Natural Language Processing (2024), 1–23
doi:10.1017/nlp.2024.49

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

Chinese word segmentation with heterogeneous graph convolutional network

Xuemei Tang[1,2], Qi Su[2,3] and Jun Wang[1,2]

[1]Department of Information Management, Peking University, Beijing, China, [2]Digital Humanities Center of Peking University, Beijing, China, and [3]School of Foreign Languages, Peking University, Beijing, China
Corresponding author: Qi Su; Email: sukia@pku.edu.cn

(Received 2 May 2023; revised 16 August 2024; accepted 16 August 2024)

Abstract

Recently, deep learning methods have achieved remarkable success in the Chinese word segmentation (CWS) task. Some of them enhance the CWS model by utilizing contextual features and external resources (e.g., sub-words, lexicon, and syntax). However, existing approaches fail to fully use the heterogeneous features and their structural information. Therefore, in this paper, we propose a heterogeneous information learning framework for CWS, named heterogeneous graph neural segmenter (HGNSeg), which exploits heterogeneous features with the graph convolutional networks and the pretrained language model. Experimental results on six benchmark datasets (e.g., SIGHAN 2005 and SIGHAN 2008) confirm that HGNSeg can effectively improve the performance of CWS. Importantly, HGNSeg also demonstrates an excellent ability to alleviate the out-of-vocabulary (OOV) issue in cross-domain scenarios.

Keywords: Chinese word segmentation; graph convolutional network; pretrained language model

1. Introduction

Chinese sentences consist of consecutive characters without delimiters between words. Therefore, Chinese word segmentation (CWS) is generally considered to be a fundamental and essential task of some Chinese natural language processing (NLP) downstream tasks, such as information extraction, machine translation, and text classification.

Recently, neural network approaches, such as long short-term memory (LSTM) (Chen *et al.* 2015) and Transformer (Tian *et al.* 2020b; Huang *et al.* 2021), have proven effective in CWS. Various methods have been proposed to integrate contextual information and external resources into neural networks, for example, Margatina *et al.* (2019), Liu *et al.* (2018), and Tian *et al.* (2020a) incorporated lexicon and syntax knowledge for CWS using attention mechanisms and a multi-task framework.

As is well known, dependency trees provide word-to-word dependency information, offering a solution to alleviate the challenge of segmentation ambiguity. For instance, consider the input sentence depicted in Fig. 1, which can be segmented as "武汉市(Wuhan city)/长江(Yangtze River)/大桥(Bridge)/建成(built)" or "武汉(Wuhan)/市长(Mayor)/江大桥(Jiang Daqiao)/建成(built)". The dependency tree captures the syntactic relations in this sentence, revealing that "武汉市(Wuhan city)" depends on "大桥(Bridge)". This dependency relation helps mitigate ambiguity in the highlighted portion of the sentence. As mentioned by Huang and Xue (2012), besides ambiguity segmentation, the out-of-vocabulary (OOV) problem is another issue that needs to be concerned with CWS. Many studies have used lexicon and n-gram lexicon as features

to alleviate the OOV problem. While some studies have attempted to leverage both dependency tree information and lexicons (Tian et al., 020a), they often overlook the heterogeneous nature of these features and the distinctive structures inherent in dependency trees. This oversight may be attributed to the absence of direct and successful methods for fully integrating heterogeneous linguistic features (e.g., lexicon, n-gram, syntax) and their structural information into neural networks.

Therefore, in this paper, we aim to explore appropriate methods that can integrate heterogeneous external knowledge and their structures for CWS. Taking inspiration from the work of Nie *et al.* (2022) and Sui *et al.* (2019), they employed graph neural networks (GNNs) to encode heterogeneous features for Chinese named entity recognition (NER), we introduce a heterogeneous feature learning framework for CWS called HGNSeg. Specifically, we build heterogeneous linguistic features (e.g., lexicon, n-gram lexicon, and dependency tree) and their structural information as a heterogeneous graph, where nodes integrate different types of features based on different types of edges absorbing information from neighboring nodes. We represent all characters in the input sequence, the candidate words from the lexicon, and n-grams from the n-gram lexicon as nodes and then add different types of edges between nodes. Characters establish connections with each other based on the dependency relation, and the characters link to words and n-grams in accordance with their relative positions within these linguistic units.

To summarize, our contributions are as follows:

- We introduce a novel framework for CWS utilizing graph convolutional networks (GCNs) to encode heterogeneous features.
- We employ GCNs to encode a variety of heterogeneous linguistic features, including then lexicon, the n-gram lexicon, and the dependency tree. Our design incorporates various types of nodes and edges to seamlessly integrate these features and their respective structural information.
- We evaluate HGNSeg on six benchmark CWS datasets and six cross-domain CWS datasets. Experimental results on benchmark datasets show that HGNSeg can improve the performance of CWS by integrating heterogeneous linguistic features. Experimental results on cross-domain corpora also confirm that HGNSeg effectively alleviates the OOV issue in the cross-domain scenario.

2. Related work

2.1 Chinese word segmentation

CWS is a typical task in Chinese language processing tasks. There are various modeling approaches for the CWS task, for example, early lexicon-based matching methods and statistical methods. For example, Huang (1995) and Sproat and Shih (1990) used mutual information to measure the binding strength in a string. Subsequently, Xue (2003) first modeled CWS as a sequence labeling task, assigning each character in the input sentence one of the four tags–LL, RR, MM, or LR–based on its position within a word. Huang *et al.* (2007) proposed to model CWS as a unary word boundary decision task, which simplifies the complexity of CWS and offers a promising solution for addressing issues related to domain adaptation and robustness. All of these methods perform well and have advantages in improving ambiguous segmentation and OOV problems. However, all of them require a large-scale labeled dataset for support. Therefore, based on Huang's work, Li *et al.* (2012) further proposed to use active learning to address the need for labeled data. By selecting the most informative boundaries for manual labeling and using automatic labeling for less informative boundaries, this method reduces the cost of manual labeling.

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

Recently, numerous neural network approaches have demonstrated remarkable success in CWS, eliminating the need for intricate feature engineering. For instance, Liu *et al.* (2018) employed a convolutional neural network (CNN) to capture character features. LSTM networks, known for their capability to learn long-distance dependencies in sentences, were utilized by Chen *et al.* (2015), Zhang *et al.* (2016), Ma *et al.* (2018), and Margatina *et al.* (2019) to extract contextual features for CWS. The Transformer, incorporating the self-attention mechanism, surpasses LSTM in capturing long-distance dependencies and parallel computing. Qiu *et al.* (2020) employed the Transformer as an encoder to extract context-aware information for multi-criteria CWS. Notably, pretrained language model (PLM) has recently shown exceptional performance in CWS, with Tian *et al.* (2020b) and Huang *et al.* (2021) leveraging BERT (Devlin *et al.* 2019), Zen (Diao *et al.* 2019), and RoBERTa (Liu *et al.* 2019) to learn text representations for CWS without extensive feature engineering.

Most studies further enhance neural models by integrating external knowledge (e.g., lexicon, syntax) and learning contextual information (e.g., n-grams). CWS aims to find out the boundaries of words in sentences, so the lexicon is one type of effective external resource for CWS. Margatina *et al.* (2019), for example, introduced an attention mechanism on LSTM to integrate word-level information for CWS. For cross-domain CWS, challenges such as a scarcity of annotated data and the presence of OOV instances are formidable. Ding *et al.* (2020) addressed these issues by leveraging automatic CWS toolkits, mutual information methods, term frequency–inverse document frequency, and other techniques to automatically construct a lexicon for the target domain. This lexicon, along with the maximum matching algorithm, facilitated the segmentation of target domain data to generate labeled data. Adversarial training was then employed to minimize the dissimilarity between the source and target domain representations. In a different approach, Zhang *et al.* (2018) integrated lexicon by constructing feature vectors for each character with several predefined templates. Meanwhile, Liu *et al.* (2021) used external lexicon knowledge to enhance BERT by a lexicon adapter layer, demonstrating robust performance across CWS, part-of-speech (POS) tagging, and NER tasks.

N-grams represent a rich contextual resource for CWS (Kurita, Kawahara, and Kurohashi, 2017; Shao *et al.* 2017; Tian *et al.* 2020a). In a notable example, Tian *et al.* (2020b) integrated wordwood information into a neural segmenter using key-value networks, achieving state-of-the-art (SOTA) performance at the time. Furthermore, the outcomes of CWS are frequently leveraged to enhance syntax parsing (Shen *et al.* 2022). Intuitively, incorporating syntax knowledge can also improve CWS models. Tian *et al.* (2020a), for instance, utilized syntax knowledge generated by existing NLP toolkits with an attention mechanism to enhance the joint CWS and POS tagging task.

While these methods yield satisfactory performance, they often overlook the heterogeneity of knowledge. Combining multiple types of knowledge requires distinct methodologies for each type. Furthermore, these approaches tend to neglect the structural information inherent in knowledge, such as the dependencies present in the structure of dependency trees.

2.2 Graph convolutional networks

GNNs have received significant attention as a novel class of neural networks (Defferrard, Bresson, and Vandergheynst, 2017; Kipf and Welling, 2017). These networks perform computations on graphs and possess the capability to retain the global structural information of the graph within its embedding, making them particularly effective for graphs with rich relational structures (Yao, Mao, and Luo, 2018).

GCNs, a subset of GNNs, operate similarly to CNN, with the distinction that GCNs utilize convolution operations on graphs (Kipf and Welling, 2017). In GCNs, each node in the graph updates its embedding by incorporating relevant information from its neighboring nodes. This mechanism has found success in various NLP tasks for encoding external knowledge with

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

intricate structures. For example, Hu *et al.* (2021) employed GCNs to integrate knowledge base information, enhancing fake news detection. In text classification tasks, Michael *et al.* (2016) demonstrated that GCNs outperform traditional CNN models. Additionally, GCNs have been utilized to encode syntax graphs, improving semantic role labeling models (Marcheggiani and Titov, 2017) and to integrate syntax information into neural machine translation (NMT) models, enriching word-level representations (Bastings *et al.* 2017).

A heterogeneous graph encompasses nodes and edges of various types. For instance, Hu *et al.* (2019) transformed relation structures and additional features (e.g., topics, entities) between short texts into heterogeneous graphs. They employed a graph attention network (GAT) to extract heterogeneous information, strengthening their text classification model. Similarly, Nie *et al.* (2022) constructed multi-granularity lexicons as heterogeneous graphs, utilizing GCNs to encode them for NER. Intuitively, incorporating heterogeneous features into CWS using heterogeneous graphs seems promising. These works prove that GCNs are an effective method for integrating linguistic features and their structure information, as long as they can be represented as graphs (Bastings *et al.* 2017).

GCNs have been applied in the CWS task. Zhao *et al.* (2020) utilized GCNs to encode multi-granularity structural information, enhancing the performance of the joint CWS and POS tagging task. In the domain of electronic medical record text segmentation, Du *et al.* (2020) employed a GNN to learn local structural features based on domain lexicons. Additionally, Huang *et al.* (2021) incorporated lexicons into a PLM using GCNs, improving CWS performance and enhancing the robustness of cross-domain CWS. Notably, Yu *et al.* (2022) proposed a lexicon-augmented GCN for cross-domain CWS, encoding candidate words' boundary information to enhance the CWS model.

However, existing CWS studies have not fully capitalized on the fact that GCNs can encode structural information. They have also overlooked structural features related to syntax and other linguistic elements. To harness the full potential of heterogeneous features for CWS, we propose converting linguistic features into heterogeneous graphs and utilizing GCNs for encoding.

3. Methodology

In this section, we introduce the workflow of HGNSeg, including the process of constructing the heterogeneous graph and how to implement word segmentation using GCNs. Specifically, we first use the dataset and multiple features to build a heterogeneous graph for each sample and then initialize all nodes with different encoders. Next, we divide the heterogeneous graph into several sub-graphs according to different edges. The graph convolution operation is performed on each sub-graph, respectively. Then the information from each sub-graph is aggregated, and we will delve into the details of this process in Section 3.2.2. Finally, we concatenate the outputs of GCNs and the outputs of the pretrained model and then feed them into the decoder to get the final label sequence. Fig. 1 shows the overall framework of HGNSeg.

CWS is generally regarded as a character-based sequence labeling task. For a given Chinese sentence $X = \{x_1, x_2, \ldots, x_T\}$, each character in the sequence is marked as a label in the label set, $\mathscr{L} = \{B, M, E, S\}$, where "B" indicates the character is the beginning of a word, "M" means the middle of a word, "E" represents character at the end of a word, and "S" denotes the single-character word. The purpose six CWS is to find the optimal label sequence $Y^* = \{y_1^*, y_2^*, \ldots, y_T^*\}$:

$$Y^* = argmax\ P(Y \mid X)_{Y \in \mathscr{L}^T} \tag{1}$$

3.1 Encoding layer

According to Ma *et al.* (2018), the main factor responsible for the disparity of CWS is insufficient training rather than lousy training. Therefore, we use BERT (Devlin *et al.* 2019) and RoBERTa

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

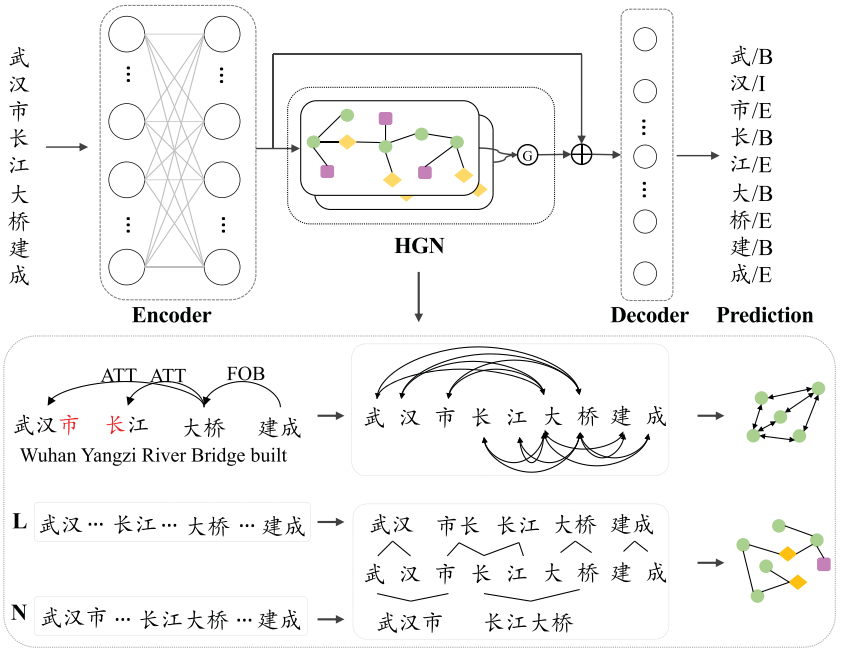

Figure 1. The architecture of HGNSeg. "HGN" at the bottom of the figure is the construction process of the heterogeneous graph. "L" means the lexicon, "N" denotes the n-gram lexicon.

(Liu *et al.* 2019) as the encoders, which are pretrained with a huge amount of unlabeled Chinese data:

$$[\mathbf{e}_1, \ldots, \mathbf{e}_t, \ldots, \mathbf{e}_T] = Encoder([x_1, \ldots, x_t, \ldots, x_T]) \tag{2}$$

where $\mathbf{e}_t$ is the representation for x_t from the encoder.

3.2 Heterogeneous graph convolutional network

3.2.1 Graph construction

Encouraged by some previous studies using GCNs for CWS (Zhao *et al.* 2020; Du *et al.* 2020), we construct the graph by representing characters from the corpus, words from a lexicon $\mathscr{L}$, and n-grams from a n-gram lexicon $\mathscr{N}$ as graph nodes. The method of constructing the lexicon and the n-gram lexicon is introduced in Section 3.4. Then two types of edges are defined according to different relations.

Syntax Edges. Marcheggiani and Titov (2017) introduced GCNs designed to operate on graphs with directional edges and labels, enabling the encoding of linguistic structures like dependency trees. Their approach incorporated edge gates, allowing the model to dynamically adjust the weight of each edge.

Inspired by their methodology, to enhance character representations, when encoding dependency trees, we do not use words as nodes, but the characters that form the words as nodes, and we hope that the information of dependencies can be used to enhance the character representations directly, instead of forming the word representations first, and then enhancing the word representations through the word representations. The word-level information will be augmented by the subsequent character-word/n-grams graph.

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

Therefore, edges connecting characters represent syntax relations between each character pair. Following the principles outlined by Marcheggiani and Titov (2017), we maintain the head-dependent relationship in a similar manner to a dependency tree. Specifically, we preserve the directional information: the outgoing edge signifies the connection from head to dependent, while the incoming edge represents the connection from dependent to head.

We use an example to show our idea for the input sentence in Fig. 1. "大桥(big bridge)" is a head points to the dependent "长江(Yangtze River)", therefore "大(big)" and "桥(bridge)" have two outgoing edges to "长(long)" and "江(river)", respectively, "大(big)" and "桥(bridge)" also have two incoming edges from "长(long)" and "江(river)", respectively. We only convert relations parsed by the toolkit, for example, there is no direct syntactic relation between "武汉市(Wuhan city)" and "建成(Built)", so there is no edge connection between them in the graph.

Character-Word/N-gram Edges. The primary objective of CWS is to determine the optimal segmentation positions for the input sentence. However, each character x_i may assume distinct roles within different words. For instance, x_i might serve as the beginning, middle, or end of a word, or it may represent a single-character word. These diverse positions convey different contextual information. To effectively capture the local context surrounding x_i and identify candidate words, the edges connecting x_i to words/n-grams in the sub-graph signify whether x_i marks the beginning or the end of words/n-grams. The word/n-gram refers to a span within the sentence that can be matched with entries in the lexicon $\mathscr{L}$ or the n-gram lexicon $\mathscr{N}$. In this type of sub-graph, we establish connections between two adjacent characters to preserve the sequential order within the sequence.

As shown in Fig. 1, since "长" serves as the end character of the word "市长" and the beginning character of the word "长江", it is connected to both "市长" and "长江". Both "市长" and "长江" are split from the input sentence and matched in the lexicon $\mathscr{L}$. Additionally, since "长" serves as the beginning character of the 4-gram "长江大桥", it is also connected to "长江大桥" belonging to the n-gram lexicon $\mathscr{N}$. Importantly, these two types of edges are unidirectional. When a span exists in both the lexicon and the n-gram lexicon, only one of them is retained in the graph.

3.2.2 Graph convolutional network

After building the heterogeneous graph, we convert the character nodes, the word nodes, and the n-gram nodes into embeddings. These embeddings are trainable and subject to refinement during the training process. Subsequently, we apply the graph convolutional operation on the two sub-graphs, followed by the aggregation of information to update the higher-order embedding of each node.

Given a graph $\mathscr{G} = (\mathscr{V}, \mathscr{E}, A)$, in which $\mathscr{V}$ and $\mathscr{E}$ denote the sets of nodes and edges in the heterogeneous graph, respectively. The adjacency matrix is represented by A, where if there is an edge between the i_{th} node and the j_{th} node, then $A_{ij} = 1$. Let $H^0 \in \mathbb{R}^{|\mathscr{V}| \times d|}$ be a feature matrix represents the nodes embeddings, with $h_v \in \mathbb{R}^d$ (each row h_v serving as the embedding for node v). We introduce self-connections and a degree matrix D for the adjacency matrix A as follows:

$$A' = A + I \tag{3}$$

$$\tilde{A} = D^{1/2} A' D^{1/2} \tag{4}$$

where $D_{ii} = \sum_j A_{ij}'$. Then the calculation rule for each layer is as follows:

$$H^{(l+1)} = \sigma(\tilde{A} \cdot H^{(l)} \cdot W^{(l)}) \tag{5}$$

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

where $\sigma(\cdot)$ represents an activation operation like rectified linear unit (ReLU) and $\tilde{A}$ denotes the symmetric normalized adjacency matrix. $H^{(l)} \in \mathbb{R}^{|\mathcal{V}\times d|}$ is the nodes hidden states of l^{th} layer and $W^{(l)}$ is a trainable parameter of the l^{th} layer in the graph.

In our heterogeneous graph, there are different types of edges $\mathcal{T} = \{\tau_{dep.}\tau_{lex.}\}$, where $\tau_{dep.}$ denotes the syntax edges and $\tau_{lex.}$ represents the character-word/n-gram edges. We update $H^{(l+1)}$, which is the representations of nodes in the $(l+1)^{th}$ layer, by summing the embeddings of their neighborhood $H_\tau^{(l)}$ with two types of edges: τ.

$$H^{(l+1)} = \sigma\Big(\sum_{\tau\in\mathcal{T}} (\tilde{A}_\tau \cdot H_\tau^{(l)} \cdot W_\tau^{(l)})\Big) \tag{6}$$

where $\sigma(\cdot)$ denotes a nonlinear activation function. $\tilde{A}_\tau$ represents a symmetric normalized adjacency sub-matrix that exclusively contains one type of edge, denoted as τ. $H_\tau^{(l)}$ denotes an embedding matrix of the adjacent nodes for each node connected by edge type τ. $W_\tau^{(l)}$ is a trainable weight matrix. Initially, $H_\tau^{(0)}$ corresponds to the node features obtained from the pretrained model or random initialization.

3.2.3 Edge-wise gating

As our method depends on automatic toolkits for parsing dependency trees, their performance may not be perfect. Consequently, absorbing equal information from all adjacent nodes may not be suitable, as downstream applications relying on potentially incorrect syntax edges can lead to error propagation. Moreover, each character receives information from multiple candidate words, yet not every candidate word represents a reasonable segmentation for the input sentence. Therefore, it becomes necessary to down-weight the corresponding nodes in the graph.

To address these challenges, drawing inspiration from recent endeavors (van den Oord *et al.* 2016; Marcheggiani and Titov, 2017; Dauphin *et al.* 2017), we introduce a scalar gate for each node pair, defined as follows:

$$g_\tau^{(l)} = \theta(H_\tau^{(l)} \cdot W_\tau^{(l),g} + b_\tau^{(l),g}) \tag{7}$$

where θ represents the logistic sigmoid function, and $W_\tau^{(l),g} \in \mathbb{R}^d$ and $b_\tau^{(l),g} \in \mathbb{R}$ are a weight and a bias for the gate, respectively. Therefore, the final heterogeneous graph CNN calculation is formulated as follows:

$$H^{(l+1)} = \sigma\Big(\sum_{\tau\in\mathcal{T}} (g_\tau^{(l)}(\tilde{A}_\tau \cdot H_\tau^{(l)} \cdot W_\tau^{(l)}))\Big) \tag{8}$$

3.3 Decoding layer

Various decoding algorithms can be implemented, including the softmax layer and conditional random fields (CRF) (Lafferty, McCallum, and Pereira, 2001). As reported by Tian *et al.* (2020b), CRF demonstrates superior performance compared to softmax in CWS tasks. Consequently, in our framework, we choose CRF as the decoder.

In the CRF decoding layer, $P(Y \mid X)$ in Eq. (1) could be represented as:

$$P(Y \mid X) = \frac{\emptyset(Y \mid X)}{\sum_{Y'\in\mathcal{L}^T} \emptyset(Y' \mid X)} \tag{9}$$

where $\emptyset(Y \mid X)$ is the potential function. CRF is concerned with the relation between two consecutive labels, y_{i-1}, y_i:

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

$$\emptyset(Y \mid X) = \prod_{i=2}^{T} \exp\left(s(X, i)_y + b_{y_{i-1}y_i}\right) \tag{10}$$

where $b_{y_{i-1}y_i} \in \mathbf{R}$ is the trainable bias term corresponding to the label pair (y_{i-1}, y_i). Finally, the score for each label of the i_{th} character is calculated by the score function $s(X, i) \in \mathbb{R}^{|\mathscr{A}|}$:

$$s(X, i) = W_s^{\top} a_i + b_s \tag{11}$$

where $a_i = e_i \bigoplus h_i$, which is generated by concatenating the outputs of the encoder E and the outputs of the graph network $H^{(l+1)}$. $W_s \in \mathbb{R}^{d_a \times L}$ and $b_s \in \mathbb{R}^{|\mathscr{A}|}$ are trainable parameters.

3.4 Lexicon and N-gram lexicon construction

To construct the character-word/n-gram sub-graph, the first thing we need to do is build the word lexicon $\mathscr{L}$ and the n-gram lexicon $\mathscr{N}$, since the edges between characters and words/n-grams are built on them. In our work, $\mathscr{L}$ consists of words from the training set. For $\mathscr{N}$, we use accessor variety (AV) to extract high-frequency n-grams from each corpus. Access variation (AV) is an unsupervised method proposed by Feng *et al.* (2004) for extracting words from a Chinese text corpus. In their approach, they attribute significance to both the characters directly preceding the string (leading characters) and those immediately following it (trailing characters) as crucial factors in determining string independence. Through experiments conducted on various corpora, they confirmed that the method based on accessor variety and adhesive characters performs efficiently in fulfilling word extraction tasks. Subsequently, this method has been frequently employed for the extraction of n-grams. The kind of accessor for a multi-character s is defined as follows:

$$AV(s) = \min\{L_{av}(s), R_{av}(s)\} \tag{12}$$

where $L_{av}(s)$ is called the left AV, which represents the number of distinct character types that can precede n-gram s (predecessors). And $R_{av}(s)$ denotes the number of distinct character types that can follow n-gram s (successors). When the AV value is not less than a predefined threshold, it means that these strings should appear in enough different contexts to be considered meaningful.

4. Experiment

4.1 Datasets and experimental setup

We conduct experiments on six CWS benchmarks to assess the effectiveness of the proposed model. These datasets include CITYU, PKU, MSR, and AS from SIGHAN 2005 (Emerson, 2005) and CTB and SXU from SIGHAN 2008 (Jin and Chen, 2008). The statistics of six datasets are listed in Table 1. For SIGHAN 2005, we generate a development set for model tuning by randomly selecting 10% data from the training set. AS and CITYU are two special datasets, and their data are in traditional Chinese, so we convert them to simplified Chinese.

We also evaluate the proposed model on six cross-domain corpora, including two Chinese fantasy novel datasets: DL (*DouLuoDaLu*) and ZX (*ZhuXian*) (Qiu and Zhang, 2015), one medicine dataset: DM (dermatology), a patent dataset: PT (Ye *et al.* 2019), and a literature dataset (Lit) and a computer dataset (Com) form SIGHAN 2010. The statistics of six datasets are listed in Table 2. We use the news corpus PKU training set as the source domain training set of the cross-domain experiments, which has different text types from the four target datasets. For PT, DL, and DM, we also use the PKU development set as the development set.

Following the previous paper (Huang *et al.* 2020), we replace all Latin letters, punctuation, and digits with a unique token. In our experiments, we use the development set to select the best model.

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

Table 1. Statistics of 6 benchmark

Datasets		AS	PKU	MSR	CITYU	CTB	SXU
Words	train	5.0M	1.1M	2.2M	1.3M	0.7M	0.5M
	dev	0.5M	0.1M	0.2M	0.2M	0.05M	0.05M
	test	0.1M	0.1M	0.1M	0.04M	0.05M	0.1M
Chars	train	7.5M	1.6M	3.7M	2.1M	11M	0.7M
	dev	0.8M	0.2M	0.4M	0.2M	0.08M	0.08M
	test	0.2M	0.2M	0.2M	0.06M	0.09M	0.2M
OOV rate (%)		2.2	2.1	1.3	3.7	3.8	2.6

Table 2. Statistics of 6 cross-domain datasets

		Cross-domain					
Datasets		ZX	PT	DL	DM	Com	Lit
Words	train	PKU training set					
	dev	20.4K	–	–	–	18.8K	21.0K
	test	34.0K	34.0K	32.0K	17.0K	23.5K	26.1K
Chars	train	PKU training set					
	dev	28.4K	–	–	–	28.7K	27.5K
	test	48.0K	57.0K	47.0K	30.0K	34.7K	32.4K
OOV rate (%)		2.9	4.9	2.3	5.3	7.3	11.5

For the encoders, we follow the default setting of the BERT (Devlin *et al.* 2019) and the RoBERTa (Liu *et al.* 2019). For GCNs, we represent all the character nodes with the PLM, and we randomly initialize word nodes and n-gram nodes. To obtain the dependency trees as mentioned, we use two automatic toolkits, Stanford CoreNLP Toolkit (SCT)[a] (Manning *et al.* 2014) and LTP 4.0 (Che *et al.* 2021)[b]. Stanford CoreNLP is a NLP toolkit developed by Stanford University. Implemented in Java, it offers fundamental language analysis functions, including tokenization, POS tagging, NER, dependency parsing, sentiment analysis, and coreference parsing. Currently, the toolkit supports nine languages, such as English, Chinese, and Italian. LTP 4.0 is an open-source neuro-linguistic technology platform supporting six essential Chinese NLP tasks. These tasks encompass lexical analysis, including CWS, POS tagging, and NER, as well as syntactic analysis involving dependency analysis. Moreover, LTP 4.0 extends to semantic analysis, covering semantic dependency analysis and semantic annotation. For convolutional operation, we use the two-layer GCNs. The key hyperparameter settings are listed in Table 3.

[a] https://stanfordnlp.github.io/CoreNLP/
[b] https://github.com/HIT-SCIR/ltp

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

Table 3. Experiment hyperparameters setting

Hyperparameters	
Word embedding size	768
1_{st} GCN layer size	128
2_{nd} GCN layer size	768
GCN learning rate	2e-5
GCN dropout	0.5
GCN active function	Relu
AV n-gram max length	5
AV threshold	5
Epochs	30

4.2 Results on benchmark datasets

Table 4 presents the experimental results on the six benchmark corpora, where the reported F1 scores and R_{oov} values are averages over three experimental runs. The results are provided using different encoders, and comparisons are made with some previous SOTA works, including both single-criterion models (Zhang and Fu, 2016; Tian *et al.* 2020b) and multi-criteria models (Chen *et al.* 2017; He *et al.* 2018; Qiu *et al.* 2020; Huang *et al.* 2020). The evaluated models and approaches are as follows:

- Three existing CWS tools: LPT4.0, Stanford CoreNLP, and THULAC (Li and Sun, 2009). LTP 4.0 utilized a PLM (ELECTRA) (Clark *et al.* 2020) as an encoder, trained on the People's Daily Corpus dataset. Stanford CoreNLP employed CRF as the CWS model and was trained with the PKU dataset. THULAC is an efficient Chinese lexical analysis tool, with the CWS model trained based on the People's Daily corpus.
- Zhang and Fu (2016) incorporated word embeddings into recurrent neural network to improve CWS performance.
- Chen *et al.* (2017) combined adversarial learning and Bi-LSTM to learn the common knowledge for multi-criteria CWS.
- He *et al.* (2018) used two special tokens at the beginning and end of the input sentence to mark target criteria and then employed LSTM + CRF as the shared encoder for multi-criteria CWS.
- Qiu *et al.* (2020) employed the Transformer as the encoder to extract the criteria-aware contextual information for multi-criteria CWS.
- Huang *et al.* (2020) retained special markers and replaced the LSTM + CRF backbone model used by He *et al.* (2018) with RoBERTa+CRF for multi-criteria CWS.
- Tian *et al.* (2020b) incorporated wordwood information for the neural segmenter using key-value networks.
- Huang *et al.* (2021) incorporated the lexicon into the PLM with GCNs to improve the performance of CWS and strengthen the robustness of cross-domain CWS.
- Liu *et al.* (2021) used the external lexicon to enhance BERT through a lexicon adapter layer.

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

Table 4. Performance comparison between HGNSeg and previous SOTA models on the test sets of six datasets. And experimental results of HGNSeg on six datasets with different encoders. "+HGN" indicates this model uses the heterogeneous graph network. Here, F and R_{oov} represent the F1 value and OOV recall rate, respectively. The maximum F1 and R_{oov} for each dataset are highlighted. "*" denotes the results of our reproduced experiments

Models			AS	PKU	MSR	CITYU	CTB	SXU
1	LTP 4.0	F	83.96	87.96	84.83	84.47	88.15	86.76
		R_{oov}	63.40	61.57	45.46	71.07	74.99	69.25
2	Stanford CoreNLP	F	89.54	89.32	84.00	88.87	96.61	91.85
		R_{oov}	60.04	69.47	45.17	68.68	93.26	76.98
3	THULAC	F	86.30	91.87	85.56	87.03	88.96	90.40
		R_{oov}	67.66	68.72	43.76	72.42	74.27	77.72
4	Zhang and Fu (2016)	F	-	95.70	97.70	-	95.95	-
		R_{oov}	-	-	-	-	-	–
5	Chen *et al.* (2017)	F	96.64	94.32	96.04	95.55	96.18	96.04
		R_{oov}	72.67	72.67	71.60	81.40	82.48	77.10
6	He *et al.* (2018)	F	-	96.06	97.25	-	96.7	96.47
		R_{oov}	-	-	-	-	-	–
7	Huang *et al.* (2020)	F	-	96.85	98.29	-	97.56	97.56
		R_{oov}	-	82.35	81.75	-	88.02	85.73
8	Qiu *et al.* (2020)	F	96.44	96.41	98.05	96.91	96.99	97.61
		R_{oov}	76.39	78.91	78.92	86.91	87.00	85.08
9	Tian *et al.* (2020b)	F	96.62	96.53	98.40	97.93	97.25	-
		R_{oov}	79.64	85.36	84.87	90.15	88.46	-
10	Huang *et al.* (2021)	F	96.86	**97.21**	**98.52**	98.13	-	-
		R_{oov}	79.22	**90.03**	86.13	91.87	**-**	-
11	Liu *et al.* (2021)	F	96.77*	96.78 *	98.45*	98.28*	97.63*	97.58*
		R_{oov}	75.93*	78.56*	86.28*	90.65*	88.76*	85.69*
12	BERT	F	96.86	96.28	97.40	98.14	97.48	97.32
		R_{oov}	79.87	78.44	85.26	91.27	89.07	84.45
13	BERT + HGN	F	96.88	96.54	97.75	98.18	97.77	97.50
		R_{oov}	80.17	80.08	86.16	91.47	**89.67**	85.13
14	RoBERTa	F	96.73	96.68	98.12	98.29	97.54	97.45
		R_{oov}	78.73	79.62	87.87	91.47	88.61	85.47
15	RoBERTa+HGN(HGNSeg)	F	**96.88**	96.81	98.26	**98.36**	**97.80**	**97.64**
		R_{oov}	**80.67**	81.41	**88.39**	**91.99**	**89.67**	**86.10**

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

Firstly, in comparing our model with existing CWS tools, it is evident that although we utilized LTP 4.0 and Stanford CoreNLP as analysis tools for dependent syntax in our experiments, these off-the-shelf tools prove less effective in CWS due to not being trained with corresponding datasets. The large gap between the experimental results of these tools and ours suggests that our improvements do not depend on additional CWS tools.

Subsequently, we validate the effectiveness of HGNseg by comparing results across different encoders with and without the HGN module. As depicted in Table 4, models incorporating the HGN module consistently outperform their baseline counterparts without the HGN module in terms of F1 value and R_{oov} across the six datasets. Additionally, we observe a noteworthy improvement in the average R_{oov} for RoBERTa-CRF with HGN compared to RoBERTa-CRF, with nearly a 1% enhancement across almost all datasets. These findings indicate that the HGN module contributes to enhancing both segmentation performance and R_{oov}.

The improvement in F1 value achieved by pretrained encoders is consistently notable across different models. When compared with models (Zhang and Fu, 2016; Chen *et al.* 2017) utilizing Bi-LSTM as the encoder, our models show an approximate 2% increase in F1 value, and R_{oov} sees an improvement of approximately 10% when RoBERTa or BERT is employed as the encoder. This improvement can be attributed to the external knowledge provided by the pretraining process in the case of CWS.

In comparison to multi-criteria scenarios implemented in models (Chen *et al.* 2017; He *et al.* 2018; Qiu *et al.* 2020; Huang *et al.* 2020), despite these models being trained with more extensive datasets, our model consistently outperforms them in terms of both F1 scores and R_{oov}. In summary, our model effectively mitigates the OOV issue through the integration of heterogeneous linguistic information with HGN.

Observing the results, it is evident that the approaches presented by Huang *et al.* (2021) and Liu *et al.* (2021) surpass our model on PKU and MSR datasets. A potential explanation for this could be that these models leverage external lexicons, such as the Jieba lexicon. This suggests that the model's performance improves when incorporating a more extensive lexicon. As part of future work, we plan to augment the lexicon by integrating additional external lexicon sources.

4.3 Cross-domain performance

To evaluate the cross-domain ability of HGNSeg, we conduct a comparative analysis with previous cross-domain models, including:

- Three existing word segmentation tools: LPT4.0, Stanford CoreNLP, and THULAC (Li and Sun, 2009).
- The work by Liu *et al.* (2014), involved training CRF on both fully and partially annotated data to enhance cross-domain CWS.
- The approach by Zhang *et al.* (2018), which integrated domain-specific dictionaries to address the OOV issue in cross-domain CWS.
- The model proposed by Ye *et al.* (2019), which trained word embeddings on the target domain corpus, leading to performance improvements in cross-domain CWS.
- The strategy presented by Ding *et al.* (2020), which constructed a target domain lexicon, utilized it to segment the target domain data for labeled data acquisition, and subsequently employed adversarial training to minimize the dissimilarity between the source and target domain representations.
- We also use RoBERTa + CRF and the classical sequence annotation task model RoBERTa + Bi-LSTM + CRF as baselines.

The main results are reported in Table 5. We use PKU training set words and n-grams from each cross-domain corpus to construct character-word/n-grams subgraphs.

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

Table 5. The F1 scores and R_{oov} on test data of six cross-domain datasets. The maximum F1 and R_{oov} scores for each domain dataset are highlighted

Models	DM		PT		DL		ZX		Com		Lit	
	F	R_{oov}	F	R_{oov}	F	R_{oov}	F	R_{oov}	F	R_{oov}	F	R_{oov}
LTP 4.0	82.30	68.92	78.79	54.51	85.48	64.46	76.53	63.36	81.93	55.94	69.09	39.80
Stanford CoreNLP	83.25	68.95	90.74	79.64	86.42	67.09	84.15	75.62	90.85	57.22	81.93	56.92
THULAC	84.07	67.68	89.23	73.20	89.50	70.64	82.20	62.52	86.80	52.75	81.40	58.54
Liu *et al.* (2014)	82.80	-	85.00	-	92.50	-	83.90	-	-	-	-	–
Zhang *et al.* (2018)	81.20	-	85.90	-	92.00	-	88.80	-	-	-	-	-
Ye *et al.* (2019)	82.20	-	85.10	-	93.50	-	89.60	-	-	-	-	-
Ding *et al.* (2020)	85.00	-	89.60	-	**94.10**	-	**90.90**	-	-	-	-	-
RoBERTa	89.86	75.91	94.06	83.96	87.87	62.80	86.43	76.00	92.99	66.41	86.83	60.30
RoBERTa + Bi-LSTM	90.08	**78.53**	94.08	84.79	92.06	63.89	86.42	76.22	93.14	69.99	**87.09**	60.22
HGNSeg	**90.31**	77.40	**94.38**	**86.92**	92.16	**64.50**	86.66	**78.71**	**93.24**	**72.03**	86.91	**60.94**

Analyzing the results in Table 5, our model compares favorably to the other baseline models, achieving optimal results on most datasets.

However, the performance of our model is less satisfactory on the two fantasy novel datasets and the literature dataset. The method proposed by Ding *et al.* (2020) demonstrates superior results on these novel datasets, suggesting that the lexicon specific to the target domain is a key factor in effective cross-domain CWS. Another factor contributing to this discrepancy could be the substantial differences between novel and news text. Novels and news articles often have different writing styles, vocabulary, and structures, which could impact the performance of algorithms or models trained on one type of text when applied to the other.

Specifically, compared to the four previous works, our model exhibits an average 5% improvement in F1 values for DM and PT. Additionally, the inclusion of HGN in our model significantly enhances R_{oov} on all six datasets compared to using RoBERTa alone, with improvements of 1.49% (DM), 2.96% (PT), 1.7% (DL), 2.71% (ZX), 5.62%(Com), and 0.64% (Lit) respectively.

As we know, the OOV problem is the main challenge of cross-domain CWS. Segmenters trained in the newswire domain are limited to segmenting domain-specific words from other domains. We also investigate the influence of the domain-specific n-gram lexicon size on R_{oov}. We randomly selected 20%, 40%, 60%, and 80% of the n-grams from each target corpus to construct new n-gram lexicons with varying proportions. As illustrated in Fig. 2, the R_{oov} values of HGNSeg consistently increase with lexicon expansion. Therefore, we can reasonably infer that HGNSeg is likely to achieve better performance when utilizing a lexicon containing a more extensive set of domain-specific n-grams.

In Fig. 2, the most substantial improvement in R_{oov} is observed for PT. Upon analyzing cases in the PT test set, we found that our model can recognize some domain-specific entity words. As illustrated in Table 6, for the first sentence, the model segments "贲亭酸甲酯(Methyl benzoate)" as two words when not using n-grams from PT. In contrast, the model correctly segments this word when utilizing the n-gram lexicon from PT. Similarly, for the second sentence, "白芥子(Baijiezi)" represents a Chinese herbal medicine, constituting a domain-specific word in PT. The model, equipped with the n-gram lexicon, correctly recognizes this word. Several

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

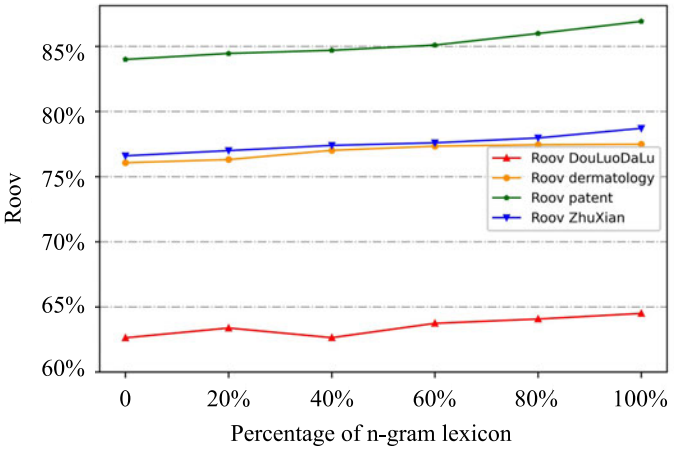

Figure 2. R_{oov} with different sizes of n-gram lexicon for four cross-domain datasets. We randomly pick different percentages of n-grams from each target corpora to build new n-gram lexicons.

other examples, such as "蛋白激酶(Protein kinase)", "聚丙烯蜡(Polypropylene wax)", and "酶制剂(Enzyme formulation)", further support the effectiveness of domain-specific n-grams in improving R_{oov} in the cross-domain scenario.

4.4 Ablation study

This section discusses the validity of each element in the heterogeneous GCNs through ablation experiments, and the results are presented in Table 7.

The first ablation experiment confirms the effectiveness of the character-word/n-gram sub-graph. The comparison between the first and the third lines in Table 7 shows an average improvement of about 1% in R_{oov} values on the six datasets, with PKU and MSR being more sensitive to this type of sub-graph, exhibiting improvements of 3% and 1.62%, respectively.

The second ablation study evaluates the influence of the dependency tree sub-graph. In this experiment, the performance of HGNSeg is tested by removing the dependency tree sub-graph from the heterogeneous graph. As observed in the second line of Table 7, although this type of sub-graph also contributes to improvements in overall F1 and R_{oov} values, it appears that the character-word/n-gram sub-graph has a more significant impact on the majority of datasets.

4.5 Effect of different lexicons

In this section, we also analyze the impact of different lexicons in the character-word/n-gram sub-graph, specifically, the use of lexicons $\mathscr{L}$ and $\mathscr{N}$. The experimental results with different lexicons are presented in Table 8, indicating varying degrees of impact on different datasets. Notably, for the PKU dataset, $\mathscr{L}$ appears to be more important, whereas for the AS dataset, $\mathscr{N}$ is deemed more crucial. In general, both the lexicon from the training set and the n-gram lexicon have a significant positive impact, considerably boosting the performance.

4.6 Effect of different syntax parsing toolkits

Through the preceding analysis, our model effectively incorporates syntax information. However, existing syntax parsing toolkits are not flawless, and parsing mistakes are noticeable, particularly

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

Table 6. The segmentation results with different proportions of n-grams from PT corpus. “0%” means this model without n-grams from the PT training set, and “100%” means this model uses all the n-grams from the PT training set

No.		Cases from PT
	Input1:	提高了贲亭酸甲酯的收率，产品纯度高。
		(Improve the yield of methyl benzoate, high product purity.)
	Gold answer:	提高/了/贲亭酸甲酯/的/收率/，/产品/纯度/高/。
		(Improve/the/methyl benzoate/of/yield/,/product/purity/high.)
	0%:	提高/了/贲亭酸/甲酯/的/收率/，/产品/纯度/高/。
		(Improve/the/benzoic acid/methyl ester/of/yield/,/product/purity/high.)
	100%:	提高/了/贲亭酸甲酯/的/收率/，/产品/纯度/高/。
		(Improve/the/methyl benzoate/of/yield/,/product/purity/high.)
2	Input2:	胆星80克；白芥子100克。
		(Danxing 80g; Baijiezi 100g)
	Gold answer:	胆星/80/克/；/白芥子/100/克/。
		(Danxing 80g; Baijiezi 100g)
	0%:	胆星/80/克/；/白/芥子/100/克/。
		(Danxing 80g; White Jiezi 100g)
	100%:	胆星/80/克/；/白芥子/100/克/。
		(Danxing 80g; Baijiezi 100g)
3	Input3:	采用单一的酶制剂，各自都具有各自的适应温度。
		A single enzyme preparation is used, each with its own adaptation temperature.
	Gold answer:	采用/单一/的/酶制剂/，各自/都/具有/各自/的/适应/温度/。
		(Use/single/of/enzyme formulation/,/each/has/own/of/adaptation/temperature.)
	0%:	采用/单一/的/酶/制剂/，/各自/都/具有/各自/的/适应/温度/。
		(Use/single/of/enzyme/formulation/,/each/has/own/of/adaptation/temperature.)
	100%:	采用/单一/的/酶制剂/，/各自/都/具有/各自/的/适应/温度/。
		(Use/single/of/enzyme formulation/,/each/has/own/of/adaptation/temperature.)

when applied to long sentences. To compare the effects of two dependency syntax parsing toolkits, we present histograms of the F1 and R_{oov} values obtained from HGNSeg with different parsing toolkits on six datasets (yellow bars for SCT, green bars for LTP 4.0) in Fig. 3.

As illustrated in Fig. 3a and b, LTP 4.0 proves to be more suitable for HGNSeg, and the models using LTP 4.0 exhibit superior performance on all six datasets compared to those using SCT. Despite SCT providing rich dependency syntax labeling information, our model only utilizes the

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

Table 7. Ablation experiments, "w/o $\mathscr{L}/\mathscr{N}$" means without the character-word/n-gram sub-graph, "w/o Dep." means without the dependency tree sub-graph

Models		AS	PKU	MSR	CITYU	CTB6	SXU
w/o $\mathscr{L}/\mathscr{N}$	F	96.76	96.64	98.09	98.08	**97.82**	97.40
	R_{oov}	79.44	78.41	86.77	91.00	89.52	84.58
w/o Dep.	F	96.79	96.78	98.05	97.98	97.65	97.45
	R_{oov}	78.89	80.32	87.49	91.07	88.8	85.15
HGNseg	F	**96.88**	**96.81**	**98.26**	**98.36**	97.80	**97.64**
	R_{oov}	**80.67**	**81.41**	**88.39**	**91.99**	**89.67**	**86.10**

Table 8. Comparison of different types of the lexicon in the character-word/n-gram sub-graph. "w/o $\mathscr{N}$" means the model without the n-gram lexicon, "w/o $\mathscr{L}$" means the model without the lexicon from the training set. The last line represents the model using $\mathscr{N}$ and $\mathscr{L}$ together

Models		AS	PKU	MSR	CITYU	CTB6	SXU
w/o $\mathscr{N}$	F	96.84	96.68	98.22	98.27	97.65	97.36
	R_{oov}	78.80	79.59	87.92	90.72	89.00	85.02
w/o $\mathscr{L}$	F	**96.89**	95.03	96.65	98.14	97.67	97.21
	R_{oov}	80.28	79.26	86.59	90.40	89.00	85.54
HGNSeg	F	96.88	**96.81**	**98.26**	**98.36**	**97.80**	**97.64**
	R_{oov}	**80.67**	**81.41**	**88.39**	**91.99**	**89.67**	**86.10**

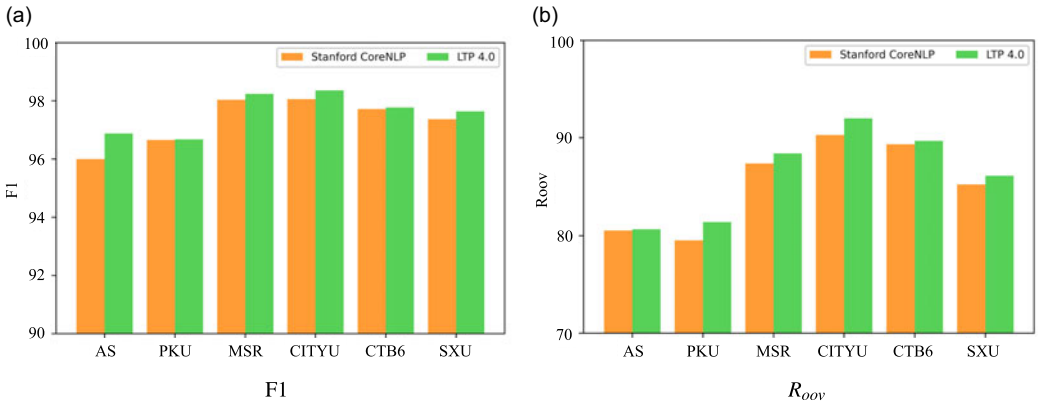

Figure 3. The F1 values and R_{oov} of HGNSeg using two different syntax parsing toolkits, namely SCT and LTP 4.0, on six datasets.

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

Table 9. The segmentation results of five examples from the AS test set with different syntax parsing toolkits

NO.	Toolkits	Cases form AS
1	LTP 4.0	项目/包括/列车/门/打开/、/站台/停车/越位/等， item/include/train/door/open/,/platform/parking/offside/etc,
	Stanford CoreNLP	项目/包括/列车门/打开/、/站台/停车/越位/等/， item/include/train door/open/,/platform/parking/offside/etc,
2	LTP 4.0	WHO/限量/为/每/天/每/公斤/体重/0.5/mg/， WHO/limitation/is/every/day/every/kg/body weight/0.5/mg/,
	Stanford CoreNLP	WHO/限量为/每/天/每/公斤/体重/0.5/mg/， WHO/limitation is/every/day/every/kg/body weight/0.5/mg/,
3	LTP 4.0	再/加上/英特尔/本/月/中/将/再度/推出/的/820/芯片组 again/plus/intel/this/month/mid/will/again/launch/the/820/chipset
	Stanford CoreNLP	再/加上/英特尔/本/月/中将/再度/推出/的/820/芯片组 again/plus/intel/this/month/vice admiral/again/launch/the/820/chipset
4	LTP 4.0	尼可拉斯凯吉/有点/动心 Nicholas Cage/a little/tempted
	Stanford CoreNLP	尼可拉斯/凯吉/有点/动心 Nicholas/Cage/a little/tempted
5	LTP 4.0	人们/在/比较/中国/与/西方/时，/常常/是/拿/当代/的/中国/与/当代/的/西方/相比/。 People /in/compare/china/and/Western/time/, /often/are/take/contemporary/of/China/and/contemporary/of/Western/compare.
	Stanford CoreNLP	人/们/在/比较/中国/与/西方/时/，/常常/是/拿/当代/的/中国/与/当代/的/西方/相比/。 People/(plural marker) /in/compare/china/and/Western/time/, /often/are/take/contemporary/of/China/and/contemporary/of/Western/compare.

head and dependent information. Consequently, the dependency syntax parsing results from LTP 4.0 suffice to meet the needs of the CWS task in our framework. In future work, we will explore the integration of additional dependency labeling features from SCT to further enhance CWS.

As depicted in Fig. 3a, there is a substantial disparity in the F1 values on the AS dataset when using LTP 4.0 and SCT. To further investigate, we analyze specific examples from the AS test set, as presented in Table 9.

For the first sentence, "项目包括列车门打开，站台停车越位等 (Items include train doors opening, platform parking offside, etc.)", although "列车门(train door)" could be considered a valid segmentation, segmenting this sequence into two words, "列车(train)" and "门(door)", aligns

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

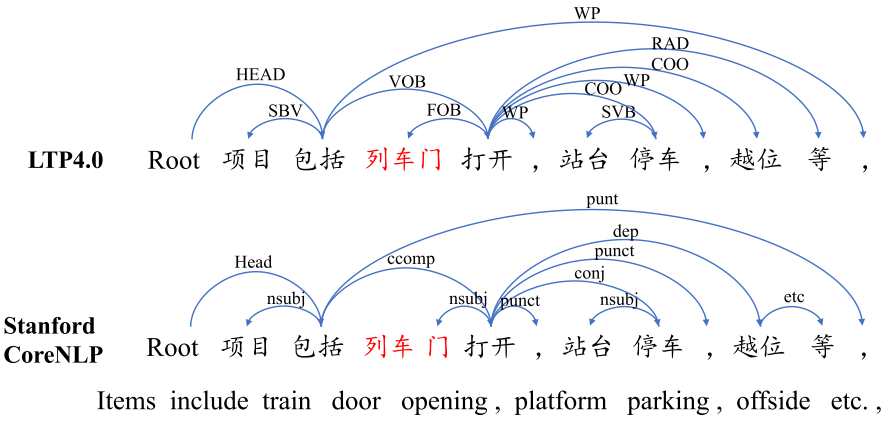

Figure 4. The dependency tree parsing results of " 项目包括列车门打开，站台停车，越位等," from two syntax parsing toolkits.

more closely with the word segmentation criteria of the AS dataset. The dependency tree parsing results in Fig. 4 highlight the evident difference between LTP 4.0 and SCT, leading to distinct segmentation outcomes. This observation reinforces the effectiveness of our model in incorporating dependency tree parsing results.

In the fourth example, a person's name is treated as a single word in AS, whereas the model with SCT segments it into two words.

The model utilizing SCT often produces noticeable errors. For instance, in the third sentence in Fig. 4, it segments "本月中将" as "本(this)/月(month)/中将(vice admiral)", while the correct gold segmentation is "本(this)/月(month)/中(mid)/将(will)". Further analysis reveals that some of these errors stem from incorrect dependency parsing results, as observed in the second and fifth sentences. Hence, the accuracy of the syntax parsing toolkit significantly impacts our model's performance.

In Fig. 3b, the R_{oov} values across all datasets using LTP 4.0 are higher compared to those using SCT. Notably, there is a substantial gap in the R_{oov} values between the two syntax parsing toolkits, particularly evident in the CITYU dataset. Examining Table 10, it becomes apparent that the model with LTP 4.0 can recognize certain OOV words, such as "龙井(Longjin)", "提引水(raised water)", and "露宿者(rough sleeper)".

In summary, across the six benchmark datasets, models employing LTP 4.0 exhibit superior performance in terms of both F1 and R_{oov} values compared to those utilizing SCT.

4.7 Statistical significance tests

In this subsection, we aim to assess the statistical significance of the improvements presented in this paper by conducting F-score tests on different models.

Following Wang *et al.* (2010), we use the bootstrapping method proposed by Zhang *et al.* (2004), which is operated as follows.

Given a test set T_0 with N test examples, we perform repeatedly sampling from T_0, create a new test set T_1 with N examples, and repeat the process $M-1$ times. Finally, we obtain a total of M test sets $\{T_0, T_1, \ldots, T_M\}$. In our test procedure, M is set to 1000.

Subsequently, system A scored a_0 on T_0 and system B scored b_0, the discrepancy between system A and system B is denoted as $\delta_0 = a_0 - b_0$. Repeat this testing process on each test set resulting in M discrepancy scores $\{\delta_0, \delta_1, \ldots, \delta_M\}$. Following Zhang *et al.* (2004), we measure the 95% confidence interval for the discrepancy (i.e., the 2.5th percentile and the 97.5th percentile) between

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

Table 10. Effect of different syntax parsing toolkits

NO.	Toolkits	Cases from CITYU
1	LTP 4.0	但/居于/附近/的/街坊/批评/有关/露宿者/不/注重/卫生/, /促请/政府/助/他们/搬走/。
		But/live/near/of/neighbors/criticize/about/rough sleepers/not attention/hygiene/, /urge/government/help/them/move out/.
	Stanford CoreNLP	但/居于/附近/的/街坊/批评/有关/露宿/者/不/注重/卫生/, /促请/政府/助/他们/搬走/。
		But/live/near/of/neighbors/criticize/about/rough/sleepers/not attention/hygiene/, /urge/government/help/them/move out/.
2	LTP 4.0	贵州/茶叶/冒充/龙井/叶/小/形/扁/浑水摸鱼/大批/流向/全/国
		Guizhou/tea/impersonation/Longjing/leaf/small/shape/flat/fishing in troubled water/large number/ flow direction/whole/country
	Stanford CoreNLP	贵州/茶叶/冒充/龙井叶/小/形/扁/浑水摸鱼/大批/流向/全/国
		Guizhou/tea/impersonation/Longjing leaf/small/shape/flat/fishing in troubled water/large number/ flow direction/whole/country
3	LTP 4.0	闻喜县/丈八村/刚刚/建/好/的/提引水/工程/,
		Wenxi County/Zhangba Village/just now/build/good/of /raised water/project/,
	Stanford CoreNLP	闻喜县/丈八村/刚刚/建/好/的/提/引水/工程/,
		Wenxi County/Zhangba Village/just now/build/good/of /lift/raised water/project/,

Table 11. Statistical significance test of F-score for our system and Liu *et al.* (2021) system

Systems			
A	B	CITYU	MSR
HGNSeg (Ours)	Liu *et al.* (2021)	>	>
		+0.36	+0.22

the two models. If the confidence interval does not overlap with zero, it can be asserted that the differences between systems A and B are statistically significant (Zhang *et al.* 2004).

Table 11 is the example of significant differences between our system and system from Liu *et al.* (2021), where ">" means our system is significantly "better" than the system from Liu *et al.* (2021), demonstrating a steady improvement on both datasets.

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

5. Conclusion

In this paper, we introduce a novel framework for CWS named HGNSeg, focusing on the integration of heterogeneous features through GCNs. The diverse set of features, including words, n-grams, and dependency trees, along with their structural information, are effectively encoded using GCNs. Experimental results conducted on six benchmark datasets illustrate the efficacy and robustness of HGNSeg. Ablation experiments emphasize the model's capability to seamlessly integrate syntax and lexicon features. The cross-domain experiments reveal that HGNSeg contributes to mitigating the OOV challenge. Additionally, a detailed analysis of various lexicons and syntax parsing toolkits suggests that a larger domain-specific n-gram lexicon and a superior parsing toolkit can significantly enhance the performance of HGNSeg.

In our proposed framework, HGNSeg did not leverage the label information of the dependency tree, which has the potential to provide additional valuable dependency information. Our future work will explore incorporating label information into the CWS framework to further enhance performance. It is worth noting that the HGNSeg framework is not limited to CWS; its applicability extends to other Chinese sequence labeling tasks, including NER and POS tagging.

Acknowledgments. This research is supported by the NSFC project "The Construction of the Knowledge Graph for the History of Chinese Confucianism" (Grant No. 72010107003).

Competing interests. All authors disclosed no relevant relationships.

Author contributions. Xuemei Tang: Conceptualization of this study, Methodology, Experiments, Writing, Original draft preparation.

Qi Su: Investigation, Data Curation, Supervision, Writing – Review and Editing.

Jun Wang: Methodology, Supervision, Project administration, Funding acquisition, Writing – Review and Editing.

Ethics statement. The datasets used in this paper are open datasets and do not involve any ethical issues.

References

Bastings J., Titov I., Aziz W., Marcheggiani D. and Simaan K. (2017). *Graph convolutional encoders for syntax-aware neural machine translation*. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark*. Association for Computational Linguistics,

Che W., Feng Y., Qin L. and Liu T. (2021). *N-LTP: An open-source neural language technology platform for Chinese*. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online and Punta Cana, Dominican Republic*. Association for Computational Linguistics, pp. 42–49.

Chen X., Qiu X., Zhu C., Liu P. and Huang X. (2015). *Long short-term memory neural networks for Chinese word segmentation*. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal*. Association for Computational Linguistics, pp. 1197–1206.

Chen X., Shi Z., Qiu X. and Huang X. (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), Vancouver, Canada*. Association for Computational Linguistics, pp. 1193–1203.

Clark K., Luong M.-T., Le Q.V. and Manning C.D. (2020). Electra: pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv: 2003.10555.

Dauphin Y.N., Fan A., Auli M. and Grangier D. (2017). Language modeling with gated convolutional networks. arXiv preprint arXiv:1612.08083.

Defferrard M., Bresson X. and Vandergheynst P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing System, NIPS'16, Barcelona, Spain*. pp. 2844–2852.

Devlin J., Chang M., Lee K. and Toutanova K. (2019). *BERT: pre-training of deep bidirectional transformers for language understanding*. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT. 2019*, June 2-7, 2019, *Minneapolis, MN, USA*. Association for Computational Linguistics, vol. **1**, pp. 4171–4186, Long and Short Papers.

Diao S., Bai J., Song Y., Zhang T. and Wang Y. (2019). Zen: pre-training chinese text encoder enhanced by n-gram representations. arXiv: 1911.00720 [cs]. arXiv: 1911.

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

Ding N., **Long D.**, **Xu G.**, **Zhu M.**, **Xie P.**, **Wang X. and Zheng H.** (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*, Online. Association for Computational Linguistics, pp. 6662–6671.

Du J., **Mi W. and Du X.** (2020). *Chinese word segmentation in electronic medical record text via graph neural network-bidirectional lstm-crf model*. In *In. 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, *Seoul, Korea (South)*. IEEE, pp. 985–989.

Emerson T. (2005). *The second international Chinese word segmentation bakeoff*. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.

Feng H., **Chen K.**, **Deng X. and Zheng W.** (2004). Accessor variety criteria for chinese word extraction. *Computational Linguistics* **30**(1), 75–93.

He H., **Wu L.**, **Yan H.**, **Gao Z.**, **Feng Y. and Townsend G.** (2018). *Effective neural solution for multi-criteria word segmentation*. In *Smart Intelligent Computing and Applications: Proceedings of the Second International Conference on SCI 2018*, vol. 2. Springer, Singapore, pp. 133–142.

Hu L., **Yang T.**, **Shi C.**, **Ji H. and Li X.** (2019). *Heterogeneous graph attention networks for semi-supervised short text classification*. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, *Hong Kong, China*. Association for Computational Linguistics, pp. 4821–4830.

Hu L., **Yang T.**, **Zhang L.**, **Zhong W.**, **Tang D.**, **Shi C.**, **Duan N. and Zhou M.** (2021). *Compare to the knowledge: graph neural fake news detection with external knowledge*. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online. Association for Computational Linguistics, pp. 754–763.

Huang C.-R. (1995). *The morpho-lexical meaning of mutual information: a corpus-based approach towards a defi- nition of mandarin words*. In *1995 Linguistic Society of America Annual Meeting*, New Orleans.

Huang C.-R. and Xue N. (2012). Words without boundaries: computational approaches to chinese word segmentation. *Language and Linguistics Compass* **6**(8), 494–505.

Huang C.-R., **Šimon P.**, **Hsieh S.-K. and Prévot L.** (2007). *Rethinking Chinese word segmentation: Tokenization, character classification, or wordbreak identification*. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, *Prague, Czech Republic*. Association for Computational Linguistics, pp. 69–72.

Huang K., **Huang D.**, **Liu Z. and Mo F.** (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)*, Online. Association for Computational Linguistics, pp. 3873–3882.

Huang K., **Yu H.**, **Liu J.**, **Liu W.**, **Cao J. and Huang D.** (2021). Lexicon-based graph convolutional network for Chinese word segmentation. In *In, Findings of the Association for Computational Linguistics: EMNLP, Punta Cana, Dominican Republic*. Association for Computational Linguistics, pp. 2908–2917.

Jin G. and Chen X. (2008). *The fourth international chinese language processing bakeoff: Chinese word segmentation, named entity recognition and chinese pos tagging*. In *Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing*.

Kipf T.N. and Welling M. (2017). Semi-supervised classification with graph convolutional networks. arXiv: 1609. 0. 2907 [cs, stat]. arXiv: 1609. 0. 2907.

Kurita S., **Kawahara D. and Kurohashi S.** (2017). Neural joint model for transition-based chinese syntactic analysis. In Long Papers.*Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, July 30 - August 4, *Vancouver, Canada*, Association for Computational Linguistics, vol. **1**, pp. 1204–1214,

Lafferty J.D., **McCallum A. and Pereira F.C.N.** (2001). *Conditional random fields: probabilistic models for segmenting and labeling sequence data*. In *Proceedings of the Eighteenth International Conference on Machine Learning, ICML'01*, *San Francisco, CA, USA*. Morgan Kaufmann Publishers Inc, pp. 282–289.

Li S., **Zhou G. and Huang C.-R.** (2012). *Active learning for chinese word segmentation*. In *Proceedings of COLING 2012: Posters*, *Mumbai, India*. The COLING 2012 Organizing Committee, pp. 683–692.

Li Z. and Sun M. (2009). Punctuation as implicit annotations for Chinese word segmentation. *Computational Linguistics* **35**(4), 505–512.

Liu J., **Wu F.**, **Wu C.**, **Huang Y. and Xie X.** (2018). Neural Chinese word segmentation with dictionary knowledge. In **Zhang M.**, **Ng V.**, **Zhao D.**, **Li S. and Zan H.** (eds), *Natural Language Processing and Chinese Computing*. Cham: Springer International Publishing, pp. 80–91.

Liu W., **Fu X.**, **Zhang Y. and Xiao W.** (2021). Lexicon enhanced chinese sequence labeling using bert adapter. arXiv: 2105.07148 [cs]. arXiv: 2105.07148.

Liu Y., **Ott M.**, **Goyal N.**, **Du J.**, **Joshi M.**, **Chen D.**, **Levy O.**, **Lewis M.**, **Zettlemoyer L. and Stoyanov V.** (2019). Roberta: a robustly optimized bert pretraining approach. arXiv:1907.11692 [cs]. arXiv: 1907.11692.

Liu Y., **Zhang Y.**, **Che W.**, **Liu T. and Wu F.** (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)*, pp. 864–874.

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

Ma J., **Ganchev K. and Weiss D.** (2018). State-of-the-art Chinese word segmentation with Bi-LSTMS. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, *Brussels, Belgium*. Association for Computational Linguistics, pp. 4902–4908.

Manning C.D., **Surdeanu M.**, **Bauer J.**, **Finkel J.R.**, **Bethard S. and McClosky D.** (2014). *The stanford corenlp natural language processing toolkit*. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60.

Marcheggiani D. and Titov I. (2017). Encoding sentences with graph convolutional networks for semantic role labeling. arXiv: 1703.04826 [cs]. arXiv: 1703.04826.

Margatina K., **Baziotis C. and Potamianos A.** (2019). *Attention-based conditioning methods for external knowledge integration*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, *Florence, Italy*. Association for Computational Linguistics, pp. 3944–3951.

Michael D., **Xavier B. and Pierre V.** (2016). Convolutional neural networks on graphs with fast localized spectral filtering. arXiv: 1606.09375[cs]. arXiv: 1606.09375.

Nie Y., **Zhang Y.**, **Peng Y. and Yang L.** (2022). Borrowing wisdom from world: modeling rich external knowledge for chinese named entity recognition. *Neural Computing and Applications* **34**(6), 4905–4922.

Qiu L. and Zhang Y. (2015). *Word segmentation for Chinese novels*. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 2440–2446.

Qiu X., **Pei H.**, **Yan H. and Huang X.** (2020). A concise model for multi-criteria Chinese word segmentation with transformer encoder. In *In, Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, pp. 2887–2897.

Shao Y., **Hardmeier C.**, **Tiedemann J. and Nivre J.** (2017). Character-based joint segmentation and pos tagging for Chinese using bidirectional RNN-CRF. arXiv preprint arXiv:1704.01314.

Shen Y., **Tan S.**, **Sordoni A.**, **Li P.**, **Zhou J. and Courville A.** (2022). *Unsupervised dependency graph network*. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, *Dublin, Ireland*. Association for Computational Linguistics, pp. 4767–4784.

Sproat R. and Shih C. (1990). A statistical method for finding word boundaries in Chinese text. *Computer Processing of Chinese and Oriental Languages* **4**(4), 336–351.

Sui D., **Chen Y.**, **Liu K.**, **Zhao J. and Liu S.** (2019). *Leverage lexical knowledge for Chinese named entity recognition via collaborative graph network*. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, *Hong Kong, China*. Association for Computational Linguistics, pp. 3830–3840.

Tian Y., **Song Y.**, **Ao X.**, **Xia F.**, **Quan X.**, **Zhang T. and Wang Y.** (2020a). *Joint Chinese word segmentation and part-of-speech tagging via two-way attentions of auto-analyzed knowledge*. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics, pp. 8286–8296.

Tian Y., **Song Y. and Xia F.** (2020a). *Joint Chinese word segmentation and part-of-speech tagging via multi-channel attention of character n-grams*. In *Proceedings of the 28th International Conference on Computational Linguistics*, *Barcelona, Spain (Online)*. International Committee on Computational Linguistics, pp. 2073–2084.

Tian Y., **Song Y.**, **Xia F.**, **Zhang T. and Wang Y.** (2020b). *Improving Chinese word segmentation with wordhood memory networks*. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics, pp. 8274–8285.

Van den Oord A., **Kalchbrenner N.**, **Vinyals O.**, **Espeholt L.**, **Graves A. and Kavukcuoglu K.** (2016). Conditional image generation with pixelcnn decoders. In *Proceedings of the 30th International Conference on Neural Information Processing System, NIPS'16, Barcelona, Spain*. pp. 4797–4805.

Wang K., **Zong C. and Su K.-Y.** (2010). *A character-based joint model for Chinese word segmentation*. In *Proceedings of the 23rd International Conference on Computational Linguistics*, *Beijing, China*. Coling 2010 Organizing Committee, pp. 1173–1181.

Xue N. (2003). Chinese word segmentation as character tagging. In *ROCLING/IJCLCLP*

Yao L., **Mao C. and Luo Y.** (2018). Graph convolutional networks for text classification. arXiv: 1809.05679 [cs]. arXiv: 1809.05679.

Ye Y., **Zhang Y.**, **Li W.**, **Qiu L. and Sun J.** (2019). Improving cross-domain chinese word segmentation with word embeddings. arXiv preprint arXiv: 1903.01698.

Yu H., **Huang K.**, **Wang Y. and Huang D.** (2022). Lexicon-augmented cross-domain Chinese word segmentation with graph convolutional network. *Chinese Journal of Electronics* **31**(5), 949–957.

Zhang M. and Fu G. (2016). *Transition-based neural word segmentation*. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, *Berlin, Germany*. Association for Computational Linguistics, pp. 421–431.

Zhang M., **Zhang Y. and Fu G.** (2016). *Transition-based neural word segmentation*. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, *Berlin, Germany*. Association for Computational Linguistics, pp. 421–431.

https://doi.org/10.1017/nlp.2024.49 Published online by Cambridge University Press

Zhang Q., **Liu X. and Fu J.** (2018). *Neural networks incorporating dictionaries for Chinese word segmentation*. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, February 2-7, 2018, *New Orleans, Louisiana, USA*. AAAI Press, pp. 5682–5689.

Zhang Y., **Vogel S. and Waibel A.** (2004). *Interpreting bleu/nist scores: how much improvement do we need to have a better system?* In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, *Lisbon, Portugal*. European Language Resources Association (ELRA).

Zhao L., **Zhang A.**, **Liu Y. and Fei H.** (2020). Encoding multi-granularity structural information for joint Chinese word segmentation and pos tagging. *Pattern Recognition Letters* **138**, 163–169.

Cite this article: Tang X, Su Q and Wang J. Chinese word segmentation with heterogeneous graph convolutional network. *Natural Language Processing* https://doi.org/10.1017/nlp.2024.49