Enhancing Chinese Pre-trained Language Model via Heterogeneous Linguistics Graph

Yanzeng Li[1*], **Jiangxia Cao**[2,3*], **Xin Cong**[2,3], **Zhenyu Zhang**[2,3]
Bowen Yu[2,3], **Hongsong Zhu**[2,3†], **Tingwen Liu**[2,3†]
[1]Wangxuan Institute of Computer Technology, Peking University. Beijing, China
[2]Institute of Information Engineering, Chinese Academy of Sciences. Beijing, China
[3]School of Cyber Security, University of Chinese Academy of Sciences. Beijing, China
liyanzeng@stu.pku.edu.cn
{caojiangxia, congxin, zhangzhenyu1996}@iie.ac.cn
{yubowen, zhuhongsong, liutingwen}@iie.ac.cn

Abstract

Chinese pre-trained language models usually exploit contextual character information to learn representations, while ignoring the linguistics knowledge, e.g., word and sentence information. Hence, we propose a task-free enhancement module termed as **H**eterogeneous **L**inguistics **G**raph (**HLG**) to enhance Chinese pre-trained language models by integrating linguistics knowledge. Specifically, we construct a hierarchical heterogeneous graph to model the characteristics linguistics structure of Chinese language, and conduct a graph-based method to summarize and concretize information on different granularities of Chinese linguistics hierarchies. Experimental results demonstrate our model has the ability to improve the performance of vanilla BERT, BERTwwm and ERNIE 1.0 on 6 natural language processing tasks with 10 benchmark datasets. Further, the detailed experimental analyses have proven that this kind of modelization achieves more improvements compared with previous strong baseline MWA. Meanwhile, our model introduces far fewer parameters (about half of MWA) and the training/inference speed is about 7x faster than MWA. Our code and processed datasets are available at https://github.com/lsvih/HLG.

1 Introduction

Pre-trained Language Models (PLM) (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2018; Yang et al., 2019) have recently demonstrated the effectiveness on a variety of natural language processing (NLP) tasks, such as machine translation and text summarization. For a specific downstream task, the parameters of PLMs can be fine-tuned with accurately labeled instances or weakly labeled instances of the task to achieve better performance.

In recent, there are a series of studies on adapting PLMs for Chinese (Meng et al., 2019; Sun et al., 2019; Cui et al., 2019a; Sun et al., 2020; Wei et al., 2019; Diao et al., 2020; Lai et al., 2021). Many researchers introduce the Chinese-specific linguistics knowledge such as word information into PLMs by conducting elaborate self-supervised tasks to pre-train Chinese PLMs from scratch. Nevertheless, pre-training a PLM is computationally expensive and time-consuming since it needs large-scale Chinese corpus and heavy computational resources. The high cost makes it difficult for researchers to pre-train a PLM from scratch.

An alternative way is to integrate the Chinese-specific linguistics knowledge into pre-trained PLMs in the fine-tuning stage in downstream tasks directly. Following this idea, the task-free enhancement module is widely used in the fine-tuning stage by adding an additional adapter in PLMs to integrate external knowledge (Li et al., 2020). As shown in Figure 1, the enhancement module is inserted between PLMs and task-specific module, and its inputs are the hidden representations of PLMs and embeddings of external knowledge. To achieve the goal of integrating external knowledge into PLMs in the fine-tuning stage, the enhancement module should have the following characteristics. First, as a plug-in adapter module in fine-tuning stage, it should maintain consistent output formulation with PLM. Second, it should not introduce unacceptable time or space complexity for training and inference. Third, it should improve the performance of downstream tasks universally.

With the core idea of the enhancement module, Li et al. (2020) proposed a multi-source word-aligned model (MWA) to enhance PLMs by integrating Chinese Word Segmentation (CWS) bound-

[*] Both authors contributed equally to this work
[†] Corresponding Author

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 1986 - 1996
May 22-27, 2022 ©2022 Association for Computational Linguistics

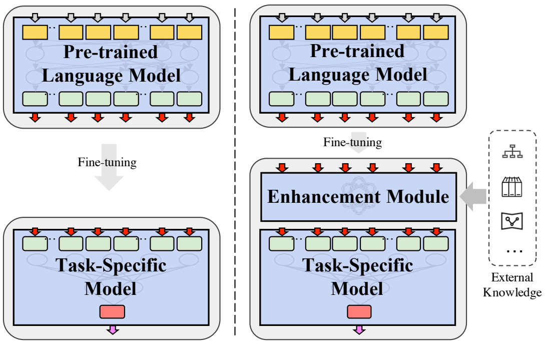


Figure 1: The diagram of Enhancement Module framework. Left: Fine-tuning PLM ordinarily. Right: Enhancement for Fine-tuning PLM.

aries information implicitly. It first exploits various CWS tools to generate multiple word sequences and then utilizes word-aligned attention with a mixed pooling to integrate the word information into characters. Experimental results show that MWA has the ability to utilize CWS segmentation information to enhance Chinese PLMs to achieve SOTA performance in many downstream NLP tasks. However, MWA has two weaknesses: 1) **Efficiency Degradation**: The model structure of MWA is naturally non-parallel and cannot benefit from GPU acceleration (detailed in §4.3.3), which results in time inefficiencies in both training and inference processes. 2) **Linguistic Information Loss**: MWA utilizes a pooling-based mechanism to perform interaction between characters and words. Such a heuristic method could not make full use of information, resulting in sub-optimal results.

To tackle the aforementioned limitations, we propose **H**eterogeneous **L**inguistics **G**raph (**HLG**), which is Graph Neural Network (GNN) based method to integrate CWS information to enhance PLMs. Specifically, the hierarchical CWS information is first conducted by a heterogeneous graph, which contains character nodes, word nodes and sentence nodes. The edge between nodes indicates the inclusion relationship of the grammatical structure between the linguistic hierarchies. Then, a simple but effective multi-step information propagation (MSIP) is proposed to incorporate the linguistics knowledge of heterogeneous graph to enhance Chinese PLMs inductively. In this way, we can obtain adequate information interaction among characters, words and sentences. Furthermore, the internal implementation of HLG is highly parallelized, which is conducive to GPU accelerate and raises the operating efficiency.

In summary, we abstract out an adapter component named enhancement module for PLMs to integrate external knowledge during the fine-tuning stage. In this paradigm, we further introduce HLG to integrate CWS information delicately and model it via an effective MSIP. Extensive experiments conducted on 10 benchmark datasets of 6 NLP tasks demonstrate that our model outperforms the BERT, BERTwwm and ERNIE 1.0 significantly and steadily. Comparing with MWA, a strong baseline that also incorporates CWS information to enhance PLMs, our model achieves a steady improvement with the same information. Meanwhile, compared with previous work, MWA, our proposed HLG introduces only half additional parameters and the training/inference speed is about 7x faster.

2 Preliminaries

2.1 Pre-trained Language Model (PLM)

As mentioned in §1, the pre-trained language models (PLMs) have achieved great success in many NLP applications with the 2-stage paradigm of pre-training and fine-tuning. The PLMs usually perform pre-training on large-scale unlabeled corpus in virtue of self-supervised reconstruction tasks. For example, BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) is a typical well-known PLM, which conducts masked language modeling and next sentence prediction as pre-training tasks. After completing the pre-training, the PLMs learn substantial contextualized text representations, and then adapt fine-tuning on specific downstream tasks.

In Chinese NLP, PLMs are generally character-based models (Li et al., 2019; Cui et al., 2019a). Specifically, given a character sequence:

$$S = [c_1, c_2, ..., c_n] \tag{1}$$

the outputs of Chinese PLMs can be treated as the character-level representations $\mathbf{H} \in \mathbb{R}^{n \times d}$, where the d is the dimension of representation.

2.2 Chinese Word Segmentation

As the same as most East-Asian languages, Chinese language is written without explicit word delimiters and the character is the smallest morpheme unit in Chinese linguistic (Cai and Zhao, 2016). Although character-based models could achieve good performance (Li et al., 2019), Li et al. (2020) point out that introducing Chinese Word Segmentation

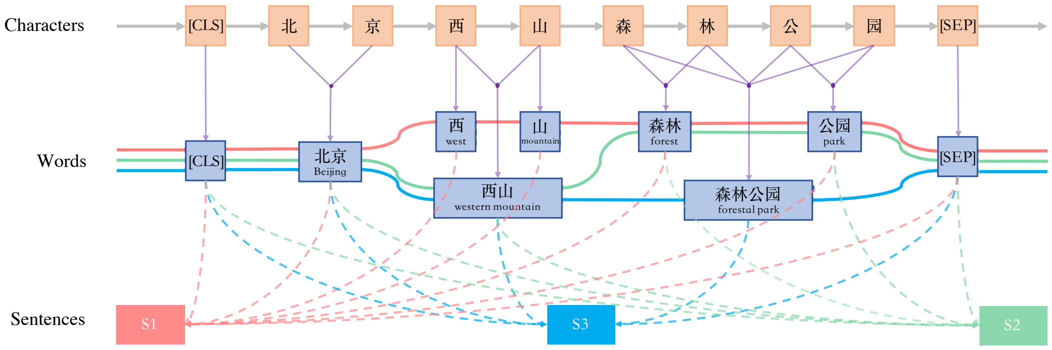


Figure 2: Overview of HLG structure. Different colored lines represent sentences formed by different CWS tools.

(CWS) information to character-based models can effectively improve the model performance.

We give a formality definition of segmenter and its partition strategy π. Given a sentence consisting of a sequence of characters as Eq. 1, a segmenter is defined as:

$$\textsc{Segmenter} \equiv \pi : S \rightarrow S'$$

where π is a partition strategy of sentence. Specifically, π partition and group the character sequence S into the word sequence S':

$$\pi(S) = S' = [w_1, w_2, ..., w_m] \qquad (2)$$

where $m \leq n$ and $w_i = [c_s, c_{s+1}, ..., c_{s+l-1}]$ is the i-th segmented word with a length of l and s is the index of w_i's first character in S. Namely, the word w_i is a sequence of characters $\{c_s, c_{s+1}, ..., c_{s+l-1}\}$, and the sentence S' is a sequence of words $\{w_1, w_2, ..., w_m\}$.

2.3 MWA for Enhancing Chinese PLM

Li et al. (2020) carried out researches on integrating CWS information into Chinese PLMs. The authors brought an architecture named Multi-source Word-aligned Attention (MWA) to incorporate multi-granularity segmentation via pooling attention weights among characters within the word.

Formally, given a character sequence S as Eq. 1 and its partition strategy π as Eq. 2. The character-based representation $\mathbf{H}$ could be gained via PLM, MWA conducted self-attention between characters:

$$\mathbf{A} = \texttt{softmax}\Big(\frac{(\mathbf{K}\mathbf{W}_k)(\mathbf{Q}\mathbf{W}_q)^T}{\sqrt{d}}\Big)$$

where $\mathbf{Q}$ and $\mathbf{K}$ are both $\mathbf{H}$, d is defined in §2.1, and $\mathbf{A}$ represents the attention score matrix. We decompose $\mathbf{A}$ over columns as $[\mathbf{a}^1, \mathbf{a}^2, ..., \mathbf{a}^n]$, and then perform partition π on it: $\pi(\mathbf{A}) = [\{\mathbf{a}^1, \mathbf{a}^2\}, \{\mathbf{a}^3\}, ...\{\mathbf{a}_c^s, ..., \mathbf{a}_c^{s+l-1}\} ..., \{\mathbf{a}^{n-1}, \mathbf{a}^n\}]$ where s and l are defined in §2.2. Pooling each group of partitioned columns:

$$\mathbf{a}_w^i = \texttt{MixPooling}(\{\mathbf{a}_c^s, ..., \mathbf{a}_c^{s+l-1}\})$$

in which MixPooling is defined in Yu et al. (2014). The gained $\mathbf{A}_w = [\mathbf{a}_w^1, \mathbf{a}_w^2, ..., \mathbf{a}_w^m] \in \mathbb{R}^{n \times m}$ is the character-to-word attention weight matrix. After performing alignment-wise multiply (Li et al., 2020) between character-to-word attention weight matrix $\mathbf{A}_w$ and the character-based representation $\mathbf{H}$, the enhanced character-based representation which integrates CWS information can be obtained.

In essence, the MWA conducts interaction between characters and words via character-to-word attention weight matrix $\mathbf{A}_w$, implicitly summary the information from characters, and performs MixPooling to aggregate the word-based segmentation information and concrete the character-level representation.

3 Heterogeneous Linguistics Graph

This section introduces the components of our model **HLG** which instantiates the enhancement module by exploiting the CWS information. We first briefly explain the graph convolutional network as our base encoder, and then describe the graph construction of HLG. Finally, we give the details of the multi-step information propagation (MSIP) to integrate the CWS information into PLMs.

3.1 Graph Convolutional Network

Graph Convolutional Network (GCN) (Bruna et al., 2014; Kipf and Welling, 2017; Defferrard et al., 2016) is a powerful tool to extend the convolution operation from the grid data to irregular graph data.

The basic idea of GCN is to aggregate the representations of neighbors to obtain better representation expression of nodes in the graph. For instance, consider a homogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ constructed by nodes set $\mathcal{V}$ and edges set $\mathcal{E}$. $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is a binary adjacency matrix where each element A_{ij} denotes whether node i has an edge with node j in the edge set $\mathcal{E}$. Formally, a standard GCN layer can be abstracted as:

$$\mathbf{H}_{out} = \sigma(\hat{\mathbf{A}}\mathbf{H}_{in}\mathbf{W}), \quad \hat{\mathbf{A}} = \texttt{Norm}(\mathbf{A}) \tag{3}$$

where $\mathbf{H}_{in}$ denotes the input representation matrix, $\mathbf{H}_{out}$ is the updated representation matrix, $\texttt{Norm}(\cdot)$ means row normalizing function, $\hat{\mathbf{A}}$ is the normalized adjacency matrix, $\sigma(\cdot)$ is the ReLU function and $\mathbf{W}$ is a parameter matrix. After such convolution operation, the representation $\mathbf{H}_{in}$ were aggregated rely on edge connections defined by $\mathbf{A}$, and transformed into $\mathbf{H}_{out}$ by linear multiplication and active function.

3.2 Graph Construction

We build a heterogeneous graph $\mathcal{G} = (\mathcal{C}, \mathcal{W}, \mathcal{S}, \mathcal{E})$ to model the structure of Chinese linguistic, where $\mathcal{C}$, $\mathcal{W}$, $\mathcal{S}$, $\mathcal{E}$ denote the character nodeset, word nodeset, sentence nodeset and edge set, respectively. Besides, different from homogeneous graph, HLG models relationship between three granularities of linguistic in a hierarchical way.

As presented in Figure 2, $\mathcal{G}$ is composed of three hierarchies including characters, words and sentences. In this case, we employed three different CWS tools, and got three different segmentation results, which resulted in three sentences with slightly different semantics. Note that the same word segmentation results in the same position obtained by different CWS tools will be regarded as the same word node to enhance the interaction (e.g., Beijing and park in Figure 2). This purpose is to **denoise** the mistake word nodes brought by segmenter error. If a word is segmented by multiple segmenters at the same time, the corresponding word node will have a higher vertex degree. Such nodes with higher betweenness centrality will lead to a stronger influence on the followed information propagation and achieve the effect of denoising intuitively, like the vote-based multi-model ensemble.

In HLG, only one adjacency matrix $\mathbf{A}$ is not enough to describe the hierarchical relationships between characters, words and sentences. Hence, we conduct two interaction matrices $\bar{\mathbf{A}}^{c2w} \in \mathbb{R}^{|\mathcal{W}|\times|\mathcal{C}|}$ and $\bar{\mathbf{A}}^{w2s} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{W}|}$ to indicate aforementioned relationships. To be specific, we take the $\bar{\mathbf{A}}^{c2w}$ as an example (the one for $\bar{\mathbf{A}}^{w2s}$ is analogous), the element $\bar{A}^{c2w}_{ij}$ denotes whether **word** i has an edge with **character** j in the edge set $\mathcal{E}$. Similar to Eq. 3, we also denote normalized interaction matrices as $\hat{\mathbf{A}}^{c2w}$ and $\hat{\mathbf{A}}^{w2s}$.

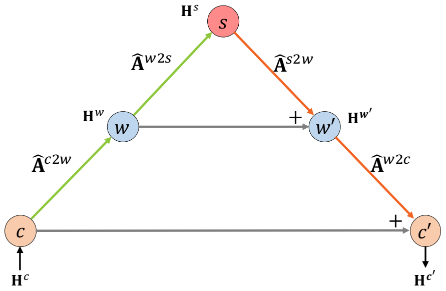


Figure 3: Illustration of learning procedure of MSIP. The colored circles denote characters, words or sentences representations. The green, orange and gray lines describe the summarization (Eq. 4), concretization (Eq. 6) and skip connection (Eq. 7) operations, respectively.

3.3 Multi-Step Information Propagation

To model the granularities hierarchical relationships in $\mathcal{G}$, we devise a multi-step information propagation to learn the linguistics knowledge. In CWS, the partition and group processes could be considered as the partition of semantic representation and the aggregation of separated semantic respectively (detailed in §2.2). Inspired by CWS processes, we introduce two operations into MSIP to simulate such processes and named as summarization and concretization. Figure 3 shows the information propagation procedure of MSIP.

Summarization. The summarization operation focuses on generalizing hierarchical word and sentence representations (e.g., from character-level to word-level). Specifically, given a heterogeneous graph $\mathcal{G}$ and corresponding character representations $\mathbf{H}^{c}$ from PLM, the summarization operation can be formulated as follows:

$$\begin{aligned} \mathbf{H}^{w} &= \sigma(\hat{\mathbf{A}}^{c2w}\mathbf{H}^{c}\mathbf{W}^{c2w}), \\ \mathbf{H}^{s} &= \sigma(\hat{\mathbf{A}}^{w2s}\mathbf{H}^{w}\mathbf{W}^{w2s}), \end{aligned} \tag{4}$$

where the $\mathbf{W}^{c2w}$, $\mathbf{W}^{w2s}$ are parameter matrices, $\mathbf{H}^{w}$, $\mathbf{H}^{s}$ are the interim representations of words and sentences.

Concretization. Concretization is the inverse operation of summarization, it is used to repartition the semantics from high-level to low-level (e.g. from sentence-level to word-level). To do so, we first calculate the normalized interaction matrices $\hat{\mathbf{A}}^{s2w}$ and $\hat{\mathbf{A}}^{w2c}$, which can be simply obtained by first transposed then normalized the predefined interaction matrices $\bar{\mathbf{A}}^{w2s}$ and $\bar{\mathbf{A}}^{c2w}$, respectively. Thus, we have:

$$\begin{aligned} \hat{\mathbf{A}}^{s2w} &= \mathtt{Norm}\big((\bar{\mathbf{A}}^{w2s})^\top\big), \\ \hat{\mathbf{A}}^{w2c} &= \mathtt{Norm}\big((\bar{\mathbf{A}}^{c2w})^\top\big), \end{aligned} \tag{5}$$

where $(\cdot)^\top$ is the transpose function. Afterward, the concretization operation is defined as follows:

$$\begin{aligned} \overline{\mathbf{H}}^{w'} &= \sigma(\hat{\mathbf{A}}^{s2w}\mathbf{H}^{s}\mathbf{W}^{s2w}), \\ \overline{\mathbf{H}}^{c'} &= \sigma(\hat{\mathbf{A}}^{w2c}\mathbf{H}^{w'}\mathbf{W}^{w2c}), \end{aligned} \tag{6}$$

where $\mathbf{W}^{s2w}$ and $\mathbf{W}^{w2c}$ are parameter matrices, $\overline{\mathbf{H}}^{w'}$ and $\overline{\mathbf{H}}^{c'}$ are also interim word and character representations, $\mathbf{H}^{w'}$ denote the final word representations defined in Eq. 7.

Skip Connection. Intuitively, it is difficult to generate satisfied low-level representations from the high-level representations directly. For example, it is easy to learn a few sentence representations from dozens of word representations, but hard to generate dozens of word representations from a few sentence representations.

To mitigate this problem, in this paper, we introduce the skip connection to enhance the MSIP, which is to simulate the self-loop in vanilla GCN. As shown in Figure 3, we add skip connections between the summarization representations and the concretization representations directly. Formally, the skip connection can be simply expressed as:

$$\begin{aligned} \mathbf{H}^{w'} &= \overline{\mathbf{H}}^{w'} + \sigma(\mathbf{H}^{w}\mathbf{W}^{w}), \\ \mathbf{H}^{c'} &= \overline{\mathbf{H}}^{c'} + \sigma(\mathbf{H}^{c}\mathbf{W}^{c}), \end{aligned} \tag{7}$$

where $\mathbf{W}^{w}$ and $\mathbf{W}^{c}$ are parameter matrices. Furthermore, $\mathbf{H}^{c'}$ denote the final representations for characters, which incorporates the fine-grained linguistics knowledge in $\mathcal{G}$.

4 Experiments

4.1 Experimental Setting

For a fair comparison with MWA, which also gives an enhancement module by incorporating CWS information. We conduct the same experiments on five NLP tasks with various benchmark datasets. Three frequently-used Chinese PLMs: BERT (Devlin et al., 2019), ERNIE 1.0 (Sun et al., 2019) and BERTwwm (Cui et al., 2019a) are employed as the basic PLM to enhance. Three CWS tools: thulac (Sun et al., 2016a), ictclas (Zhang et al., 2003) and hanlp (He, 2014) are employed to gain the segmentation information. The time of pre-processing including applying CWS tools is ignored in the experimental report. In the production, preprocessing and inference can be asynchronously executed in parallel (while inference a batch of data, the subsequence data can be preprocessed with multiprocess) (Cheng et al., 2019), all three of the CWS tools we've introduced are fast enough to achieve this effect. According to rough estimates and technical reports, the processing speed of these tools are thulac 1221KB/s, ictclas 769KB/s, hanlp 1375KB/s, respectively.

Specifically, we instantiate the enhancement module as HLG and incorporate with downstream task-specific model. To verify the effectiveness of HLG, we execute 5 times fine-tuning on 10 benchmark datasets of 6 NLP tasks and report the average score. The tasks include Sentiment Classification (SC), Document Classification (DC), Named Entity Recognition (NER), Sentence Pair Matching (SPM), Natural Language Inference (NLI) and Machine Reading Comprehension (MRC). Specifically, the following benchmark datasets are chosen to evaluate the performance: 1) **SC**: ChnSenti[1] and weibo100k[2] sentiment datasets are used for evaluating the capacity of short text classification. 2) **DC**: THUCNews (Sun et al., 2016b) dataset contains 10 types of news for performing long text classification. 3) **NER**: Ontonotes 4.0 (Weischedel et al., 2011) and MSRA-NER (Levow, 2006a) are used for testing model in sequence tagging task. 4) **SPM**: LCQMC (Liu et al., 2018) and BQ (Chen et al., 2018) are used to evaluate the text matching ability of model. 5) **NLI**: We conduct experiments on the Chinese part of XNLI (Conneau et al., 2018) dataset, and adopt the same pre-processing strategy as ERNIE (Sun et al., 2019). 6) **MRC**: Commonly used datasets DRCD (Shao et al., 2018) and CMRC2018 (Cui et al., 2019b) are tested. CMRC2018 is only evaluated on dev set as same as (Wei et al., 2019; Sun et al., 2020).

[1] https://github.com/pengming617/bert_classification

[2] https://github.com/SophonPlus/ChineseNlpCorpus/

	SC		NER		SPM	
	ChnSenti	weibo100k	MSRA-NER	ontonotes	LCQMC	BQ
BERT	94.72	97.31	93.62	79.18	86.50	84.73
+MWA	95.34(+0.62)	98.14(+0.83)	**93.86**(+0.24)	79.86(+0.68)	86.92(+0.42)	84.87(+0.14)
+HLG	**95.83**(+1.11)	**98.17**(+0.86)	93.82(+0.20)	**80.42**(+1.24)	**87.79**(+1.29)	**85.01**(+0.28)
BERTwwm	94.38	97.36	93.83	79.28	86.11	84.75
+MWA	95.01(+0.63)	**98.13**(+0.77)	93.84(+0.01)	80.32(+1.04)	86.28(+0.17)	**85.02**(+0.27)
+HLG	**95.25**(+0.87)	98.11(+0.75)	**93.96**(+0.13)	**80.46**(+1.18)	**88.13**(+2.02)	84.98(+0.23)
ERNIE 1.0	95.17	97.30	93.97	77.74	87.27	84.78
+MWA	95.52(+0.35)	98.18(+0.88)	**94.04**(+0.07)	78.78(+1.04)	87.58(+0.31)	**85.06**(+0.28)
+HLG	**95.83**(+0.66)	**98.22**(+0.92)	**94.04**(+0.07)	**79.16**(+1.42)	**87.80**(+0.53)	85.04(+0.26)

	DC	NLI	MRC			
	THUNews	XNLI	DRCD[EM | F1]		CMRC2018[EM | F1]	
BERT	96.78	78.19	85.57	91.16	66.36	85.88
+MWA	97.13(+0.35)	78.42(+0.23)	86.86(+1.29)	92.22(+1.06)	67.21(+0.85)	86.22(+0.34)
+HLG	**97.20**(+0.42)	**78.68**(+0.49)	**86.96**(+1.39)	**92.28**(+1.12)	**67.30**(+0.94)	**86.27**(+0.39)
BERTwwm	97.01	77.92	84.11	90.46	66.20	85.85
+MWA	97.28(+0.27)	78.68(+0.76)	**87.00**(+2.89)	**92.21**(+1.75)	67.43(+1.23)	86.49(+0.64)
+HLG	**97.32**(+0.31)	**79.01**(+1.09)	86.92(+2.81)	92.15(+1.69)	**67.51**(+1.31)	**86.53**(+0.68)
ERNIE 1.0	97.04	78.04	87.85	92.85	65.74	85.78
+MWA	97.34(+0.30)	78.71(+0.67)	**88.61**(+0.76)	**93.72**(+0.87)	**67.12**(+1.38)	**86.30**(+0.52)
+HLG	**97.35**(+0.31)	**78.80**(+0.76)	88.58(+0.73)	93.60(+0.75)	67.03(+1.29)	86.26(+0.48)

Table 1: The experimental results on various datasets. All of the experiments except CMRC2018 are conducted on test set, the reported values are F1 unless specified (EM means exact match score). We run each experiment with a random seed for five times and report the average score. Numbers in brackets indicate the relative increment brought by enhancement module. The bold numbers mark the highest value within the same base-model.

We implement the presented approach in PyTorch and fine-tune the downstream tasks on multiple Nvidia Tesla V100 GPUs. The basic architecture of PLMs and pre-trained parameters are provided by Huggingface (Wolf et al., 2020). The initial learning rate and other hyper-parameters refer to the previous works reported (Cui et al., 2019a; Li et al., 2020; Sun et al., 2020). Since the parameters of PLMs have been optimized, while the parameters of HLG and the downstream tasks are untrained. Hence, the learning rate of HLG part is larger than PLM part, we manually tuned the learning rates of PLM and HLG separately.

4.2 Experimental Results

The experimental results are shown in Table 1. Overall, we can observe that both HLG and MWA outperform baseline models (BERT, BERTwwm and ERNIE 1.0). Comparing with WMA, HLG achieves further improvement and significantly outperforms baseline models on 10 tasks. In detail, HLG outperforms MWA on ChnSent, weibo100k, MSRA-NER, ontonotes, LCQMC, BQ, THUCNews and XNLI tasks, and obtains comparable results on DRCD and CMRC2018 datasets.

For the text classification tasks, namely **SC** and **DC**, HLG respectively achieves 0.88% and 0.84% average improvement on ChnSenti and weibo100k dataset, while MWA gains 0.53% and 0.82%. Meanwhile, HLG obtains 0.35% improvement on the long text multi-classification benchmark THUCNews, and MWA gets 0.31% points.

Comparing with text classification tasks, the improvements over **NER** tasks are more obvious. The main reason may be that CWS explicitly provides the word boundaries, which are important to recognize entities accurately. On the ontonotes dataset, the promotion of HLG (1.28% averagely) is distinctly higher than that of MWA (0.92% averagely). Compared to the strong baseline models, the F1 scores of MSRA-NER have improved average 0.13% and 0.10% by HLG and MWA, respectively.

HLG achieves the best results on the text matching tasks (**SPM**) and its variant **NLI**, which brings 1.28% average improvement to LCQMC, 0.26% average improvement to BQ, and 0.78% average improvement to XNLI. The improvements of HLG are much higher than that of MWA (0.3%, 0.23% and 0.55%). As described in Chen et al. (2020) and

No.	CWS tool	Accumulative Word Count	
		ChnSenti	weibo100k
0	None	0	0
1	thulac(Sun et al., 2016a)	69,877	398,046
2	ictclas(Zhang et al., 2003)	78,695	452,059
3	hanlp(He, 2014)	82,768	479,134
4	pkuseg(Luo et al., 2019)	84,273	481,201
5	jieba(Sun, 2013)	85,062	483,390

Table 2: Adding the CWS tools one by one, and accumulate the total number of word nodes.

Lyu et al. (2021), text matching tasks can benefit from the interaction between the paired sentences. HLG follows them to construct graphs over sentence pairs collectively, which naturally obtains advantages in text matching tasks.

For **MRC** task, HLG and MWA achieve comparable results on those datasets. HLG gets an average improvement of 1.41 in EM and 0.85 in F1 score, while MWA gets 1.4 EM and 0.86 F1 score. However, HLG has dominant advantage in training speed and inference speed. Detail analysis of time efficiency is in §4.3.3.

4.3 Analyses

4.3.1 Ablation Study

We conduct ablation experiments to explore the effectiveness of the number of CWS tools. The ablation experiments are organized on sentiment classification task, ChnSenti and weibo100k dev set. As shown in Table 2, 5 popular CWS tools are added into our model successively according to the order, and we also show the total number of word nodes in our HLG. Meanwhile, the information from multiple word segmentation tools can be integrated at the same time without increasing parameter size in HLG (only the **A** is changed).

Figure 4 shows the performance of BERT+HLG with different numbers of CWS tools on ChnSenti and weibo100k dev sets. Experimental results demonstrate the effectiveness of introducing word segmentation information. We can observe that when the number of CWS tools is larger, the number of generated word nodes gradually increasing to converge, and the performance of the model slightly is not always increasing as the word count.

The more CWS tools introduced will bring more diversity but also bring noise caused by segmenter error. In practice, we find using 4 or more CWS tools can slightly increase the performance but take much longer preprocessing time, hence we select the elbow of the curve as the number of CWS tools.

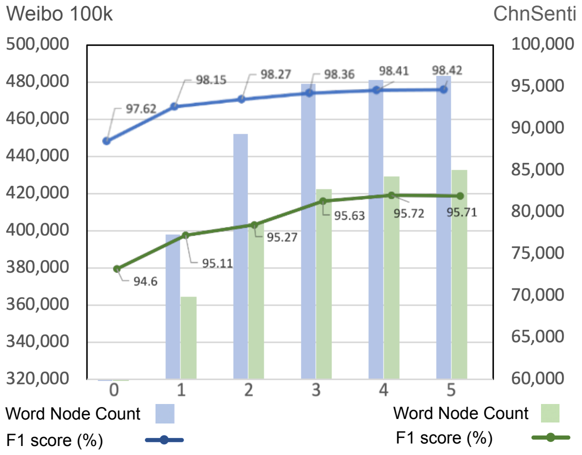


Figure 4: The histogram chart is the cumulative number of word nodes obtained by CWS tools, and the line chart is the performance of the model (BERT+HLG) in dev-set with the corresponding number of CWS tools.

Model	Params. ($K = 3$)	F1
BERT	110M	79.28
+MWA	117.7M(+7.7M)	79.68(+0.40)
+HLG	113.5M(+**3.5M**)	79.75(+0.47)
BERTwwm	110M	79.32
+MWA	117.7M(+7.7M)	79.77(+0.45)
+HLG	113.5M(+**3.5M**)	80.16(+0.84)
ERNIE 1.0	110M	79.75
+MWA	117.7M(+7.7M)	79.98(+0.23)
+HLG	113.5M(+**3.5M**)	80.21(+0.46)

Table 3: The amount of additional parameters and performance improvement of MWA and HLG.

That is, using 3 as the number of CWS tools might be a balance between the performance of model and the cost of preprocessing. This number also coincides with the configuration in MWA.

4.3.2 Parameter-Efficient Analysis

In general, the enhancement module should be able to bring performance improvements without unacceptable space complexity. Therefore, we conduct a comparative experiment on XNLI dev set to explore the performance improvement and the space overhead between MWA and HLG.

To be specific, the number of parameters in MWA depends on the dimension of PLM's representation and the number of CWS tools K. Concretely, MWA contains K transformer layers and 1 aggregation layer. Nevertheless, our HLG only depends on the dimension of PLM's representation and simply contains 4 basic GCN layers and 2 skip connections. Thus, the number of parameters of

Model	Params.	F1
BERTwwm-base	110M	79.32
+HLG(`random tokenizer`)	113.5M	79.16(-0.16)
+HLG(`character tokenizer`)	113.5M	79.41(+0.09)
+HLG(`thulac`)	113.5M	79.68(+0.36)
+HLG(`ictlas`)	113.5M	79.91(+0.59)
+HLG(`hanlp`)	113.5M	79.81(+0.49)
+HLG(`thulac+ictlas+hanlp`)	113.5M	80.16(+0.84)

Table 4: The performance comparison between random tokenizer, character tokenizer that segments each character into a single word, and sole segmenters.

them can be calculated as:

$$size(\text{MWA}) = K \times (4 \times d^2)_{\text{Transformer}} + d^2$$

$$size(\text{HLG}) = (4 \times d^2)_{\text{GCN}} + 2 \times d^2$$

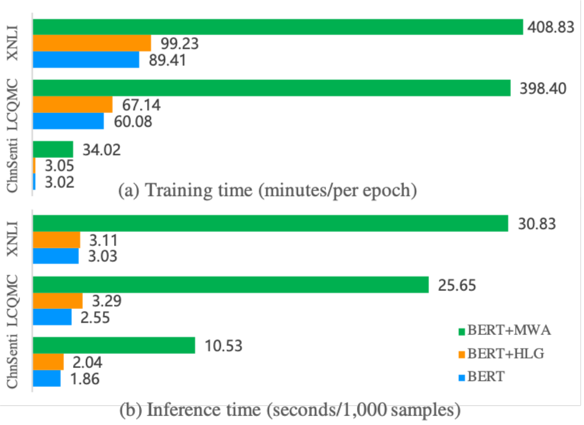
As discussed before, we employ 3 CWS tools in both MWA and HLG. Table 3 reports the performance of BERT, BERTwwm, and ERNIE 1.0 on the XNLI dev set. Obviously, HLG can get a greater performance improvement with only half additional parameters. It shows that as an enhancement module, HLG is superior to MWA in terms of parameter utilization efficiency.

In addition, **to verify the impact of the additional parameters**, we also conduct an ablation experiment on XNLI dev set that utilizes the random tokenizer, the single-character tokenizer, and sole segmenter to obtain the different word segmentation results, and send those results to HLG to eliminate the additional benefit from the change of neural network structure and the increase of parameters. The results are shown in Table 4, which indicates that the increment of parameters can slightly affect character-based model performance, and the CWS information is significantly useful to promote the performance of character-based PLM.

4.3.3 Time Efficiency Analysis

Time efficiency is an important indicator in the real-world production. Less training time and inference time means lower costs. In order to analyze the additional time cost of different enhancement modules, we conduct comparative experiments among BERT, BERT+MWA, and BERT+HLG on ChnSenti, LCQMC and XNLI datasets. For the fair comparison, we remain other hyper-parameters consistent for the three models.

As shown in Figure 5, we compare time cost during training and inference between vanilla BERT, BERT+MWA and BERT+HLG. We can observe that the training time and inference time of BERT+HLG are basically consistent with vanilla BERT. However, when MWA is introduced, the average training time increases by 7 times, and the average inference time increases by 7.6 times. This is because MWA must calculate aligned attention weights token by token, and it cannot benefit from CUDNN parallelization, resulting in terrible operating efficiency. On the contrary, HLG is composed of GCNs, and its internal implementation is basically the simplest non-linear transformation. Therefore, HLG could be maximally accelerated through the optimized matrix operation of CUDNN primitive, which only produces a negligible impact on time efficiency.



(a) Training time (minutes/per epoch)

(b) Inference time (seconds/1,000 samples)

Figure 5: The training time, inference time of vanilla BERT, BERT+MWA and BERT+HLG on ChnSenti, LCQMC and XNLI benchmarks. All of these time dose not include CWS process.

5 Related Works

Pre-training language models, such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), XLNET (Yang et al., 2019) and GPT (Radford et al., 2018), have shown their powerful performances on various natural language processing tasks and have been applied in many applications.

In recent past, there are studies adapting PLMs for Chinese with Chinese-specific features such as word information. Glyce (Meng et al., 2019) proposed to use the glyph information of Chinese characters to enhance PLMs. ERNIE 1.0/2.0 (Sun et al., 2019, 2020) and BERTwwm (Cui et al., 2019a) used the whole word mask to learn the structure of words or entities in the pre-training stage and conducted more and better pre-training tasks to perceive large-scale data. NEZHA (Wei et al., 2019) used a series of methods such as functional relative positional encoding and whole word masking to improve the pre-training tasks, which had brought improvement. ZEN (Diao et al., 2020) adopted n-gram masking to enhance pre-trained encoder and obtained outstanding performance. Lattice-BERT (Lai et al., 2021) introduced word lattice information (Zhang and Yang, 2018) into pre-training framework via lattice position attention.

As a fundamental feature of Chinese, word segmentation information is flexibility, granularity, and easy-to-get (Sproat and Emerson, 2003; Levow, 2006b). Further, Zhang et al. (2018); Li et al. (2019, 2020) conducted detailed research and experiments on the application of CWS in deep learning, and found that CWS information can effectively improve the performance of Chinese character-based PLMs.

Recently, a lot of works have been proposed to prompt NLP applications by constructing graph on text and modeling with graph neural networks. Yao et al. (2019) first constructed word co-occurrence graph between documents and introduced GCN to modeling and aggregating document representation for text classification. Chen et al. (2020); Lyu et al. (2021) constructed lattice graph to maintain multi-granularity information and external knowledge in Chinese short text matching task. Nguyen and Grishman (2018) proposed performing GCN over dependency trees to extract event trigger. Sui et al. (2019) conducted a character-word interaction graph and performed graph attention network on it to recognize Chinese named entities. Shu et al. (2020) introduced a bipartite-graph based transformer PLM for integrating hierarchical semantic information.

6 Conclusion

In this paper, we propose HLG which acts as the enhancement module to enhance Chinese PLMs with CWS information. The HLG firstly constructs heterogeneous graph based on multiple word segmentations to model the hierarchy of Chinese. Then, the MSIP is proposed to model the fine-grained linguistics knowledge of the heterogeneous graph. Experimental results on 6 NLP tasks with 10 benchmark datasets demonstrate that the performance of our model outperforms previous work, MWA. Besides the performance improvements, HLG introduces only half the additional parameters of MWA and its training/inference speed is 7x faster than MWA. At present, the experimental results of HLG are lagging behind SOTA, and we will try to migrate it to some of the latest PLMs. Besides, HLG has the expansibility to introduce the representation layer of the CWS model directly, or introduce some other information sources such as the knowledge graph, etc. We leave these further improvements to the future.

Acknowledgement

This work is supported by the National Key Research and Development Program of China (grant No.2021YFB3100600 and 2020YFB2103803), the Strategic Priority Research Program of Chinese Academy of Sciences (grant No.XDC02040400), the Youth Innovation Promotion Association of CAS (Grant No. 2021153) and National Natural Science Foundation of China (Grant No.61902394). This work performed while the first author was at IIE, CAS.

References

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral networks and locally connected networks on graphs. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

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.

Jing Chen, Qingcai Chen, Xin Liu, Haijun Yang, Daohe Lu, and Buzhou Tang. 2018. The BQ corpus: A large-scale domain-specific Chinese corpus for sentence semantic equivalence identification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4946–4951, Brussels, Belgium. Association for Computational Linguistics.

Lu Chen, Yanbin Zhao, Boer Lyu, Lesheng Jin, Zhi Chen, Su Zhu, and Kai Yu. 2020. Neural graph matching networks for Chinese short text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6152–6158, Online. Association for Computational Linguistics.

Yang Cheng, Dan Li, Zhiyuan Guo, Binyao Jiang, Jiaxin Lin, Xi Fan, Jinkun Geng, Xinyi Yu, Wei Bai, Lei Qu, et al. 2019. Dlbooster: Boosting end-to-end deep learning workflows with offloading data preprocessing pipelines. In *Proceedings of the 48th International Conference on Parallel Processing*, pages 1–11.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

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

Yiming Cui, Ting Liu, Wanxiang Che, Li Xiao, Zhipeng Chen, Wentao Ma, Shijin Wang, and Guoping Hu. 2019b. A span-extraction dataset for Chinese machine reading comprehension. 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)*, pages 5883–5889, Hong Kong, China. Association for Computational Linguistics.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3837–3845.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Shizhe Diao, Jiaxin Bai, Yan Song, Tong Zhang, and Yonggang Wang. 2020. ZEN: Pre-training Chinese text encoder enhanced by n-gram representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4729–4740, Online. Association for Computational Linguistics.

Han He. 2014. HanLP: Han Language Processing. *GitHub Repository*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Yuxuan Lai, Yijia Liu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2021. Lattice-BERT: Leveraging multi-granularity representations in Chinese pre-trained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1716–1731, Online. Association for Computational Linguistics.

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

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

Xiaoya Li, Yuxian Meng, Xiaofei Sun, Qinghong Han, Arianna Yuan, and Jiwei Li. 2019. Is word segmentation necessary for deep learning of Chinese representations? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3242–3252, Florence, Italy. Association for Computational Linguistics.

Yanzeng Li, Bowen Yu, Xue Mengge, and Tingwen Liu. 2020. Enhancing pre-trained Chinese character representation with word-aligned attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3442–3448, Online. Association for Computational Linguistics.

Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. 2018. LCQMC:a large-scale Chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1952–1962, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ruixuan Luo, Jingjing Xu, Yi Zhang, Xuancheng Ren, and Xu Sun. 2019. Pkuseg: A toolkit for multi-domain chinese word segmentation. *arXiv preprint arXiv:1906.11455*.

Boer Lyu, Lu Chen, Su Zhu, and Kai Yu. 2021. Let: Linguistic knowledge enhanced graph transformer for chinese short text matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13498–13506.

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 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2742–2753.

Thien Huu Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. 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), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5900–5907. AAAI Press.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning.

Chih Chieh Shao, Trois Liu, Yuting Lai, Yiying Tseng, and Sam Tsai. 2018. Drcd: a chinese machine reading comprehension dataset. *arXiv preprint arXiv:1806.00920*.

Xiaobo Shu, Mengge Xue, Yanzeng Li, Zhenyu Zhang, and Tingwen Liu. 2020. Big-transformer: Integrating hierarchical features for transformer via bipartite graph. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Richard Sproat and Thomas Emerson. 2003. The first international Chinese word segmentation bakeoff. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 133–143, Sapporo, Japan. Association for Computational Linguistics.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 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)*, pages 3830–3840, Hong Kong, China. Association for Computational Linguistics.

J Sun. 2013. Jieba chinese word segmentation tool. *GitHub Repository*.

Maosong Sun, Xinxiong Chen, Kaixu Zhang, Zhipeng Guo, and Zhiyuan Liu. 2016a. Thulac: An efficient lexical analyzer for chinese. Technical report.

Maosong Sun, Jingyang Li, Zhipeng Guo, Z Yu, Y Zheng, X Si, and Z Liu. 2016b. Thuctc: an efficient chinese text classifier. *GitHub Repository*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8968–8975.

Junqiu Wei, Xiaozhe Ren, Xiaoguang Li, Wenyong Huang, Yi Liao, Yasheng Wang, Jiashu Lin, Xin Jiang, Xiao Chen, and Qun Liu. 2019. Nezha: Neural contextualized representation for chinese language understanding. *arXiv preprint arXiv:1909.00204*.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2011. Ontonotes 4.0. *Linguistic Data Consortium LDC2011T03*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377.

Dingjun Yu, Hanli Wang, Peiqiu Chen, and Zhihua Wei. 2014. Mixed pooling for convolutional neural networks. In *International Conference on Rough Sets and Knowledge Technology*, pages 364–375. Springer.

Hua-Ping Zhang, Hong-Kui Yu, De-Yi Xiong, and Qun Liu. 2003. HHMM-based Chinese lexical analyzer ICTCLAS. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 184–187, Sapporo, Japan. Association for Computational Linguistics.

Qi Zhang, Xiaoyu Liu, and Jinlan Fu. 2018. Neural networks incorporating dictionaries for chinese word segmentation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5682–5689. AAAI Press.

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