Lexicon-Based Graph Convolutional Network for Chinese Word Segmentation

Kaiyu Huang **Hao Yu** **Junpeng Liu**
Wei Liu **Jingxiang Cao** **Degen Huang***
School of Computer Science, Dalian University of Technology
{kaiyuhuang, yuhaodlut,liujunpeng_nlp, liuweidlut}
@mail.dlut.edu.cn
{caojx, huangdg}@dlut.edu.cn

Abstract

Precise information of word boundary can alleviate the problem of lexical ambiguity to improve the performance of natural language processing (NLP) tasks. Thus, Chinese word segmentation (CWS) is a fundamental task in NLP. Due to the development of pre-trained language models (PLM), pre-trained knowledge can help neural methods solve the main problems of the CWS in significant measure. Existing methods have already achieved high performance on several benchmarks (e.g., Bakeoff-2005). However, recent outstanding studies are limited by the small-scale annotated corpus. To further improve the performance of CWS methods based on fine-tuning the PLMs, we propose a novel neural framework, LBGCN, which incorporates a **l**exicon-**b**ased **g**raph **c**onvolutional **n**etwork into the Transformer encoder. Experimental results on five benchmarks and four cross-domain datasets show the LBGCN successfully captures the information of candidate words and helps to improve performance on the benchmarks (Bakeoff-2005 and CTB6) and the cross-domain datasets (SIGHAN-2010). Further experiments and analyses demonstrate that our proposed framework effectively models the lexicon to enhance the ability of basic neural frameworks and strengthens the robustness in the cross-domain scenario.[1]

1 Introduction

Neural methods often leverage word-level information to improve the performance of many downstream natural language processing (NLP) tasks such as text classification and machine translation (Yang et al., 2018), etc. Therefore, in determining the word boundary, word segmentation is regarded as a prerequisite for most downstream NLP tasks. Unlike most written languages, the Chinese written language has no explicit delimiters to separate words in the written text. Thus, Chinese word segmentation (CWS) is an essential and pre-processing step for many Chinese NLP tasks.

With the development of deep learning techniques, recent neural CWS approaches that do not heavily rely on the hand-craft feature engineering have already achieved high performance on several benchmark datasets (Cai and Zhao, 2016; Cai et al., 2017; Ma et al., 2018). In particular, recent outstanding studies have also exploited the learning paradigm in applying pre-trained language models (PLM) for many NLP tasks. Various methods that fine-tune PLMs have achieved progress on in-domain and cross-domain CWS without much manual effort (Meng et al., 2019; Huang et al., 2020; Tian et al., 2020; Ke et al., 2021).

Prior research has shown that the problems of CWS are segmentation ambiguity and out-of-vocabulary (OOV) words (Zhao et al., 2019). With the help of the pre-trained knowledge (Devlin et al., 2018; Liu et al., 2019), the fine-tuning CWS methods can effectively alleviate these two issues and outperform other neural network architectures. The methods fine-tuning PLMs become the mainstream approach for CWS. However, the performance of fine-tuning CWS methods is limited by the scale and quality of annotated CWS corpus. The dependencies between neighboring Chinese characters are diverse and it is hard to build a large-scale annotated corpus because of the characteristics of linguistics in Chinese. The difficulty of manual annotation restricts the scale and quality of CWS datasets. Besides, directly fine-tuning methods do not utilize contextual n-grams or other contextual information, which is important for previous model architectures (e.g., BiLSTM and Transformer) (Huang et al., 2015; Ma et al., 2018; Qiu et al., 2020). The methods that fine-tune PLMs may generate segmentation errors because of ambigu-

*Corresponding author
[1]Source codes of this paper are available on https://github.com/koukaiu/lbgcn

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 2908–2917
November 7–11, 2021.

ous contextual information. Thus, it is a challenge to design a framework that can effectively transfer pre-trained knowledge into the CWS.

In this paper, we propose the LBGCN, a neural framework with a lexicon-based graph convolutional network (GCN), to improve the performance of the CWS by leveraging lexicon knowledge. In detail, we utilize the GCN to extract contextual features of candidate words and the information of word boundary from the pre-defined lexicon. The neural framework incorporates the GCN into the Transformer encoder (Vaswani et al., 2017) which is a part of PLM (e.g., BERT (Devlin et al., 2018)). The additional lexicon-based GCN can supply a gap of fine-tuning paradigm and better transfer pre-trained knowledge into the in-domain and cross-domain CWS tasks. Besides, through multi-feature interaction, the disambiguation and OOV word recognition are effectively carried out.

To sum up, the contributions of this work are as follows:

- Our proposed framework mainly consists of a lexicon-based GCN and the Transformer encoder. The lexicon-based GCN captures rich contextual information to alleviate the problem of lack-training by the small-scale annotated corpus. This framework achieves a noticeable improvement for CWS.
- Experimental results obtained from widely used benchmark datasets demonstrate that LBGCN can improve the performance compared with powerful baseline methods and outperform previous state-of-the-art studies.
- The novel method extracts the information from the lexicon via the GCN and is not over-reliant on the quality of the lexicon. Experimental results in the cross-domain scenario prove that the method can enhance the robustness of the basic neural CWS approaches.

2 Related Work

Chinese Word Segmentation Since Xue (2003) formalizes the CWS as a sequence labeling problem, most studies follow the character-based paradigm to predict segmentation labels for each character in the sentence. In particular, the adopted methods fall into two categories, including 1) statistical machine learning methods (Peng et al., 2004; Tseng et al., 2005; Zhao and Kit, 2008; Zhao et al., 2010) and 2) neural network methods (Zheng et al., 2013; Pei et al., 2014; Chen et al., 2015a,b; Cai and Zhao, 2016; Yang et al., 2017). As the studies of deep learning techniques develop in-depth, the neural CWS methods achieve better performance compared with statistical learning methods (Cai et al., 2017; Zhou et al., 2017; Ma et al., 2018; Yang et al., 2019a; Wang et al., 2019). And neural network architectures gradually replace statistical machine learning methods as the mainstream approaches for CWS.

Cross-Domain CWS However, there is an obvious gap in the cross-domain CWS scenario. Neural CWS methods still suffer from the OOV problems. To alleviate this problem, many kinds of research utilize external resources (e.g., pre-trained embeddings, unlabeled data, and lexicons) to improve the performance of the cross-domain CWS (Zhao et al., 2018; Zhang et al., 2018; Ye et al., 2019; Ding et al., 2020). For example, Huang et al. (2020) try to transfer pre-trained knowledge into the cross-domain CWS in full by leveraging more annotated datasets with different segmentation criteria (Chen et al., 2017). Tian et al. (2020) utilize lexicons and wordhood measures to enhance the robustness in the cross-domain CWS scenario.

Graph Neural Network In recent years, the graph neural network has been fully explored and achieved significant progress in several kinds of NLP tasks (Zhou et al., 2020). When dealing with text scenarios, graphs can extract the features from non-structural data by modeling a set of objects (nodes) and their relationships (edges). In particular, we can consider each variable in the text as a node and the dependencies as edges for the sequence labeling task. Marcheggiani and Titov (2017) present a syntactic GCN to solve the problem of semantic role labeling. Ding et al. (2019) utilize a multi-graph structure to capture the information that the gazetteers offer. In addition, the graph neural network based on the domain lexicon is used to learn the local composition features for medical domain CWS (Du et al., 2020).

3 Proposed Framework

The framework of LBGCN is illustrated in Figure 1. It mainly consists of two parts: an encoder-decoder layer and a GCN. In the first part, we utilize the Transformer as the encoder and the Dense as the decoder. The Transformer encoder adopts the PLMs

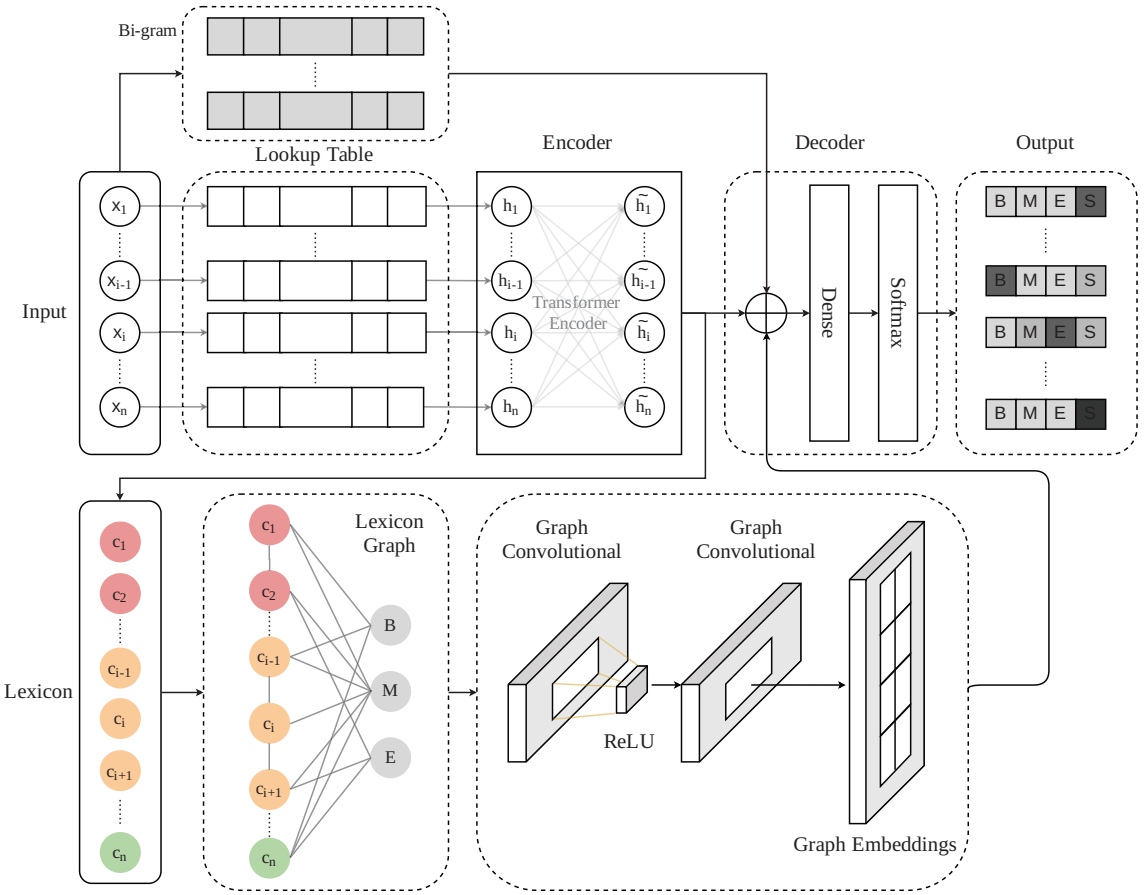

Figure 1: The illustration of the proposed framework. Continuous nodes with the same color denote a Chinese word in the pre-defined lexicon. The gray nodes of "B", "M" and "E" indicate the three additional nodes, named Begin, Middle, and End, respectively.

(e.g., BERT and RoBERTa) which contain rich pre-trained knowledge to train. The pre-trained knowledge can effectively help the model alleviate the problem of OOV word recognition. In the second part, a GCN based on the pre-defined lexicon is built. The generated graph embeddings can make up the deficiency of contextual information from candidate words. In addition, the proposed framework that is integrated with bi-gram features and multiple contextual features can improve the performance of CWS.

Following previous studies (Xue, 2003), we regard the CWS as the character-based sequence labeling task. The framework predicts a tag that represents the position in a word for each character (e.g., tag "B" represents the first character in a word). The process of LBGCN to find the most possible path $\hat{\mathcal{Y}}$ can be formalized as:

$$\hat{\mathcal{Y}} = \underset{\mathcal{Y} \in \mathcal{T}^N}{argmax}\, p\left(\mathcal{Y}|\mathcal{X}\right) \quad (1)$$

where $\mathcal{T}$ denotes the set of all types of segmentation labels, and N is the length of the input sentence $\mathcal{X}$.

The rest of this section describes the architecture of the encoder-decoder layer, the construction of the lexicon graph, and how it is integrated with the GCN, respectively.

3.1 Encoder and Decoder

Transformer Encoder Recently, there are several PLMs (e.g., BERT and RoBERTa) that have shown state-of-the-art performance of many NLP tasks. In particular, a modified method based on RoBERTa model is built for the Chinese NLP tasks (Cui et al., 2019). With the previous success on PLMs, we adopt the main architecture (Transformer) as the encoder of our proposed framework, which can straightforwardly leverage the pre-trained knowledge from PLMs for the Transformer encoder because of the similar structure.

The PLM is trained for predicting the word in general. To transfer the pre-trained knowledge into the CWS, we need to fine-tune the PLM by the

annotated corpus of CWS. Given an input sentence $\mathcal{X} = x_1...x_{i-1}x_i...x_n$ from the training data, the input sentence is converted to the corresponding vector embeddings $H = [h_1...h_{i-1}h_i...h_n]$ in the "Lookup Table" layer, where $H \in \mathbb{R}^{N*d_{model}}$, N and d_{model} represent the length and the same dimensions with the PLM. To be consistent with the pre-trained process, two tags ("[CLS]" and "[SEP]") are added to the beginning and the end of each sentence, respectively.

Given an input vector sentence $H \in \mathbb{R}^{N \times d_{model}}$, the Transformer encoder utilizes self-attention layers to extract the contextual feature for each character. The self-attention layer adopts "Scaled Dot-Product Attention" to compute representation.

$$Q, K, V = HW^Q, HW^K, HW^V \quad (2)$$

$$Attn\,(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3)$$

where Q, K, V represents a query and a set of key-value pairs through a linear transformation respectively, the matrices $W^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W^K \in \mathbb{R}^{d_{model} \times d_k}$, $W^V \in \mathbb{R}^{d_{model} \times d_v}$ are trainable parameters, and d_k is the dimension of K.

Instead of performing a single-head attention function, the Transformer encoder uses the multi-head self-attention layer in order to extract contextual features from different representation spaces and utilizes feed forward network (FFN) to enhance representation ability. Assuming the input of the multi-head self-attention layer is H, the output $\tilde{H}$ is calculated by

$$Z = LN\,(H + MultiHead(H)) \quad (4)$$

$$\tilde{H} = LN\,(Z + FFN(Z)) \quad (5)$$

where "LN" indicates the layer normalization (Ba et al., 2016).

Dense Decoder A dense layer with $W^D \in \mathbb{R}^{d_{model} \times T_n}$ converts hidden dimensions to the 4-tag set $\mathcal{T} = \{B, M, E, S\}$, where T_n presents the size of the tag sets ($T_n = 4$). After linear mapping, the framework adopts the function $Softmax$ and the greedy search for decoding. In previous studies, many kinds of research adopt the CRF as the decoder layer to improve the performance of sequence labeling tasks (Lample et al., 2016). However, the CRF layer has larger time complexity and space complexity for CWS (Duan and Zhao, 2020). For practicality, the proposed framework utilizes the lightweight function $Softmax$ as the decoder layer and also achieves competitive performance compared with other studies using the CRF.

$$p(x) = Softmax(\tilde{H} \cdot W^D + b) \quad (6)$$

The training step of the framework is to minimize the errors by solving the following optimization function:

$$\min_{\Theta^t, \Theta^g} \mathcal{J}_{seg}(y(x)|p(x; \Theta^t, \Theta^g)) \quad (7)$$

where $y(x)$ denotes the true labels on the annotated corpus, Θ^t and Θ^g are all trainable parameters in the transformer layer and GCN, respectively, and the loss function $\mathcal{J}_{seg}$ is given by:

$$\mathcal{J}_{seg}(y(x)|p(x)) = -\sum_x y(x) log p(x) \quad (8)$$

3.2 Lexicon-Based GCN

Lexicon-Based Construction The bottom part of the Figure 1 starts with a lexicon and we construct the graph by the pre-defined lexicon. Given the input sentence $\mathcal{X} = x_1...x_{i-1}x_i...x_n$, the graph utilizes a pre-defined lexicon to extract candidate words in the sentence after the Transformer encoder. For example, $\mathcal{X}$ = ["水仙花是草本植物"] (Daffodils are herbaceous plant) consists of 8 Chinese characters, and the word list $\mathcal{L}$ = ["水仙花"(daffodils), "是"(are), "草本"(herbaceous), "植物"(plant)] is obtained from the lexicon. The lexicon-based graph is defined as $G := (V, E)$, where V and E are the sets of nodes and edges, respectively. Each character is represented as a node in the graph and adjacent nodes connect to each other by undirected edges for capturing the contextual information. The set of these undirected edges is E_c. Besides, we integrate three additional nodes $V_d = (V_B, V_M, V_E)$ with the character set of nodes V_c, and the entire set of nodes is $V = V_c \cup V_d$. To extract the information of the word boundary, we also build edges between candidate words $w_i = c_1...c_n, w_i \in \mathcal{L}$ and additional nodes V_d. The entire set of edges is $E = E_c \cup E_d$, where E_d represents the set of edges between candidate words and additional nodes. The 1st character c_1 in the candidate word connects to the node V_B and V_M, and the last character c_n connects to the node V_M and V_E. For instance, the candidate word "水仙花" (daffodils) consists of three characters "水(water), 仙(fairy) and 花(flower)". In particular, the character node "水" (water) connects to the

Benchmarks	MSR		PKU		AS		CITYU		CTB6	
	TRAIN	TEST	TRAIN	TEST	TRAIN	TEST	TRAIN	TEST	TRAIN	TEST
CHAR #	4,050K	184K	1,826K	173K	8,368K	198K	2,403K	68K	1,156K	134K
WORD #	2,368K	107K	1,110K	104K	5,500K	123K	1,456K	41K	701K	82K
Cross-domain	LITERATURE		COMPUTER		MEDICINE		FINANCE			
CHAR #	50K		54K		51K		53K			
WORD #	35K		35K		31K		33K			

Table 1: The size of the benchmark, the top blocks indicate the CWS benchmarks (Bakeoff-2005 and CTB6) and the bottom blocks indicate the cross-domain CWS datasets (SIGHAN-2010). Note that the cross-domain datasets do not contain the training sample, so we use the "PKU" which is the most similar to them as the training data.

node V_B and V_M. The character node "仙" (fairy) only connects to the node V_M. The character node "花" (flower) connects to the node V_M and V_E. The construction of the lexicon graph is illustrated in Figure 1.

GCN After the construction of the lexicon-based graph, we utilize a GCN (Kipf and Welling, 2016) to encode the graph G.

$$\begin{aligned} \tilde{A} &= A + I_N, \tilde{D}_{ii} = \sum_j \tilde{A}_{ij} \\ \hat{H}^{(l+1)} &= \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}\hat{H}^{(l)}W^{(l)}\right) \end{aligned} \tag{9}$$

Here, A is the adjacency matrix of the undirected graph G. I_N is the identity matrix and $W^{(l)}$ is a layer-specific trainable weight matrix. $\sigma(\cdot)$ denotes the ReLU activation function. $\hat{H}^{(l)} \in \mathbb{R}^{N\times D}$ is the matrix of hidden states in the l_{th} layer; $H(0) = X$. GCN mainly consists of two matrices. One is the symmetric normalized Laplacian matrix $\tilde{A}$. The other is the layer-specific trainable weight matrix $W^{(l)}$. The GCN can extract the features from the lexicon-based graph. In addition, the weight matrix adopts the random initialization and the learning rate of this layer is different from the transformer encoder.

4 Experiments

4.1 Datasets and Settings

To verify the improvement of our proposed framework LBGCN, we do comparative experiments on both benchmarks (Bakeoff-2005 (Emerson, 2005) and CTB6) and cross-domain datasets (SIGHAN-2010) (Zhao and Liu, 2010). The size of the benchmark is shown in Table 1. We randomly pick 10% sentences from the training data as the development data for tuning hyper-parameters. For the experiments on the cross-domain datasets, we follow the settings of the "PKU" dataset. For consistency, we pre-process the unsegmented sentences, which is similar to the previous paper (Cai et al., 2017). The evaluation values for CWS are F-score and R_{oov}.

Parameters	
Hidden states	768
GCN hidden states	[128,**256**,768]
Bi-gram embeds	128
Learning rate	[2e-4,1e-4,**2e-5**]
GCN learning rate	[**1e-3**,1e-4,1e-5]
Batch size	[64,128,**256**]
Dropout	[0.1, **0.2**, 0.4]
GCN dropout	[0.1,**0.2**,0.4]
Hidden layers	12
Epochs	20

Table 2: The crucial hyper-parameters and search ranges.

We utilize three mainstream PLMs for training the Transformer encoder, including XLNET-BASE (Yang et al., 2019b; Cui et al., 2020), BERT-BASE and ROBERTA-WWM (Cui et al., 2019).[2] To fine-tune PLMs, we tune a few crucial hyper-parameters with the development sets for the model. The hyper-parameters and search ranges are shown in Table 2. We deploy the model on the same device (GPU environment: Nvidia Tesla V100).

4.2 Experimental Results

This section first reports the results of LBGCN with different configurations on five benchmarks and comparison with existing models. Then it describes the effect of LBGCN in the cross-domain scenario.

Results on Benchmarks In the benchmark scenario, we verify the validity on Bakeoff-2005 and CTB6 by comparing LBGCN with three different PLMs, i.e., XLNET, BERT, and RoBERTa. As

[2]The PLMs are available at https://huggingface.co/models

	PKU		MSR		AS		CITYU		CTB6	
	F	R_{oov}	F	R_{oov}	F	R_{oov}	F	R_{oov}	F	R_{oov}
MA ET AL. (2018)	96.1	78.8	98.1	80.0	96.2	70.7	97.2	87.5	96.7	85.4
GONG ET AL. (2019)	96.15	69.88	97.78	64.20	95.22	77.33	96.22	73.58	97.26	83.89
MENG ET AL. (2019)	96.7	-	98.3	-	96.7	-	97.9	-	-	-
QIU ET AL. (2020)	96.41	78.91	98.05	78.92	96.44	76.39	96.91	86.91	96.99	87.00
HUANG ET AL. (2020)	96.85	82.35	98.29	81.75	-	-	-	-	97.56	88.02
TIAN ET AL. (2020)	96.53	85.36	98.40	84.87	96.62	79.64	97.93	90.15	97.25	88.46
XLNET	96.62	87.81	98.16	79.83	96.68	79.68	97.74	89.57	97.47	89.43
XLNET+LBGCN	96.92	88.05	98.26	82.39	96.92	79.59	97.77	89.96	97.49	87.85
BERT	96.85	88.15	98.29	79.92	96.99	**83.54**	98.12	91.21	97.75	89.88
BERT+LBGCN	97.00	88.58	98.42	80.24	**97.03**	80.35	**98.13**	91.74	**97.83**	89.12
ROBERTA	97.00	89.80	98.42	84.59	96.80	79.15	98.09	**92.32**	97.70	89.33
ROBERTA+LBGCN	**97.21**	**90.03**	**98.52**	**86.13**	96.87	79.22	**98.13**	91.87	97.79	**90.15**

Table 3: Results and comparison with existing models on the benchmarks Bakeoff-2005 and CTB6, LBGCN is trained based on different PLMs and components. The best values are bolded for each column.

	LIT.	COM.	MED.	FIN.	AVG.
LIU ET AL. (2014)	92.49	94.07	92.63	95.54	93.68
CHEN ET AL. (2015B)	92.89	93.71	92.16	95.20	93.49
CAI ET AL. (2017)	92.90	94.04	92.10	95.38	93.61
HUANG ET AL. (2017)	94.33	93.99	92.26	95.81	94.10
ZHAO ET AL. (2018)	93.23	95.32	93.73	95.84	94.53
ZHANG ET AL. (2018)	94.76	94.70	94.18	96.06	94.93
HUANG ET AL. (2020)	96.13	96.08	95.21	96.82	96.06
XLNET	95.87	96.07	95.09	96.72	95.93
BERT	96.16	95.57	95.38	96.89	96.00
ROBERTA	96.20	96.11	95.44	96.75	96.12
XLNET-LBGCN	96.09	**96.33**	95.21	96.88	96.12
BERT-LBGCN	**96.51**	95.59	95.66	97.04	96.20
ROBERTA-LBGCN	96.49	96.13	**95.66**	**97.14**	**96.33**

Table 4: Results and comparison with existing models on the cross-domain datasets SIGHAN-2010, where "LIT., COM., MED., and FIN." represent the domain of literature, computer, medicine, and finance, respectively. The best values are bolded for each column.

ID	Bi-gram	GCN	PKU		MSR	
			F	R_{oov}	F	R_{oov}
1	×	×	97.00	89.80	98.42	84.59
2	√	×	-0.02	+0.17	-0.01	+0.42
3	×	√	+0.15	**+0.25**	**+0.10**	**+1.54**
4	√	√	**+0.21**	+0.23	+0.08	+0.85

Table 5: Ablation experiments. The baseline (ID:1) is based on the RoBERTa model.

shown in Table 3, three baseline models which utilize different PLMs to train the Transformer encoder of our proposed framework, are represented as "XLNET", "BERT, and "ROBERTA", respectively. There are three observations drawn from the results. First, The framework which integrates with our proposed LBGCN outperforms the baseline models for all 5 datasets in terms of F-scores and for the majority of datasets in terms of R_{oov}. Second, the proposed LBGCN make small improvements in some datasets, whereas considerable improvements are shown in the other datasets. The extent of improvement of LBGCN does not depend on PLMs which the encoder utilizes. For instance, when training the Transformer encoder fine-tuning the RoBERTa, LBGCN improves the F-score on the PKU dataset from 97.00 to 97.21 and R_{oov} from 89.80 to 90.03. With XLNET or BERT as the baseline PLM, the improvement of LBGCN on F-scores and R_{oov} are still decent. Lastly, the methods that fine-tune the RoBERTa can achieve better performance on most benchmarks, and our proposed LBGCN utilizes the GCN to get further promotion on the baseline model which already achieves competitive performance of CWS.

Besides, we compare the proposed framework with existing methods. The comparison is also presented in Table 3, where the proposed framework LBGCN based on the BERT or RoBERTa outperforms all existing models in terms of the F-scores on all benchmarks.

Results on Cross-Domain CWS Domain variance is important to affect the performance of word semgenters. To demonstrate the efficiency of LBGCN, we also run frameworks with and without the LBGCN in the cross-domain scenario. Table 4 reports the results in F-score, which shows a similar trend as that in Table 3, where LBGCN outperforms baselines in all 5 domains. And the

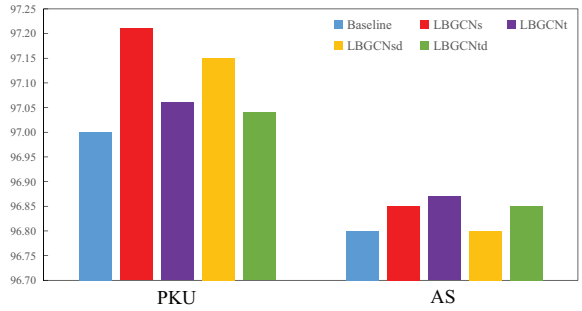

Figure 2: The F-scores of LBGCN using four different lexicons on two benchmark datasets, where "PKU" is the simplified Chinese dataset and "AS" is the traditional Chinese dataset.

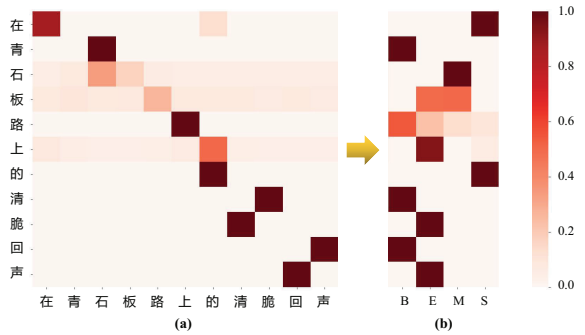

Figure 3: Heatmaps of weights that learn from the Transformer encoder (a), and (b) the tags from the decoder. Each row corresponds to a character in the input sentence. Higher weights are visualized with darker colors.

framework ROBERTA-LBGCN achieves state-of-the-art performance in terms of average F-score. In particular, the XLNET has better performance on the "Computer" domain, and the BERT has better performance on the "Literature" domain. In general, our proposed LBGCN mechanism can effectively improve performance in the cross-domain scenario, and all LBGCNs fine-tuning different PLMs achieve competitive performance, compared with existing methods.

4.3 Effect of Using Different Lexicons

LBGCN utilizes a general way of integrating lexicon for CWS. To analyze the effect of methods using different lexicons, we adopt four different lexicons into the ROBERTA-LBGCN and compare them with the baseline model, as shown in Figure 2. Four lexicons consist of two simplified Chinese dictionaries[3] and two traditional Chinese dictionaries[4]. Particularly, two simplified Chinese dictionaries consist of a basic version "LBGCNs" (red) and a modified version "LBGCNsd" (yellow), respectively. Similarly, two traditional Chinese dictionaries are also a basic version "LBGCNt" (purple), and a modified version "LBGCNtd" (green).

As shown in Figure 2, the performance of using the four lexicons are all better than those of the baseline models on both the "PKU" and "AS" dataset, indicating the efficiency of our proposed lexicon-based framework. The framework using the basic simplified Chinese dictionary (red) achieves the biggest improvement on the "PKU" and the one using the basic traditional Chinese dictionary (purple) achieves the biggest improvement on the "AS".

[3] https://github.com/fxsjy/jieba/blob/master/jieba/dict.txt

[4] https://github.com/L706077/jieba-zh_TW/blob/master/jieba/dict.txt

4.4 Ablation Study

LBGCN integrates two additional components for CWS, including the bi-gram features and the lexicon-based GCN. To analyze the effect of LBGCN with respect to different components, we do an ablation experiment based on the ROBERTA PLM which performs better for both "PKU" and "MSR" benchmarks and the results are shown in Table 5. Table 5 shows that the GCN (ID:3,4) effectively improves the performance of the baseline model on "PKU" and "MSR", and it also alleviates the issue of OOV words, indicating the effectiveness of our proposed framework. While the GCN that integrates with the bi-gram component (ID:4) achieves progress on the "PKU" from +0.15 to +0.21, it hurts the R_{oov}. A single bi-gram component (ID:2) hardly affects the F-score but it can improve the recall of OOV words. In terms of the results in Table 5, the bi-gram and GCN boost the performance considerably.

4.5 Case Study

To investigate how the proposed framework learns from the lexicon-based GCN, we choose an example input sentence "在/青石板/路上/的/清脆/回声" (The clear echo on the flagstone road) in the literature domain scenario as a case study. In this sentence, the n-gram "青石板/路" (flagstone road) is the road that is made of a special kind of stone and always occurs in the Chinese literature. However, the split "板路" is short for the plate circuit. The baseline model may confuse this case because of the character diversity in Chinese. Intuitively, the "青石板" is in the pre-defined lexicon and an

undirected graph is constructed with the information of the word boundary. Then the lexicon-based GCN capture this information and integrates the graph embeddings with the original hidden states. The integrated embeddings with the knowledge of lexicon information are transferred into correct tags by the decoder. In Figure 3, we visualize the resulted weights that learn from the basic Transformer encoder (a), as well as from the final tagger (b).

In addition, in another case, "地瓜/粥" (sweet potato congee) represents a Chinese food and "粥" (congee) should be regarded as a single suffix word on "PKU" segmentation criterion. The baseline model cannot segment it correctly, because it keeps the superabundant pre-trained knowledge of PLMs. In the LBGCN, "地瓜粥" does not exist in the lexicon but "地瓜" is a lexicon word. The LBGCN constructs this relationship in the graph to distinguish important n-grams and improves performance accordingly for CWS.

5 Conclusion

To make up for the insufficiency of previous methods that fine-tune PLMs, in this paper, we propose a lexicon-based graph convolutional network to better transfer pre-trained knowledge from PLMs into the CWS. Our proposed framework LBGCN provides baseline models with the information of word boundary and contextual information, in addition to preserving the merits of baseline models in applying PLMs. In summary, the advantages of LBGCN are threefold. First, the novel framework does not rely on a particular PLM, and it can get further promotion on all baseline methods based on three mainstream PLMs for CWS. Second, the results on extensive experiments show that LBGCN achieves competitive performance on the CWS benchmarks, compared with previous methods. Third, further experiments and analyses demonstrate the effectiveness of LBGCN in the cross-domain scenario as well as when using different lexicons and components. Overall, this paper presents an elegant way to use a graph neural network for CWS and enhance fine-tuning CWS methods. For future work, we plan to investigate other sequence labeling tasks using the same methodology.

Acknowledgments

We sincerely thank the reviewers for their insightful comments and suggestions to improve the quality of the paper. The authors gratefully acknowledge the financial support provided by the National Key Research and Development Program of China (2020AAA0108004) and the National Natural Science Foundation of China under(No.U1936109).

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *stat*, 1050:21.

Deng Cai and Hai Zhao. 2016. Neural word segmentation learning for chinese. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–420, Berlin, Germany. Association for Computational Linguistics.

Deng Cai, Hai Zhao, Zhisong Zhang, Yuan Xin, Yongjian Wu, and Feiyue Huang. 2017. Fast and accurate neural word segmentation for chinese. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 608–615, Vancouver, Canada. Association for Computational Linguistics.

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, and Xuanjing Huang. 2015a. Gated recursive neural network for chinese word segmentation. 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 1744–1753, Beijing, China. Association for Computational Linguistics.

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuanjing Huang. 2015b. 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, Lisbon, Portugal. Association for Computational Linguistics.

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, Vancouver, Canada. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668, Online. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.

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*.

Ning Ding, Dingkun Long, Guangwei Xu, Muhua Zhu, Pengjun Xie, Xiaobin Wang, and Haitao Zheng. 2020. Coupling distant annotation and adversarial training for cross-domain Chinese word segmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6662–6671, Online. Association for Computational Linguistics.

Ruixue Ding, Pengjun Xie, Xiaoyan Zhang, Wei Lu, Linlin Li, and Luo Si. 2019. A neural multi-digraph model for chinese ner with gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1462–1467.

Jinlian Du, Wei Mi, and Xiaolin Du. 2020. Chinese word segmentation in electronic medical record text via graph neural network-bidirectional lstm-crf model. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 985–989. IEEE.

Sufeng Duan and Hai Zhao. 2020. Attention is all you need for Chinese word segmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3862–3872, Online. Association for Computational Linguistics.

Thomas Emerson. 2005. The second international chinese word segmentation bakeoff. In *Proceedings of the Fourth SIGHAN workshop on Chinese Language Processing*, pages 123–133, Jeju Island, Korea.

Jingjing Gong, Xinchi Chen, Tao Gui, and Xipeng Qiu. 2019. Switch-lstms for multi-criteria chinese word segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6457–6464.

Kaiyu Huang, Degen Huang, Zhuang Liu, and Fengran Mo. 2020. A joint multiple criteria model in transfer learning for cross-domain Chinese word segmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3873–3882, Online. Association for Computational Linguistics.

Shen Huang, Xu Sun, and Houfeng Wang. 2017. Addressing domain adaptation for chinese word segmentation with global recurrent structure. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 184–193.

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

Zhen Ke, Liang Shi, Songtao Sun, Erli Meng, Bin Wang, and Xipeng Qiu. 2021. Pre-training with meta learning for chinese word segmentation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5514–5523.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

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.

Yijia Liu, Yue Zhang, Wanxiang Che, Ting Liu, and Fan Wu. 2014. Domain adaptation for crf-based chinese word segmentation using free annotations. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 864–874, Doha, Qatar. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ji Ma, Kuzman Ganchev, and David Weiss. 2018. State-of-the-art chinese word segmentation with bi-lstms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4902–4908.

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.

Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, Qinghong Han, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for chinese character representations. In *Advances in Neural Information Processing Systems*, pages 2742–2753.

Wenzhe Pei, Tao Ge, and Baobao Chang. 2014. Max-margin tensor neural network for chinese word segmentation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 293–303, Baltimore, Maryland, USA. Association for Computational Linguistics.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING '04)*, pages 562–

568, Geneva, Switzerland. Association for Computational Linguistics, Association for Computational Linguistics.

Xipeng Qiu, Hengzhi Pei, Hang Yan, and Xuan-Jing Huang. 2020. A concise model for multi-criteria chinese word segmentation with transformer encoder. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2887–2897.

Yuanhe Tian, Yan Song, Fei Xia, Tong Zhang, and Yonggang Wang. 2020. Improving Chinese word segmentation with wordhood memory networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8274–8285, Online. Association for Computational Linguistics.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter for sighan bakeoff2005. In *Proceedings of the Fourth SIGHAN workshop on Chinese Language Processing*, pages 168–171, Jeju Island, Korea.

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.

Xiaobin Wang, Deng Cai, Linlin Li, Guangwei Xu, Hai Zhao, and Luo Si. 2019. Unsupervised learning helps supervised neural word segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7200–7207.

Nianwen Xue. 2003. Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, 8(1):29–48.

Jie Yang, Yue Zhang, and Fei Dong. 2017. Neural word segmentation with rich pretraining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 839–849, "Vancouver, Canada". "Association for Computational Linguistics".

Jie Yang, Yue Zhang, and Shuailong Liang. 2019a. Subword encoding in lattice lstm for chinese word segmentation. 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 2720–2725.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2018. Improving neural machine translation with conditional sequence generative adversarial nets. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1346–1355.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32:5753–5763.

Yuxiao Ye, Weigang Li, Yue Zhang, Likun Qiu, and Jian Sun. 2019. Improving cross-domain chinese word segmentation with word embeddings. 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 2726–2735.

Qi Zhang, Xiaoyu Liu, and Jinlan Fu. 2018. Neural networks incorporating dictionaries for chinese word segmentation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Hai Zhao, Deng Cai, Changning Huang, and Chunyu Kit. 2019. Chinese word segmentation: Another decade review (2007-2017). *arXiv preprint arXiv:1901.06079*.

Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2010. A unified character-based tagging framework for chinese word segmentation. *ACM Transactions on Asian Language Information Processing*, 9(2):1–32.

Hai Zhao and Chunyu Kit. 2008. Unsupervised segmentation helps supervised learning of character tagging for word segmentation and named entity recognition. In *The Sixth SIGHAN Workshop on Chinese Language Processing*, pages 106–111, Hyderabad, India.

Hongmei Zhao and Qiu Liu. 2010. The cips-sighan clp2010 chinese word segmentation backoff. In *CIPS-SIGHAN Joint Conference on Chinese Language Processing*.

Lujun Zhao, Qi Zhang, Peng Wang, and Xiaoyu Liu. 2018. Neural networks incorporating unlabeled and partially-labeled data for cross-domain chinese word segmentation. In *IJCAI*, pages 4602–4608.

Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. 2013. Deep learning for chinese word segmentation and pos tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 647–657, Seattle, Washington, USA. Association for Computational Linguistics.

Hao Zhou, Zhenting Yu, Yue Zhang, Shujian Huang, Xinyu Dai, and Jiajun Chen. 2017. Word-context character embeddings for chinese word segmentation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 760–766, Copenhagen, Denmark. Association for Computational Linguistics.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81.