A Lexicon-Based Graph Neural Network for Chinese NER

Tao Gui[*], **Yicheng Zou**[*], **Qi Zhang,**
Minlong Peng, Jinlan Fu, Zhongyu Wei, Xuanjing Huang
Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
School of Computer Science, Fudan University
825 Zhangheng Road, Shanghai, China
{tgui16,yczou18,qz,mlpeng16,fujl16,zywei,xjhuang}@fudan.edu.cn

Abstract

Recurrent neural networks (RNN) used for Chinese named entity recognition (NER) that sequentially track character and word information have achieved great success. However, the characteristic of chain structure and the lack of global semantics determine that RNN-based models are vulnerable to word ambiguities. In this work, we try to alleviate this problem by introducing a lexicon-based graph neural network with global semantics, in which lexicon knowledge is used to connect characters to capture the local composition, while a global relay node can capture global sentence semantics and long-range dependency. Based on the multiple graph-based interactions among characters, potential words, and the whole-sentence semantics, word ambiguities can be effectively tackled. Experiments on four NER datasets show that the proposed model achieves significant improvements against other baseline models.

1 Introduction

The task of named entity recognition (NER) involves determining entity boundaries and recognizing categories of named entities, which is a fundamental task in the field of natural language processing (NLP). NER plays an important role in many downstream NLP tasks, including information retrieval (Chen et al., 2015b), relation extraction (Bunescu and Mooney, 2005), question answering systems (Diefenbach et al., 2018), and other applications. Compared with English NER, Chinese named entities are more difficult to identify due to their uncertain boundaries, complex composition, and NE definitions within the nest (Duan and Zheng, 2011).

One intuitive way to alleviate word boundary problems is to first perform word segmentation and then apply word sequence labeling (Yang et al., 2016; He and Sun, 2017). However, the rare gold-standard segmentation in NER datasets and incorrectly segmented entity boundaries both negatively impact the identification of named entities (Peng and Dredze, 2015; He and Sun, 2016). Hence, character-level Chinese NER using lexicon features to better leverage word information has attracted research attention (Passos et al., 2014; Zhang and Yang, 2018). In particular, Zhang and Yang (2018) introduced a variant of a long short-term memory network (lattice-structured LSTM) that encodes all potential words matching a sentence to exploit explicit word information, achieving state-of-the-art results.

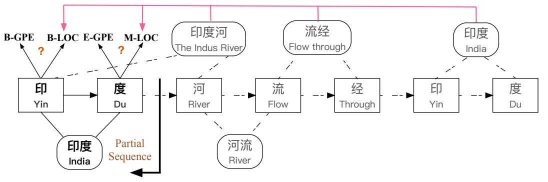


Figure 1: Example of word character lattice with partial input. Because of the characteristic of chain structure, RNN-based methods must predict the label “度” using only previous partial sequences “印度 (India)”, which may suffer from word ambiguities without global sentence semantics.

However, these methods are usually based on RNN or CRF to sequentially encode a sentence, while the underlying structure of language is not strictly sequential (Shen et al., 2019). As a result, these models would encounter serious word ambiguity problems (Mich et al., 2000). Especially in Chinese texts, the recognition of named entities with overlapping ambiguous strings is even more challenging. As shown in Figure 1, the middle character of an overlapping ambiguous string can constitute words with the characters to both their left and their right (Yen et al., 2012), such as “河

[*]Equal contribution.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing
and the 9th International Joint Conference on Natural Language Processing, pages 1039–1049,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

流 (River)” and “流经 (Flow through)”, which share a common character “流”. However, RNN-based models process characters in a strictly serial order, which is similar to reading Chinese, and a character has priority in being assigned to the word on the left (Perfetti and Tan, 1999). More seriously, RNN-based models must give the label of “度” using only previous partial sequences “印度 (India)”, which is problematic without seeing the remaining characters. Hence, Ma et al. (2014) suggested that the overlapping ambiguity must be settled using sentence context and high-level information.

In this work, we introduce a lexicon-based graph neural network (LGN) that achieves Chinese NER as a node classification task. The proposed model breaks the serialization processing structure of RNNs with better interaction results between characters and words through careful connections. The lexicon knowledge connects related characters to capture the local composition. Meanwhile, a global relay node is designed to capture long-range dependency and high-level features. LGN follows a neighborhood aggregation scheme wherein the node representation is computed by recursively aggregating its incoming edges and the global relay node. Because of multiple iterations of aggregation, the model can use global context information to repeatedly compare ambiguous words for better prediction. Experimental results show that the proposed method can achieve state-of-the-art performance on four NER datasets.

The main contributions of this paper can be summarized as follows: 1) we propose the use of a lexicon to construct a graph neural network and achieve Chinese NER as a graph node classification task; 2) the proposed model can capture global context information and local compositions to tackle Chinese word ambiguity problems through recursively aggregating mechanism; 3) several experimental results demonstrate the effectiveness of the proposed method in different aspects.

2 Related Work

2.1 Chinese NER with Lexicon.

Some previous Chinese NER researches have compared word-based and character-based methods (Li et al., 2014) and show that due to the limited performance of the current Chinese word segmentation, character-based name taggers can outperform their word-based counterparts (He and Wang, 2008; Liu et al., 2010). Lexicon features have been widely used to better leverage word information for Chinese NER (Huang et al., 2015; Luo et al., 2015; Gui et al., 2019). Especially, Zhang and Yang (2018) proposed a lattice LSTM to model characters and potential words simultaneously. However, their lattice LSTM used a concatenation of independently trained left-to-right and right-to-left LSTM to represent features, which was also limited (Devlin et al., 2018). In this work, we propose a novel character-based method that treats the named entities as a node classification task. The proposed method can utilize global information (both the left and the right context) (Dong et al., 2019) to tackle word ambiguities.

2.2 Graph Neural Networks on Texts

Graph neural networks have been successfully applied to several text classification tasks (Veličković et al., 2017; Yao et al., 2018; Zhang et al., 2018b). Peng et al. (2018) proposed a GCN-based deep learning model for text classification. Zhang et al. (2018c) proposed using the dependency parse trees to construct a graph for relation extraction. Recently, multi-head attention mechanisms (Vaswani et al., 2017) have been widely used by graph neural networks during the fusion process (Zhang et al., 2018a; Lee et al., 2018), which can aggregate graph information by assigning different weights to neighboring nodes or associated edges. Given a set of vectors $\mathbf{H} \in \mathbb{R}^{n\times d}$, a query vector $\hat{\mathbf{q}} \in \mathbb{R}^{1\times d}$, and a set of trainable parameters $\mathbf{W}$, this mechanism can be formulated as:

$$\begin{aligned}
&\text{Attention}(\mathbf{q},\mathbf{K},\mathbf{V}) = softmax(\frac{\mathbf{q}\mathbf{K}^\top}{\sqrt{d_k}})\mathbf{V} \\
&\text{head}_i = \text{Attention}(\hat{\mathbf{q}}\mathbf{W}_i^Q, \mathbf{H}\mathbf{W}_i^K, \mathbf{H}\mathbf{W}_i^V) \\
&\text{MultiAtt}(\hat{\mathbf{q}},\mathbf{H}) = [\text{head}_1; \ldots; \text{head}_k]\mathbf{W}^O. \quad (1)
\end{aligned}$$

However, very little work has explored how to use the relationship among characters to construct graphs in raw Chinese texts. The few previous studies on morphological processing in Chinese proposed a decomposed lexical structure (Zhang and Peng, 1992; Zhou and Marslen-Wilson, 1994) in which Chinese words are represented in terms of their constituent characters. Inspired by these

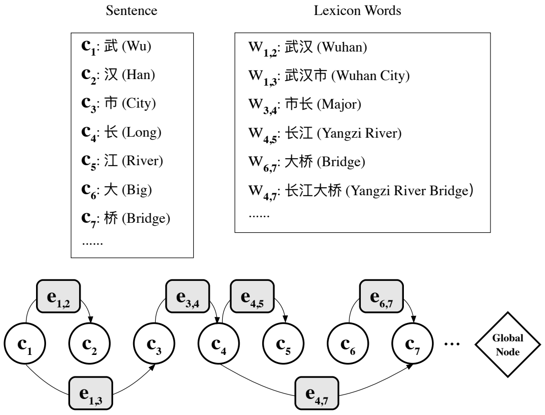


Figure 2: Illustration of graph construction.

theoretical basis, we propose the use of graph neural networks to construct the relationship between constituent characters and words.

3 Lexicon-Based Graph Neural Network

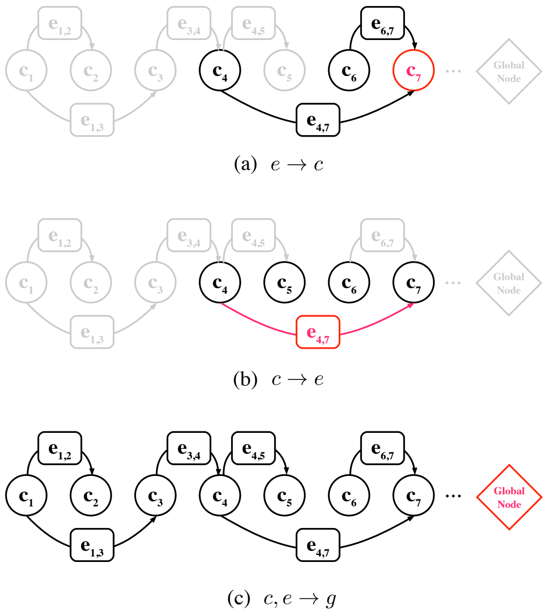
In this work, we propose the use of lexicon information to construct graph neural networks, and achieve Chinese NER as a node classification task. The proposed model obtains better interaction among characters, words, and sentences through, **aggregation** → **update** → **aggregation** → . . . , an efficient graph message passing architecture (Gilmer et al., 2017).

3.1 Graph Construction and Aggregation

We use the lexicon knowledge to connect characters to capture the local composition and potential word boundaries. In addition, we propose a global relay node to capture long-range dependency and high-level features. The implementation of the aggregation module for nodes and edges is similar to the multi-head attention mechanism in Transformer (Vaswani et al., 2017).

Graph Construction The whole sentence is converted into a directed graph, as shown in Figure 2, where each node represents a character and the connection between the first and last characters in a word can be treated as an edge. The state of the i-th node represents the features of the i-th token in a text sequence. The state of each edge represents the features of a corresponding potential word. The global relay node is used as a virtual hub to gather the information from all the nodes and edges, and then utilizes the global information to help the node remove ambiguity.

Formally, let $s = c_1, c_2, ..., c_n$ denote a sentence, where c_i denotes the i-th character. The potential words in the lexicon that match a character subsequence can be formulated as $w_{b,e} = c_b, c_{b+1}, ..., c_{e-1}, c_e$, where the index of the first and last letters are b and e, respectively. In this work, we propose the use of a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to model a sentence, where each character $c_i \in \mathcal{V}$ is a graph node and $\mathcal{E}$ is the set of edges. Once a character subsequence matches a potential word $w_{b,e}$, we construct one edge $e_{b,e} \in \mathcal{E}$, pointing from the beginning character c_b to the ending character c_e.



(a) $e \rightarrow c$

(b) $c \rightarrow e$

(c) $c, e \rightarrow g$

Figure 3: Aggregation in LGN. Red indicates the element that is being updated, and black indicates other elements involved in the aggregation. The aggregation results are then used in update modules.

To capture global information, we add a global relay node to connect each character node and word edge. For a graph with n character nodes and m edges, there are $n + m$ global connections linking each node and edge to the shared relay node. With the global connections, every two non-adjacent nodes are two-hop neighbors and receive non-local information with a two-step update.

In addition, we consider the transpose of the constructed graph[1]. It is another directed graph on the same set of nodes with all of the edges reversed compared to the orientation of the corresponding edges in $\mathcal{G}$. We denote the transpose graph as $\mathcal{G}^{\top}$. Similar to the bidirectional LSTM, we compose $\mathcal{G}$ and $\mathcal{G}^{\top}$ as a bidirectional graph and concatenate the node states of $\mathcal{G}$ and $\mathcal{G}^{\top}$ as final outputs.

Local Aggregation Given the node features $\mathbf{c}_i^t$

[1] https://en.wikipedia.org/wiki/Transpose_graph

and the incoming edge features $\mathbf{E}_{c_i}^t = \{\forall_k \mathbf{e}_{k,i}^t\}$, we use multi-head attention to aggregate $\mathbf{e}_{k,i}$ and the corresponding predecessor nodes $\mathbf{c}_k$ for each node $\mathbf{c}_i$, where intuition is that the incoming edges and predecessor nodes can effectively indicate potential word boundary information, as shown in Figure 3 (a). Formally, the node aggregation function can be formulated as follows:

$$e \rightarrow c: \hat{\mathbf{c}}_i^t = \mathrm{MultiAtt}(\mathbf{c}_i^t, \{\forall_k [\mathbf{c}_k^t; \mathbf{e}_{k,i}^t]\}), \quad (2)$$

where t refers to the aggregation at the t-th step and $[\cdot;\cdot]$ represents concatenation operation.

For edge aggregation, all the forces or potential energies acting on the edges should be considered (Battaglia et al., 2018). To exploit the word orthographic information, lexicons used to construct edges should consider all the character composition, as shown in Figure 3 (b). Hence, different from the classic graph neural networks that use the features of terminal vertices to aggregate edges, we use the whole matching character subsequence $\mathbf{C}_{b,e}^t = \{\mathbf{c}_b^t, \ldots, \mathbf{c}_e^t\}$ for the edge aggregation function, as follows:

$$c \rightarrow e: \quad \hat{\mathbf{e}}_{b,e}^t = \mathrm{MultiAtt}(\mathbf{e}_{b,e}^t, \mathbf{C}_{b,e}^t). \quad (3)$$

Given the character sequence embeddings $\mathbf{C} \in \mathbb{R}^{n \times d}$ and potential word embeddings $\mathbf{E} \in \mathbb{R}^{m \times d}$, we first fed $\mathbf{C}$ into an LSTM network to generate contextualized representations as the initial node states $\mathbf{C}^0$ (Zhang et al., 2018c), and we used the word embeddings as the initial edge states $\mathbf{E}^0$.

Global Aggregation The underlying structure of language is not strictly sequential (Shen et al., 2019). To capture long-range dependency and high-level features, as shown in Figure 3 (c), we utilized a global relay node to aggregate each character node and edge, as follows:

$$\begin{aligned}
\mathbf{g}_c^t &= \mathrm{MultiAtt}(\mathbf{g}^t, \mathbf{C}_{1,n}^t) \\
\mathbf{g}_e^t &= \mathrm{MultiAtt}(\mathbf{g}^t, \{\forall \mathbf{e}^t \in \mathcal{E}\}) \\
c, e \rightarrow g: \quad \hat{\mathbf{g}}^t &= [\mathbf{g}_c^t; \mathbf{g}_e^t]. \quad (4)
\end{aligned}$$

After multiple exchanges of information (§ 3.2), $\hat{\mathbf{g}}_t$ aggregates node vectors and edge vectors to summarize the global information, and $\hat{\mathbf{e}}_{b,e}^t$ captures the compositional character information to form the local composition. As a result, the proposed model, with a thorough knowledge of both local and non-local composition, would contribute character nodes to distinguish ambiguous words (Ma et al., 2014).

3.2 Recurrent-based Update Module

Node Update The effective use of sentence context to tackle the ambiguity among the potential words is still a key issue (Ma et al., 2014). For a general graph, it is common practice to apply recurrent-based modules to update hidden representations of nodes (Scarselli et al., 2009; Li et al., 2015). Hence, we fused the global feature $\hat{\mathbf{g}}$ into a character nodes update module, as follows:

$$\begin{aligned}
\boldsymbol{\xi}_i^t &= [\mathbf{c}_{i-1}^{t-1}; \mathbf{c}_i^{t-1}], \quad \boldsymbol{\chi}_i^t = [\hat{\mathbf{c}}_i^{t-1}; \hat{\mathbf{g}}^{t-1}] \\
\hat{\boldsymbol{a}}_i^t &= \sigma(\mathbf{W}_i^a \boldsymbol{\xi}_i^t + \mathbf{V}_i^a \boldsymbol{\chi}_i^t + \mathbf{b}_i^a), \quad \boldsymbol{a} = \{\boldsymbol{i}, \boldsymbol{f}, \boldsymbol{l}\} \\
\boldsymbol{u}_i^t &= tanh(\mathbf{W}_{cu} \boldsymbol{\xi}_i^t + \mathbf{V}_{cu} \boldsymbol{\chi}_i^t + \mathbf{b}_{cu}) \\
\boldsymbol{i}_i^t, \boldsymbol{f}_i^t, \boldsymbol{l}_i^t &= softmax(\hat{\boldsymbol{i}}_i^t, \hat{\boldsymbol{f}}_i^t, \hat{\boldsymbol{l}}_i^t) \\
\mathbf{c}_i^t &= \boldsymbol{l}_i^t \odot \mathbf{c}_{i-1}^{t-1} + \boldsymbol{f}_i^t \odot \mathbf{c}_i^{t-1} + \boldsymbol{i}_i^t \odot \boldsymbol{u}_i^t, \quad (5)
\end{aligned}$$

where $\mathbf{W}$, $\mathbf{V}$, and $\mathbf{b}$ are trainable parameters. $\boldsymbol{\xi}_i^t$ is the concatenation of adjacent vectors of a context window. The window size in our model is 2 and actually plays a role as a character bigram, which has been shown to be useful for representing characters in sequence labeling tasks (Chen et al., 2015a; Zhang and Yang, 2018). $\boldsymbol{\chi}_i^t$ is the concatenation of the global information vector $\hat{\mathbf{g}}^t$ and the $e \rightarrow c$ aggregation result $\hat{\mathbf{c}}_i^t$. The gates $\boldsymbol{i}_i^t$, $\boldsymbol{f}_i^t$ and $\boldsymbol{l}_i^t$ control information flow from global features to $\mathbf{c}_i^t$, which can make further readjustment of the weights of the lexicon attention ($e \rightarrow c$) to tackle the ambiguities at the subsequent aggregation step.

Edge Update To better leverage the interaction among characters, words, and whole sentences, we not only designed a recurrent module for nodes but also for edges and the global relay node (Battaglia et al., 2018). We update the edges as follows:

$$\begin{aligned}
\boldsymbol{\chi}_{b,e}^t &= [\hat{\mathbf{e}}_{b,e}^{t-1}; \hat{\mathbf{g}}^{t-1}], \quad \boldsymbol{a} = \{\boldsymbol{i}, \boldsymbol{f}\} \\
\hat{\boldsymbol{a}}_{b,e}^t &= \sigma(\mathbf{W}_i^a \mathbf{e}_{b,e}^{t-1} + \mathbf{V}_i^a \boldsymbol{\chi}_{b,e}^t + \mathbf{b}_i^a) \\
\boldsymbol{u}_{b,e}^t &= tanh(\mathbf{W}_{eu} \mathbf{e}_{b,e}^{t-1} + \mathbf{V}_{eu} \boldsymbol{\chi}_{b,e}^t + \mathbf{b}_{eu}) \\
\boldsymbol{i}_{b,e}^t, \boldsymbol{f}_{b,e}^t &= softmax(\hat{\boldsymbol{i}}_{b,e}^t, \hat{\boldsymbol{f}}_{b,e}^t) \\
\mathbf{e}_{b,e}^t &= \boldsymbol{f}_{b,e}^t \odot \mathbf{e}_{b,e}^{t-1} + \boldsymbol{i}_{b,e}^t \odot \boldsymbol{u}_{b,e}^t, \quad (6)
\end{aligned}$$

where $\boldsymbol{\chi}_{b,e}^t$ is the concatenation of $\hat{\mathbf{g}}^t$ and the $c \rightarrow e$ aggregation result $\hat{\mathbf{e}}_{b,e}^t$. Similar to the node update function, $\boldsymbol{i}_{b,e}^t$ and $\boldsymbol{f}_{b,e}^t$ are gates that control information flow from $\mathbf{e}_{b,e}^{t-1}$ and $\hat{\mathbf{g}}^t$ to $\mathbf{e}_{b,e}^t$.

Global Relay Node Update In terms of the global state $\mathbf{g}$, recent works (Zhang et al., 2018b; Guo et al., 2019) have shown the effectiveness of sharing useful messages across contexts. Thus, we also designed an update function for $\mathbf{g}$, with the initialization $\mathbf{g}^0 = \text{average}(\mathbf{C}, \mathbf{E})$. More formally:

$$\begin{aligned}
\hat{\boldsymbol{a}}^t &= \sigma(\mathbf{W}_i^a \mathbf{g}^{t-1} + \mathbf{V}_i^a \hat{\mathbf{g}}^{t-1} + \mathbf{b}_i^a),\ \boldsymbol{a} = \{\boldsymbol{i}, \boldsymbol{f}\} \\
\boldsymbol{u}^t &= tanh(\mathbf{W}_{gu} \mathbf{g}^{t-1} + \mathbf{V}_{gu} \hat{\mathbf{g}}^{t-1} + \mathbf{b}_{gu}) \\
\boldsymbol{i}^t, \boldsymbol{f}^t &= softmax(\hat{\boldsymbol{i}}^t, \hat{\boldsymbol{f}}^t) \\
\mathbf{g}^t &= \boldsymbol{f}^t \odot \mathbf{g}^{t-1} + \boldsymbol{i}^t \odot \boldsymbol{u}^t. \qquad (7)
\end{aligned}$$

3.3 Decoding and Training

A standard conditional random field (CRF) is used in the graph message passing process. Given the sequence of final node states $\mathbf{c}_1^T, \mathbf{c}_2^T, \ldots, \mathbf{c}_n^T$, the probability of a label sequence $\hat{y} = \hat{l}_1, \hat{l}_2, \ldots, \hat{l}_n$ can be defined as follows:

$$p(\hat{y}|s) = \frac{exp(\sum_{i=1}^{n} \phi(\hat{l}_{i-1}, \hat{l}_i, \mathbf{c}_i^T))}{\sum_{y' \in \mathrm{Y}(s)} exp(\sum_{i=1}^{n} \phi(l'_{i-1}, l'_i, \mathbf{c}_i^T))}, \qquad (8)$$

where, $\mathrm{Y}(s)$ is the set of all arbitrary label sequences. $\phi(l_{i-1}, l_i, \mathbf{c}_i^T) = \mathbf{W}_{(l_{i-1}, l_i)} \mathbf{c}_i^T + \mathbf{b}_{(l_{i-1}, l_i)}$, where $\mathbf{W}_{(l_{i-1}, l_i)}$ and $\mathbf{b}_{(l_{i-1}, l_i)}$ are the weight and bias parameters specific to the labels l_{i-1} and l_i.

For training, we minimize the sentence-level negative log-likelihood loss as follows:

$$L = -\sum_{i=1}^{N} log(p(y_i|s_i)). \qquad (9)$$

For testing and decoding, we maximized the likelihood to find the optimal sequence y^*:

$$y^* = \operatorname*{argmax}_{y \in \mathrm{Y}(s)} p(y|s). \qquad (10)$$

We used the Viterbi algorithm to calculate the above equations, which can reduce the computational complexity efficiently.

4 Experimental Setup

In this section, we describe the datasets across different domains and the baseline methods applied for comparison. We also detail the hyperparameter configuration of the proposed model. Our codes and datasets can be found at `https://github.com/RowitZou/LGN`.

Dataset	Type	Train	Dev.	Test
OntoNotes	Sent.	15.7k	4.3k	4.3k
	Char.	491.9k	200.5k	208.1k
MSRA	Sent.	46.4k	-	4.4k
	Char.	2169.9k	-	172.6k
Weibo	Sent.	1.4k	0.27k	0.27k
	Char.	73.8k	14.5k	14.8k
Resume	Sent.	3.8k	0.46k	0.48k
	Char.	124.1k	13.9k	15.1k

Table 1: Statistics of datasets.

4.1 Data

We conducted experiments on four Chinese NER datasets. (1) **OntoNotes 4.0** (Weischedel et al., 2011): OntoNotes is a manually annotated multilingual corpus in the news domain that contains various text annotations, including Chinese named entity labels. Gold-standard segmentation is available. We only use Chinese documents (about 16k sentences) and process the data in the same way as Che et al. (2013). (2) **MSRA** (Levow, 2006): MSRA is also a dataset in the news domain and contains three types of named entities: LOC, PER, and ORG. Chinese word segmentation is available in the training set but not in the test set. (3) **Weibo NER** (Peng and Dredze, 2015): It consists of annotated NER messages drawn from the social media Sina Weibo[2]. The corpus contains PER, ORG, GPE, and LOC for both named entity and nominal mention. (4) **Resume NER** (Zhang and Yang, 2018): It is composed of resumes collected from Sina Finance[3] and is annotated with 8 types of named entities. Both Weibo and Resume datasets do not contain the gold-standard Chinese segmentation. Statistics of the above datasets are detailed in Table 1.

4.2 Lexicon

We used the lexicon over automatically segmented Chinese Giga-Word [4], obtaining 704.4k words in the final lexicon. The embeddings of lexicon words were pre-trained using word2vec (Mikolov et al., 2013) and fine-tuned during training. According to the lexicon statistics, the number of single-character, two-character and three-character words are 5.7k, 291.5k, 278.1k, respectively. It covers 31.2% of the named entities in the four data sets, which means most of the lexicon words are not named entities. For a

[2] https://www.weibo.com

[3] https://finance.sina.com.cn/stock/

[4] https://catalog.ldc.upenn.edu/LDC2011T13

fair comparison, we used such a general lexicon instead of a professional named entity lexicon in our experiments and we still obtained competitive results. Empirically, a high-quality lexicon could lead to further performance improvements.

Character embeddings are pre-trained on Chinese Giga-Word using word2vec and fine-tuned at model training. Both the pre-trained character and lexicon word embeddings are released by Zhang and Yang (2018)[5].

4.3 Comparison Methods

We applied the character-level and word-level methods as baselines for comparison, which incorporate the bichar, softword, and lexicon features. We also compared several state-of-the-art methods on the four datasets to verify the effectiveness of our method. We used the BMES tagging scheme for both character-level and word-level NER tagging.

Character-level methods: These methods are based on character sequences. We applied the bi-directional LSTM (Hochreiter and Schmidhuber, 1997) and CNN (Kim, 2014) as classic baseline methods.

Character-level methods + bichar + softword: Character bigrams are useful for capturing adjacent features and representing characters. We concatenated bigram embeddings with character embeddings to better leverage the bigram information. In addition, we added the segmentation information by incorporating segmentation label embeddings into the character representation. The BMES scheme is used for representing the word segmentation (Xue and Shen, 2003).

Word-level methods: For the datasets with gold segmentation, we directly employed word-level NER methods to evaluate the performance, which are denoted as **Gold seg**. Otherwise, we first used open source segmentation toolkit[6] to automatically segment the datasets. Then word-level NER methods are applied, which are denoted as **Auto seg**. The bi-directional LSTM and CNN are also applied as baselines.

Word-level methods + char + bichar: For characters in the subsequence $w_{b,e}$, we first used a bi-directional LSTM to learn their hidden states and bigram states. We then augmented the word-level methods with the character-level features.

Input	Models	P	R	F1
Gold seg.	Yang et al. (2016)	65.59	71.84	68.57
	Yang et al. (2016)*†	72.98	**80.15**	**76.40**
	Che et al. (2013)*	77.71	72.51	75.02
	Wang et al. (2013)*	76.43	72.32	74.32
	Word-level LSTM	76.66	63.60	69.52
	+char+bichar	**78.62**	73.13	75.77
	Word-level CNN	66.84	62.99	64.86
	+char+bichar	68.22	72.37	70.24
Auto seg.	Word-level LSTM	72.84	59.72	65.63
	+char+bichar	73.36	70.12	71.70
	Word-level CNN	54.62	55.20	54.91
	+char+bichar	64.69	65.09	64.89
No seg.	Char-level LSTM	68.79	60.35	64.30
	+bichar+softword	74.36	69.43	71.89
	Char-level CNN	56.78	60.99	58.81
	+bichar+softword	59.60	65.14	62.25
	Lattice LSTM	**76.35**	71.56	73.88
	LGN	76.13	**73.68**	**74.89**

Table 2: Main results on OntoNotes.

Lattice LSTM: Lattice LSTM (Zhang and Yang, 2018) incorporates word information into character-level recurrent units, which can avoid segmentation errors. This method achieved state-of-the-art performance on the four datasets.

4.4 Hyper-parameter Settings

We used the Adam (Kingma and Ba, 2014) as the optimizer, with a learning rate of 2e-5 for large datasets like Ontonotes and MSRA, while a rate of 2e-4 for small datasets Weibo and Resume. A densely connected structure (Huang et al., 2017) was applied, which composites all hidden states from previous update steps as final inputs for aggregation modules at step t. To further reduce overfitting, we employed the Dropout (Srivastava et al., 2014) with a rate of 0.5 for embeddings and a rate of 0.2 for aggregation module outputs. The embedding size and state size were both set to 50. The head number of multi-head attention was 10. The head dimension was set to 10 for small datasets like Weibo and Resume, while the head dimension was 20 for Ontonotes and MSRA. Step number T was selected among $\{1, 2, 3, 4, 5, 6\}$, which is detailed analyzed in § 5.2. The standard Precision (P), Recall (R), and F1-score (F1) were used as evaluation metrics.

5 Results and Discussion

In this section, we demonstrate the main results of LGN for the Chinese NER task across different domains. The model achieving best results on the development set was chosen for the final evaluation on the test set. We also probe the

[5] https://github.com/jiesutd/LatticeLSTM

[6] https://github.com/lancopku/PKUSeg-python

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
Word-level LSTM	90.57	83.06	86.65
+char+bichar	91.05	89.53	90.28
Char-level LSTM	90.74	86.96	88.81
+bichar+softword	92.97	90.80	91.87
Lattice LSTM	94.18	92.20	93.18
LGN	**94.19**	**92.73**	**93.46**

Table 3: Main results on MSRA.

Models	NE	NM	All
Peng and Dredze (2015)	51.96	61.05	56.05
Peng and Dredze (2015)*	55.28	62.97	58.99
He and Sun (2016)	50.60	59.32	54.82
He and Sun (2017)*	54.50	62.17	58.23
Word-level LSTM	36.02	59.38	47.33
+char+bichar	43.40	60.30	52.33
Char-level LSTM	46.11	55.29	52.77
+bichar+softword	50.55	60.11	56.75
Lattice LSTM	53.04	62.25	58.79
LGN	**55.34**	**64.98**	**60.21**

Table 4: Main results on Weibo.

Models	P	R	F1
Word-level LSTM	93.72	93.44	93.58
+char+bichar	94.07	94.42	94.24
Char-level LSTM	93.66	93.31	93.48
+bichar+softword	94.53	94.29	94.41
Lattice LSTM	94.81	94.11	94.46
LGN	**95.28**	**95.46**	**95.37**

Table 5: Main results on Resume.

effectiveness and interpretability of LGN by explanatory experiments.

5.1 Main Results

OntoNotes Table 2[7] shows the results of word-level and character-level methods on OntoNotes with various settings. In the gold or automatic segmentation settings, the char and bichar features boost the performance of word-level methods. In particular, with the gold-standard segmentation, these methods are able to achieve competitive state-of-the-art results on the dataset (Che et al., 2013; Wang et al., 2013). However, the gold-standard segmentation is not always available. On the other hand, the automatic segmentation may induce word segmentation errors and result in a loss of performance for the downstream NER task. A feasible solution is applying character-level methods to avoid the need for word segmentation. Our proposed LGN is a character-level model based on graphic structure. It outperforms lattice LSTM by 1.01% in F1 score and leads to a 3.00% increment of F1 score over the LSTM with bichar and softword features. The LGN also significantly outperforms the word-level models with automatic segmentation.

MSRA/Weibo/Resume Results on the MSRA, Weibo, and Resume datasets are shown in Table 3, 4, and 5, respectively. Gold-standard segmentation is not available for the Weibo and Resume datasets and the test set of MSRA. The best classic methods leverage rich handcrafted features (Chen et al., 2006; Zhang et al., 2006; Zhou et al., 2013), embedding features (Lu et al., 2016; Peng and Dredze, 2015), radical features (Dong et al., 2016), cross-domain, and semi-supervised data (He and Sun, 2017) for Chinese NER. Compared with the existing methods and the word-level and character-level methods, our LGN model gives the best results by a large margin. Moreover, different from the lattice LSTM, which also leverages lexicon features, our LGN model integrates lexicon information into the graph neural network in a more effective fashion. As a result, it outperforms the lattice LSTM on all three datasets.

5.2 Steps of Message Passing

To investigate the influence of step number T during the update process, we analyzed the performance of LGN under different step numbers. Figure 4 illustrates the variation of F1 score on the development sets[8] as the step number increases. We used **D-F1** to represent the F1 scores at different steps minus the best results.

The results indicate that the number of update steps is crucial to the performance of LGN, which peaks when $T \geq 3$ on all four datasets. The F1 score decreases 1.20% on average against the best results when the step number is less than 3. In particular, the F1 score of the OntoNotes and Weibo datasets even suffered a serious reduction around 1.5% and 1.8%, respectively. After several rounds of updates, the model gives steady and competitive results and reveals that LGN benefits from the update process. Empirically, at each

[7] In Table 2, 3, 4 and 5, the models with * use external labeled data for semi-supervised learning. Those with † also use discrete features.

[8] Since the MSRA dataset does not have the development set, we used the test set instead.

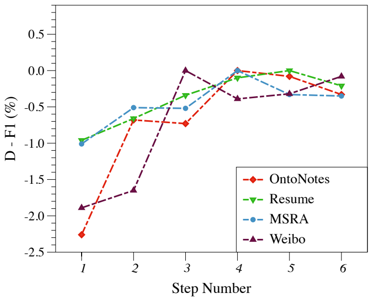


Figure 4: F1 variation under different update steps on the development sets. D-F1 represents the F1 scores at different steps minus the best results.

Models	Onto-Notes	MSRA	Weibo	Resume
LGN	71.96	93.46	62.42	94.43
-global	71.26	92.99	62.30	94.31
-edge /lexicon	65.88	89.63	59.19	94.05
-edge-global	65.34	89.47	58.62	94.09
-bidirection	67.52	90.98	59.67	94.23
-crf	66.37	91.13	57.73	92.70
Lattice LSTM	71.62	93.18	61.64	93.64
-bidirection	66.63	90.72	58.75	93.21
-crf	61.74	85.38	55.71	92.31

Table 6: An ablation study of LGN. F1 scores were evaluated on the development sets.

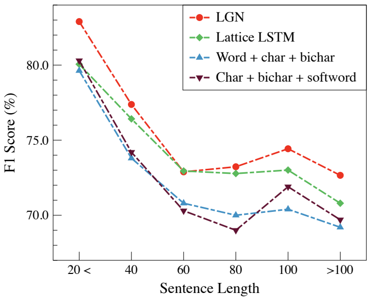


Figure 5: F1 score against sentence length on the OntoNotes dateset.

update step, graph nodes aggregate information from their neighbors and incrementally gain more information from further reaches of the graph as the process iterates (Hamilton et al., 2017). In the LGN model, more valuable information can be captured through the recursive aggregation.

5.3 Ablation Experiments

To study the contribution of each component in LGN, we conducted ablation experiments on the four datasets and display the results in Table 6. The results show that the model's performance is degraded if the global relay node is removed, indicating that global connections are useful in the graph structure. We also find that lexicons play an important role in character-level NER. In particular, the performance of the OntoNotes, MSRA and Weibo datasets are seriously hurt by over 3.0% without lexicons. Moreover, missing both edges and the global node will cause a further performance loss.

To better illustrate the advantage of our model, we remove the CRF decoding layer and simplify the structure to a non-bidirectional version on both LGN and the lattice LSTM model. The results show that, with a single direction structure, the LGN achieves a higher F1 score by 0.77% on average than the lattice LSTM. In addition, the two models have an obvious performance gap when they get rid of the CRF layer. The F1 score of LGN decreases by 3.59% on average on the four datasets without CRF. In contrast, the lattice LSTM decreases by 6.24%. It manifests the LGN has stronger ability to model sentences.

5.4 Performance Against Sentence Length

Figure 5 shows the performance of LGN and several baseline models on the OntoNotes dataset. We split the dataset into six parts according to the sentence length. The lattice is a strong baseline that outperforms the word+char+bichar and char+bichar+softword models over different sentence lengths. However, the lattice accuracy decreases significantly as the sentence length increases. In contrast, the LGN not only gives higher results over short sentences, but also shows its effectiveness and robustness when the sentence length is more than 80 characters. It gives a higher F1 score in most cases compared to the baselines, which indicates that global sentence semantics and long-range dependency can be better captured under the graph structure.

5.5 Case Study

Table 7 illustrates an example that probes the ability of LGN to tackle the word ambiguity problems. The lattice LSTM ignores the sentence context and wrongly identifies "印度(India)". Removing the global relay node, LGN also makes the same mistake, which indicates that global connections are indispensable and can capture high-level information to help LGN better understand the sentence context. In contrast, with the global relay node, the LGN can correctly identify the entity boundary, even though the

Sentence	印度河流经巴基斯坦 The Indus River flows through Pakistan.
Gold seg	印度河 流经 巴基斯坦 The Indus River, flow through, Pakistan
Lexicon words	印度 河流 印度河 流经 巴基斯坦 India, river, The Indus River, flow through, Pakistan
Lattice LSTM	B E (GPE) O O O B M M E (GPE) 印度 (GPE) 河流经 巴基斯坦 (GPE) India (GPE) ... Pakistan (GPE).
LGN -global	B E (GPE) O O O B M M E (GPE) 印度 (GPE) 河流经 巴基斯坦 (GPE) India (GPE) ... Pakistan (GPE).
LGN (one step)	B M E (GPE) O O B M M E (GPE) 印度河 (GPE) 流经 巴基斯坦 (GPE) The Indus River (GPE) flows through Pakistan (GPE).
LGN	B M E (LOC) O O B M M E (GPE) 印度河 (LOC) 流经 巴基斯坦 (GPE) The Indus River (LOC) flows through Pakistan (GPE).

Table 7: An example with overlapping ambiguity. Contents with red and blue colors represent incorrect and correct entities, respectively.

graph composition states are updated for only one step. However, it gives an incorrect class of the entity "印度河(The Indus River)", which is a location entity but not a GPE (Geo-Political Entity). Because of the multi-step graph message passing process, the LGN is able to fuse the context information and finally detects the correct location entity in success.

6 Conclusion

In this work, we investigated a GNN-based approach to alleviate the word ambiguity in Chinese NER. Lexicons are used to construct the graph and provide word-level features. The LGN enables interactions among different sentence compositions and can capture non-sequential dependencies between characters based on the global sentence semantics. As a result, it shows improved performance significantly on multiple datasets in different domains. The explanatory experiments also illustrate the effectiveness and interpretability of our proposed model.

Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by China National Key R&D Program (No.2018YFC0831105, 2018YFB1005104, 2017YFB1002104), National Natural Science Foundation of China (No. 61976056, 61532011, 61751201), Shanghai Municipal Science and Technology Major Project (No.2018SHZDZX01), STCSM (No.16JC1420401, 17JC1420200).

References

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. 2018. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

Razvan C Bunescu and Raymond J Mooney. 2005. A shortest path dependency kernel for relation extraction. In *HLT—EMNLP*.

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, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuanjing Huang. 2015a. Long short-term memory neural networks for chinese word segmentation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1197–1206.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015b. Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL—IJCNLP*, volume 1, pages 167–176.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dennis Diefenbach, Vanessa Lopez, Kamal Singh, and Pierre Maret. 2018. Core techniques of question answering systems over knowledge bases: a survey. *KAIS*, 55(3):529–569.

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. Springer.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*.

Huanzhong Duan and Yan Zheng. 2011. A study on features of the crfs-based chinese named entity recognition. *IJAI*, 3(2):287–294.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org.

Tao Gui, Ruotian Ma, Qi Zhang, Lujun Zhao, Yu-Gang Jiang, and Xuanjing Huang. 2019. Cnn-based chinese ner with lexicon rethinking. In *IJCAI*.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. 2019. Star-transformer. *arXiv preprint arXiv:1902.09113*.

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. 2016. F-score driven max margin neural network for named entity recognition in chinese social media. *arXiv preprint arXiv:1611.04234*.

Hangfeng He and Xu Sun. 2017. A unified model for cross-domain and semi-supervised named entity recognition in chinese social media. In *AAAI*.

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.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

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

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

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

John Boaz Lee, Ryan Rossi, and Xiangnan Kong. 2018. Graph classification using structural attention. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1666–1674. ACM.

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. 2014. Comparison of the impact of word segmentation on name tagging for chinese and japanese. In *LREC*, pages 2532–2536.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2015. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*.

Zhangxun Liu, Conghui Zhu, and Tiejun Zhao. 2010. Chinese named entity recognition with a sequence labeling approach: based on characters, or based on words? In *International Conference on Intelligent Computing*, pages 634–640. Springer.

Yanan Lu, Yue Zhang, and Dong-Hong Ji. 2016. Multi-prototype chinese character embedding. In *LREC*.

Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint entity recognition and disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 879–888.

Guojie Ma, Xingshan Li, and Keith Rayner. 2014. Word segmentation of overlapping ambiguous strings during chinese reading. *Journal of Experimental Psychology: Human Perception and Performance*, 40(3):1046.

Luisa Mich, Roberto Garigliano, et al. 2000. Ambiguity measures in requirement engineering. In *International Conference on Software Theory and Practice. ICS*.

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.

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. *CoNLL-2014*, page 78.

Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. 2018. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 1063–1072. International World Wide Web Conferences Steering Committee.

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for chinese social media with jointly trained embeddings. In *EMNLP*.

Charles A Perfetti and Li Hai Tan. 1999. The constituency model of chinese word identification.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. *Proceedings of ICLR*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

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*.

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*.

Nianwen Xue and Libin Shen. 2003. Chinese word segmentation as lmr tagging. In *Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17*, pages 176–179. Association for Computational Linguistics.

Jie Yang, Zhiyang Teng, Meishan Zhang, and Yue Zhang. 2016. Combining discrete and neural features for sequence labeling. In *CICLing*. Springer.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2018. Graph convolutional networks for text classification. *arXiv preprint arXiv:1809.05679*.

Miao-Hsuan Yen, Ralph Radach, Ovid J-L Tzeng, and Jie-Li Tsai. 2012. Usage of statistical cues for word boundary in reading chinese sentences. *Reading and writing*, 25(5):1007–1029.

Biyin Zhang and Danling Peng. 1992. Decomposed storage in the chinese lexicon. In *Advances in psychology*, volume 90, pages 131–149. Elsevier.

Jiani Zhang, Xingjian Shi, Junyuan Xie, Hao Ma, Irwin King, and Dit-Yan Yeung. 2018a. Gaan: Gated attention networks for learning on large and spatiotemporal graphs. *arXiv preprint arXiv:1803.07294*.

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, Qi Liu, and Linfeng Song. 2018b. Sentence-state lstm for text representation. *arXiv preprint arXiv:1805.02474*.

Yue Zhang and Jie Yang. 2018. Chinese ner using lattice lstm. *arXiv preprint arXiv:1805.02023*.

Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018c. Graph convolution over pruned dependency trees improves relation extraction. *arXiv preprint arXiv:1809.10185*.

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.

Xiaolin Zhou and William Marslen-Wilson. 1994. Words, morphemes and syllables in the chinese mental lexicon. *Language and Cognitive Processes*, 9(3):393–422.