiLSTM+CRF	61.84	52.77	88.05

Table 5: Ablation study on reducing word-character interactive graphs, For example, ”w/o C” means removing word-character containing graph from the complete model.

T & L : without T-graph and L-graph, only keep L-graph. 7) BiLSTM+CRF: baseline model.

Comparison Results. Table 5 shows the results of ablation experiments. We can clearly see that removing any graph causes obvious performance degradation, but the importance of different graphs varies from dataset to dataset. Specifically, on OntoNotes and MSRA, ’w/o T-graph’ obtains worse performance than others, showing that T-graph is important. However, T-graph performs poorly without cooperating with other graphs. We guess that “T-graph” graph can only capture the information of the nearest contextual lexical words, and it is not enough to rely solely on T-graph. On Weibo NER, these graphs show equal importance. Since dialects slangs and irregular phrases are very common in social domain, we must rely on C-graph, T-graph, and L-graph jointly to handle the informal and complex contexts. In conclusion, from ablation experiments, we can find that each graph can be implemented independent of the other, but together they can achieve the best result, showing that all these three graphs are essential to our model.

5 Case Study

To show visually that our model can solve the challenges when integrating self-matched lexical words and the nearest contextual lexical words, a case study comparing without C-graph, without T-graph and the complete model is shown in Table 6. In the first case, there is an entity “ 西安电子科技大学”(Xidian University) with nested “西安” (Xi’an) and “ 电子科技大学” (UESTC). These common entities are all in the lexicon. Without C-graph, the model can’t integrate the information of the self-matched lexical word ‘西安电子科技大学” (Xidian University) into the characters “ 电” and “安”. Influenced by another lexical word “ 电子科技大学” (UESTC), the predicted label of the character “ 电” is “B-ORG”, and the label of the character “安” is predicted to be “I-ORG”, affect by the lexical word “西安” (Xi’an). In the second case, there is an entity “ 联想” (Lenovo), which can also be a common verb (“Associate”) in Chinese. Without T-graph, the model can’t integrate the information of the nearest contextual lexical words “ 腾讯” (Tencent) and “联合” (Joint) into the characters “ 联” and “ 想”, so the predicted labels of the characters “联” and “ 想” are ”O”s. However, with the help of T-graph, the model can use the information of the nearest contextual lexical words “腾讯” (Tencent) and “ 联合” (Joint) to predict the correct labels.

6 Conclusion

In this paper, we propose a Collaborative Graph Network for integrating lexical knowledge in Chinese NER. The core of the network is three lexical word-character interactive graphs. These interactive graphs can capture different lexical knowledge and are built without external NLP tools. We show through various experiments that our model has complementary strengths to the SOTA model and these interactive graphs are effective.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.61533018), the Natural Key R&D Program of China (No.2017YFB1002101), the National Natural Science Foundation of China (No.61806201) and the independent research project of National Laboratory of Pattern Recognition. This work is also supported by the CCF-Tencent Open Research Fund. We thank the anonymous reviewers for their insightful comments. We also thank Xiangrong Zeng, Pengfei Cao and Yushan Xie for helpful comments and suggestions.

References

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 724–728.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018: The 27th International Conference on Computational Linguistics*, pages 1638–1649.

Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Simaan. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. 1997. Nymble: a high-performance learning name-finder. In *Fifth Conference on Applied Natural Language Processing*.

Razvan C Bunescu and Raymond J Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 724–731.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2018. Adversarial transfer learning for Chinese named entity recognition with self-attention mechanism. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 182–192.

Yu Cao, Meng Fang, and Dacheng Tao. 2019. BAG: Bi-directional attention entity graph convolutional network for multi-hop reasoning question answering. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 357–362.

Alberto Cetoli, Stefano Bragaglia, Andrew O'Harney, and Marc Sloan. 2017. Graph convolutional networks for named entity recognition. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, pages 37–45.

Wanxiang Che, Mengqiu Wang, Christopher D. Manning, and Ting Liu. 2013. Named entity recognition with bilingual constraints. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 52–62.

Aitao Chen, Fuchun Peng, Roy Shan, and Gordon Sun. 2006. Chinese named entity recognition with conditional probabilistic models. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 173–176.

Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-criteria learning for Chinese word segmentation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1193–1203.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537.

Chuanhai Dong, Jiajun Zhang, Chengqing Zong, Masanori Hattori, and Hui Di. 2016. Character-based lstm-crf with radical-level features for chinese named entity recognition. In *Natural Language Understanding and Intelligent Applications*, pages 239–250.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034.

Hangfeng He and Xu Sun. 2017a. F-score driven max margin neural network for named entity recognition in Chinese social media. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 713–718.

Hangfeng He and Xu Sun. 2017b. A unified model for cross-domain and semi-supervised named entity recognition in chinese social media. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Jingzhou He and Houfeng Wang. 2008. Chinese named entity recognition and word segmentation based on character. In *Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Hideki Isozaki and Hideto Kazawa. 2002. Efficient support vector classifiers for named entity recognition. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Zhanming Jie, Pengjun Xie, Wei Lu, Ruixue Ding, and Linlin Li. 2019. Better modeling of incomplete annotations for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 729–734.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270.

Gina-Anne Levow. 2006. The third international chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 108–117.

Haibo Li, Masato Hagiwara, Qi Li, and Heng Ji. Ji. 2014. Comparison of the impact of word segmentation on name tagging for chinese and japanese. In *LREC*, page 2532 ¨ C2536.

Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. 2018. Analogical reasoning on chinese morphological and semantic relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 138–143.

Liyuan Liu, Jingbo Shang, Frank Xu, Xiang Ren, Jian Gui, Huan Peng, and Jiawei. Han. 2018. Empower sequence labeling with task-aware neural language model. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zhangxun Liu, Conghui Zhu, and Tieju Zhao. 2010. 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*, page 634 ¨ C640.

Yanan Lu, Yue Zhang, and Donghong Ji. 2016. Multi-prototype Chinese character embedding. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 855–859.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2019. Abusive Language Detection with Graph Convolutional Networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2145–2150.

Diego Mollá, Menno van Zaanen, and Daniel Smith. 2006. Named entity recognition for question answering. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 51–58, Sydney, Australia.

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for Chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 548–554.

Nanyun Peng and Mark Dredze. 2016. Improving named entity recognition for Chinese social media with word segmentation representation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 149–155.

Meng Qu, Yoshua Bengio, and Jian Tang. 2019. Gmnn: Graph markov neural networks. In *International Conference on Machine Learning*, pages 5241–5250.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations*.

Andrew Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269.

Mengqiu Wang, Wanxiang Che, and Christopher D Manning. 2013. Effective bilingual constraints for semi-supervised learning of named entity recognizers. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 349–357.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, et al. 2011. Ontonotes release 4.0. *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium*.

Fangzhao Wu, Junxin Liu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2019. Neural chinese named entity recognition via cnn-lstm-crf and joint training with word segmentation. In *The World Wide Web Conference*, WWW '19, pages 3342–3348.

Yan Xu, Yining Wang, Tianren Liu, Jiahua Liu, Yubo Fan, Yi Qian, Junichi Tsujii, and Eric I Chang. 2013. Joint segmentation and named entity recognition using dual decomposition in chinese discharge summaries. *Journal of the American Medical Informatics Association*, 21(e1):e84–e92.

Jie Yang, Zhiyang Teng, Meishan Zhang, and Yue Zhang. 2016. Combining discrete and neural features for sequence labeling. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 140–154.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Thirty-Third AAAI Conference on Artificial Intelligence*.

Ningyu Zhang, Shumin Deng, Zhanlin Sun, Guanying Wang, Xi Chen, Wei Zhang, and Huajun Chen. 2019. Long-tail relation extraction via knowledge graph embeddings and graph convolution networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3016–3025.

Suxiang Zhang, Ying Qin, Juan Wen, and Xiaojie Wang. 2006. Word segmentation and named entity recognition for sighan bakeoff3. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 158–161.

Yue Zhang and Jie Yang. 2018. Chinese NER using lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1554–1564.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215.

Junsheng Zhou, Weiguang Qu, and Fen Zhang. 2013. Chinese named entity recognition via joint identification and categorization. *Chinese journal of electronics*, 22(2):225–230.

Yuying Zhu and Guoxin Wang. 2019. CAN-NER: Convolutional Attention Network for Chinese Named Entity Recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3384–3393.