Lexicon Enhanced Chinese Sequence Labelling Using BERT Adapter

Wei Liu[1], **Xiyan Fu**[2], **Yue Zhang**[3], **Wenming Xiao**[1]
[1]DAMO Academy, Alibaba Group, China
[2]College of Computer Science, Nankai University, China
[3]School of Engineering, Westlake University, China
[3]Institute of Advanced Technology, Westlake Institute for Advanced Study
hezan.lw@alibaba-inc.com, fuxiyan@mail.nankai.edu.cn,
yue.zhang@wias.org.cn, wenming.xiaowm@alibaba-inc.com

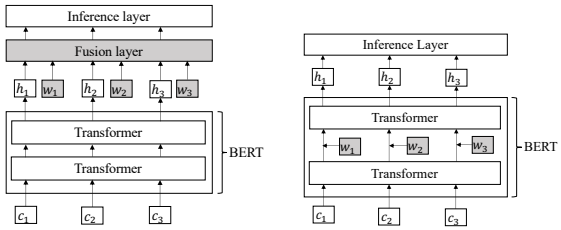
Abstract

Lexicon information and pre-trained models, such as BERT, have been combined to explore Chinese sequence labeling tasks due to their respective strengths. However, existing methods solely fuse lexicon features via a shallow and random initialized sequence layer and do not integrate them into the bottom layers of BERT. In this paper, we propose Lexicon Enhanced BERT (LEBERT) for Chinese sequence labeling, which integrates external lexicon knowledge into BERT layers directly by a Lexicon Adapter layer. Compared with existing methods, our model facilitates deep lexicon knowledge fusion at the lower layers of BERT. Experiments on ten Chinese datasets of three tasks including Named Entity Recognition, Word Segmentation, and Part-of-Speech Tagging, show that LEBERT achieves state-of-the-art results.

1 Introduction

Sequence labeling is a classic task in natural language processing (NLP), which is to assign a label to each unit in a sequence (Jurafsky and Martin, 2009). Many important language processing tasks can be converted into this problem, such as part-of-speech (POS) tagging, named entity recognition (NER), and text chunking. The current state-of-the-art results for sequence labeling have been achieved by neural network approaches (Lample et al., 2016; Ma and Hovy, 2016; Chiu and Nichols, 2016; Gui et al., 2017).

Chinese sequence labeling is more challenging due to the lack of explicit word boundaries in Chinese sentences. One way of performing Chinese sequence labeling is to perform Chinese word segmentation (CWS) first, before applying word sequence labeling (Sun and Uszkoreit, 2012; Yang et al., 2016). However, it can suffer from the segmentation errors propagated from the CWS system (Zhang and Yang, 2018; Liu et al., 2019).



(a) Model-level Fusion (b) BERT-level Fusion

Figure 1: Comparison of fusing lexicon features and BERT at different levels for Chinese sequence labeling. For simplicity, we only show two Transformer layers in BERT and truncate the sentence to three characters. c_i denotes the i-th Chinese character, w_j denotes the j-th Chinese word.

Therefore, some approaches (Cao et al., 2018; Shen et al., 2016) perform Chinese sequence labeling directly at the character level, which has been empirically proven to be more effective (Ng and Low, 2004; Liu et al., 2010; Zhang and Yang, 2018).

There are two lines of recent work enhancing character-based neural Chinese sequence labeling. The first considers integrating word information into a character-based sequence encoder, so that word features can be explicitly modeled (Zhang and Yang, 2018; Yang et al., 2019; Liu et al., 2019; Ding et al., 2019; Higashiyama et al., 2019). These methods can be treated as designing different variants to neural architectures for integrating discrete structured knowledge. The second considers the integration of large-scale pre-trained contextualized embeddings, such as BERT (Devlin et al., 2019), which has been shown to capture implicit word-level syntactic and semantic knowledge (Goldberg, 2019; Hewitt and Manning, 2019).

The two lines of work are complementary to each other due to the different nature of discrete

and neural representations. Recent work considers the combination of lexicon features and BERT for Chinese NER (Ma et al., 2020; Li et al., 2020), Chinese Word Segmentation (Gan and Zhang, 2020), and Chinese POS tagging (Tian et al., 2020b). The main idea is to integrate contextual representations from BERT and lexicon features into a neural sequence labeling model (shown in Figure 1 (a)). However, these approaches do not fully exploit the representation power of BERT, because the external features are not integrated into the bottom level.

Inspired by the work about BERT Adapter (Houlsby et al., 2019; Bapna and Firat, 2019; Wang et al., 2020), we propose Lexicon Enhanced BERT (LEBERT) to integrate lexicon information between Transformer layers of BERT directly. Specifically, a Chinese sentence is converted into a char-words pair sequence by matching the sentence with an existing lexicon. A lexicon adapter is designed to dynamically extract the most relevant matched words for each character using a char-to-word bilinear attention mechanism. The lexicon adapter is applied between adjacent transformers in BERT (shown in Figure 1 (b)) so that lexicon features and BERT representation interact sufficiently through the multi-layer encoder within BERT. We fine-tune both the BERT and lexicon adapter during training to make full use of word information, which is considerably different from the BERT Adapter (it fixes BERT parameters).

We investigate the effectiveness of LEBERT on three Chinese sequence labeling tasks[1], including Chinese NER, Chinese Word Segmentation[2], and Chinese POS tagging. Experimental results on ten benchmark datasets illustrate the effectiveness of our model, where state-of-the-art performance is achieved for each task on all datasets. In addition, we provide comprehensive comparisons and detailed analyses, which empirically confirm that bottom-level feature integration contributes to span boundary detection and span type determination.

2 Related Work

Our work is related to existing neural methods using lexicon features and pre-trained models to improve Chinese sequence labeling.

[1] https://github.com/liuwei1206/LEBERT

[2] We follow the mainstream methods and regard Chinese Word Segmentation as a sequence labeling problem.

Lexicon-based. Lexicon-based models aim to enhance character-based models with lexicon information. Zhang and Yang (2018) introduced a lattice LSTM to encode both characters and words for Chinese NER. It is further improved by following efforts in terms of training efficiency (Gui et al., 2019a; Ma et al., 2020), model degradation (Liu et al., 2019), graph structure (Gui et al., 2019b; Ding et al., 2019), and removing the dependency of the lexicon (Zhu and Wang, 2019). Lexicon information has also been shown helpful for Chinese Word Segmentation (CWS) and Part-of-speech (POS) tagging. Yang et al. (2019) applied a lattice LSTM for CWS, showing good performance. Zhao et al. (2020) improved the results of CWS with lexicon-enhanced adaptive attention. Tian et al. (2020b) enhanced the character-based Chinese POS tagging model with a multi-channel attention of N-grams.

Pre-trained Model-based. Transformer-based pre-trained models, such as BERT (Devlin et al., 2019), have shown excellent performance for Chinese sequence labeling. Yang (2019) simply added a softmax on BERT, achieving state-of-the-art performance on CWS. Meng et al. (2019); Hu and Verberne (2020) showed that models using the character features from BERT outperform the static embedding-based approaches by a large margin for Chinese NER and Chinese POS tagging.

Hybrid Model. Recent work tries to integrate the lexicon and pre-trained models by utilizing their respective strengths. Ma et al. (2020) concatenated separate features, BERT representation and lexicon information, and input them into a shallow fusion layer (LSTM) for Chinese NER. Li et al. (2020) proposed a shallow Flat-Lattice Transformer to handle the character-word graph, in which the fusion is still at model-level. Similarly, character N-gram features and BERT vectors are concatenated for joint training CWS and POS tagging (Tian et al., 2020b). Our method is in line with the above approaches trying to combine lexicon information and BERT. The difference is that we integrate lexicon into the bottom level, allowing in-depth knowledge interaction within BERT.

There is also work employing lexicon to guide pre-training. ERNIE (Sun et al., 2019a,b) exploited entity-level and word-level masking to integrate knowledge into BERT in an implicit way. Jia et al. (2020) proposed Entity Enhanced BERT, further pre-training BERT using a domain-

specific corpus and entity set with a carefully designed character-entity Transformer. ZEN (Diao et al., 2020) enhanced Chinese BERT with a multi-layered N-gram encoder but is limited by the small size of the N-gram vocabulary. Compared to the above pre-training methods, our model integrates lexicon information into BERT using an adapter, which is more efficient and requires no raw texts or entity set.

BERT Adapter. BERT Adapter (Houlsby et al., 2019) aims to learn task-specific parameters for the downstream tasks. Specifically, they add adapters between layers of a pre-trained model and tune only the parameters in the added adapters for a certain task. Bapna and Firat (2019) injected task-specific adapter layers into pre-trained models for neural machine translation. MAD-X (Pfeiffer et al., 2020) is an adapter-based framework that enables high portability and parameter-efficient transfer to arbitrary tasks. Wang et al. (2020) proposed K-ADAPTER to infuse knowledge into pre-trained models with further pre-training. Similar to them, we use a lexicon adapter to integrate lexicon information into BERT. The main difference is that our goal is to better fuse lexicon and BERT at the bottom-level rather than efficient training. To achieve it, we fine-tune the original parameters of BERT instead of fixing them, since directly injecting lexicon features into BERT will affect the performance due to the difference between that two information.

3 Method

The main architecture of the proposed Lexicon Enhanced BERT is shown in Figure 2. Compared to BERT, LEBERT has two main differences. First, LEBERT takes both character and lexicon features as the input given that the Chinese sentence is converted to a character-words pair sequence. Second, a lexicon adapter is attached between Transformer layers, allowing lexicon knowledge integrated into BERT effectively.

In this section we describe: 1) Char-words Pair Sequence (Section 3.1), which incorporates words into a character sequence naturally; 2) Lexicon Adapter (Section 3.2), by injecting external lexicon features into BERT; 3) Lexicon Enhanced BERT (Section 3.3), by applying the Lexicon Adapter to BERT.

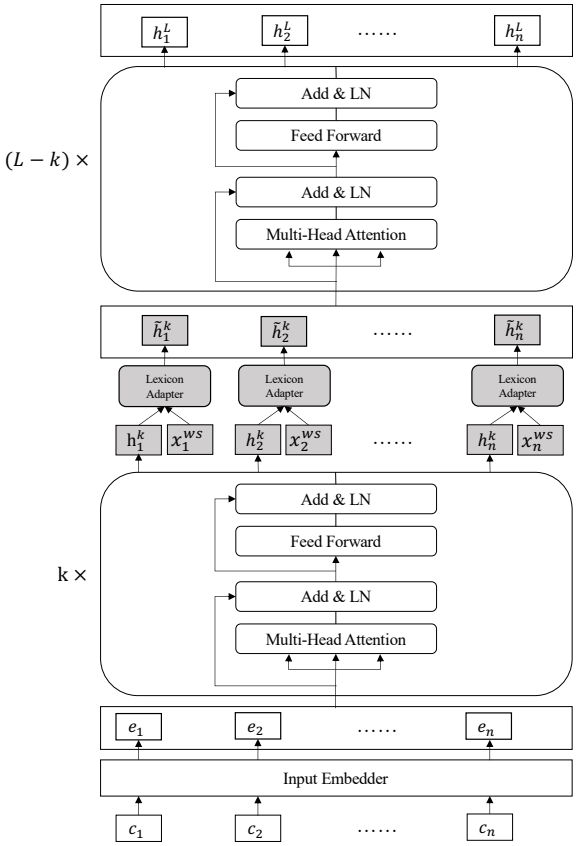


Figure 2: The architecture of Lexicon Enhanced BERT, in which lexicon features are integrated between k-th and $(k+1)$-th Transformer Layer using Lexicon Adapter. Where c_i denote the i-th Chinese character in the sentence, and x_i^{ws} denotes matched words assigned to character c_i.

3.1 Char-Words Pair Sequence

A Chinese sentence is usually represented as a character sequence, containing character-level features solely. To make use of lexicon information, we extend the character sequence to a character-words pair sequence.

Given a Chinese Lexicon $\mathbf{D}$ and a Chinese sentence with n characters $\mathbf{s_c} = \{c_1, c_2, ..., c_n\}$, we find out all the potential words inside the sentence by matching the character sequence with $\mathbf{D}$. Specifically, we first build a Trie based on the $\mathbf{D}$, then traverse all the character subsequences of the sentence and match them with the Trie to obtain all potential words. Taking the truncated sentence "美国人民 (American People)" for example, we can find out four different words, namely "美国 (America)", "美国人 (American)", "国人 (Compatriot)", "人民 (People)". Subsequently, for each matched word, we assign it to the characters it contains. As shown in Figure 3,

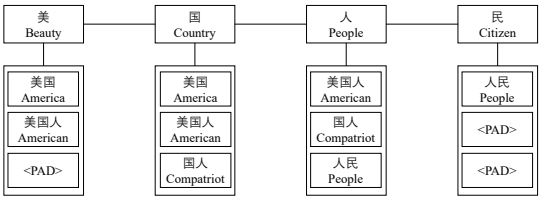


Figure 3: Character-words pair sequence of a truncated Chinese sentence "美国人民 (American People)". There are four potential words, namely "美国 (America)", "美国人 (American)", "国人 (Compatriot)", "人民 (People)". "<PAD>" denotes padding value and each word is assigned to the characters it contains.

the matched word "美国 (America)" is assigned to the character "美" and "国" since they form that word. Finally, we pair each character with assigned words and convert a Chinese sentence into a character-words pair sequence, i.e. $\mathbf{s_{cw}} = \{(c_1, ws_1), (c_2, ws_2), ..., (c_n, ws_n)\}$, where c_i denotes the i-th character in the sentence and ws_i denotes matched words assigned to c_i.

3.2 Lexicon Adapter

Each position in the sentence consists of two types of information, namely character-level and word-level features. In line with the existing hybrid models, our goal is to combine the lexicon feature with BERT. Specifically, inspired by the recent works about BERT adapter (Houlsby et al., 2019; Wang et al., 2020), we propose a novel Lexicon Adapter (LA) shown in Figure 4, which can directly inject lexicon information into BERT.

A Lexicon Adapter receives two inputs, a character and the paired words. For the i-th position in a char-words pair sequence, the input is denoted as (h_i^c, x_i^{ws}), where h_i^c is a character vector, the output of a certain transformer layer in BERT, and $x_i^{ws} = \{x_{i1}^w, x_{i2}^w, ..., x_{im}^w\}$ is a set of word embeddings. The j-th word in x_i^{ws} is represented as following:

$$x_{ij}^w = \mathbf{e}^w(w_{ij}) \tag{1}$$

where $\mathbf{e}^w$ is a pre-trained word embedding lookup table and w_{ij} is the j-th word in ws_i.

To align those two different representations, we apply a non-linear transformation for the word vectors:

$$v_{ij}^w = \mathbf{W}_2(\tanh(\mathbf{W}_1 x_{ij}^w + \mathbf{b}_1)) + \mathbf{b}_2 \tag{2}$$

where $\mathbf{W}_1$ is a d_c-by-d_w matrix, $\mathbf{W}_2$ is a d_c-by-d_c matrix, and $\mathbf{b}_1$ and $\mathbf{b}_2$ are scaler bias. d_w and d_c denote the dimension of word embedding and the hidden size of BERT respectively.

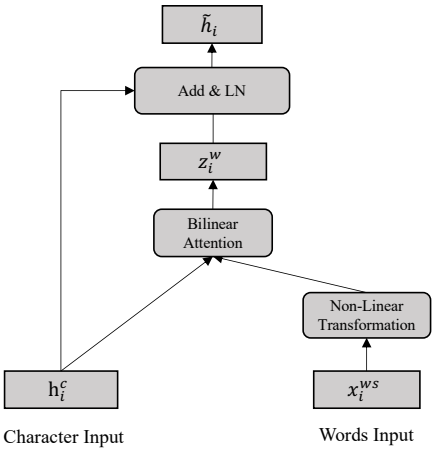


Figure 4: Structure of Lexicon Adapter (LA). The adapter takes as input a character vector and the paired word features. Subsequently, a bilinear attention over both character and words is used to weighted the lexicon feature into a vector, which is then added to the input character-level vector and followed by a layer normalization.

As Figure 3 shows, each character is paired with multiple words. However, the contribution to each task varies from word to word. For example, as for Chinese POS tagging, words "美国 (America)" and "人民 (People)" are superior to "美国人 (American)" and "国人 (Compatriot)", since they are ground-truth segmentation of the sentence. To pick out the most relevant words from all matched words, we introduce a character-to-word attention mechanism.

Specifically, we denote all v_{ij}^w assigned to i-th character as $V_i = (v_{i1}^w, ..., v_{im}^w)$, which has the size m-by-d_c and m is the total number of the assigned word. The relevance of each word can be calculated as:

$$\mathbf{a}_i = \text{softmax}(h_i^c \mathbf{W}_{attn} V_i^T) \tag{3}$$

where $\mathbf{W}_{attn}$ is the weight matrix of bilinear attention. Consequently, we can get the weighted sum of all words by:

$$z_i^w = \sum_{j=1}^{m} a_{ij} v_{ij}^w \tag{4}$$

Finally, the weighted lexicon information is injected into the character vector by:

$$\tilde{h}_i = h_i^c + z_i^w \tag{5}$$

It is followed by a dropout layer and layer normalization.

3.3 Lexicon Enhanced BERT

Lexicon Enhanced BERT (LEBERT) is a combination of Lexicon Adapter (LA) and BERT, in which LA is applied to a certain layer of BERT shown in Figure 2. Concretely, LA is attached between certain transformers within BERT, thereby injecting external lexicon knowledge into BERT.

Given a Chinese sentence with n characters $\mathbf{s}_c = \{c_1, c_2, ..., c_n\}$, we build the corresponding character-words pair sequence $\mathbf{s}_{cw} = \{(c_1, ws_1), (c_2, ws_2), ..., (c_n, ws_n)\}$ as described in Section 3.1. The characters $\{c_1, c_2, ..., c_n\}$ are first input into Input Embedder which outputs $E = \{e_1, e_2, ..., e_n\}$ by adding token, segment and position embedding. Then we input E into Transformer encoders and each Transformer layer acts as following:

$$\begin{aligned} G &= \text{LN}(H^{l-1} + \text{MHAttn}(H^{l-1})) \\ H^l &= \text{LN}(G + \text{FFN}(G)) \end{aligned} \tag{6}$$

where $H^l = \{h_1^l, h_2^l, ..., h_n^l\}$ denotes the output of the l-th layer and $H^0 = E$; LN is layer normalization; MHAttn is the multi-head attention mechanism; FFN is a two-layer feed-forward network with ReLU as hidden activation function.

To inject the lexicon information between the k-th and $(k+1)$-th Transformer, we first get the output $H^k = \{h_1^k, h_2^k, ..., h_n^k\}$ after k successive Transformer layers. Then, each pair (h_i^k, x_i^{ws}) are passed through the **Lexicon Adapter** which transforms the i_{th} pair into $\tilde{h}_i^k$:

$$\tilde{h}_i^k = \text{LA}(h_i^k, x_i^{ws}) \tag{7}$$

Since there are $L = 12$ Transformer layers in the BERT, we input $\widetilde{H}^k = \{\tilde{h}_1^k, \tilde{h}_2^k, ..., \tilde{h}_n^k\}$ to the remaining $(L-k)$ Transformers. At the end, we get the output of L-th Transformer H^L for the sequence labeling task.

3.4 Training and Decoding

Considering the dependency between successive labels, we use a CRF layer to make sequence labeling. Given the hidden outputs of the last layer $H^L = \{h_1^L, h_2^L, ..., h_n^L\}$, we first calculate scores P as:

$$O = \mathbf{W}_o H^L + \mathbf{b}_o \tag{8}$$

For a label sequence $\mathbf{y} = \{y_1, y_2, ..., y_n\}$, we define its probability to be:

$$p(\mathbf{y}|\mathbf{s}) = \frac{\exp(\sum_i (O_{i,y_i} + T_{y_{i-1},y_i}))}{\sum_{\tilde{\mathbf{y}}} \exp(\sum_i (O_{i,\tilde{y}_i} + T_{\tilde{y}_{i-1},\tilde{y}_i}))} \tag{9}$$

Dataset		Type	Train	Dev	Test
NER	Weibo	Sent	1.4k	0.27k	0.27k
		Char	73.8k	14.5k	14.8k
	Ontonotes	Sent	15.7k	4.3k	4.3k
		Char	491.9k	200.5k	208.1k
	MSRA	Sent	46.4k	-	4.4k
		Char	2169.9k	-	172.6k
	Resume	Sent	3.8k	0.46k	0.48k
		Char	124.1k	13.9k	15.1k
CWS	PKU	Sent	19.1k	-	1.9k
		Char	1826k	-	173k
	MSR	Sent	86.9k	-	4.0k
		Char	4050k	-	184k
	CTB6	Sent	23k	2k	3k
		Char	1056k	100k	134k
POS	CTB5	Sent	18k	350	348
		Char	805k	12k	14k
	CTB6	Sent	23k	2k	3k
		Char	1056k	100k	134k
	UD	Sent	4k	500	500
		Char	156k	20k	19k

Table 1: The statistics of the datasets.

where T is the transition score matrix and $\tilde{\mathbf{y}}$ denotes all possible tag sequences.

Given N labelled data $\{\mathbf{s}_j, \mathbf{y}_j\}|_{j=1}^N$, we train the model by minimize the sentence-level negative log-likelihood loss as:

$$\mathcal{L} = -\sum_j \log(p(\mathbf{y}|\mathbf{s})) \tag{10}$$

While decoding, we find out the label sequence obtaining the highest score using the Viterbi algorithm.

4 Experiments

We carry out an extensive set of experiments to investigate the effectiveness of LEBERT. In addition, we aim to empirically compare model-level and BERT-level fusion in the same setting. Standard F1-score (F1) is used as evaluation metrics.

4.1 Datasets

We evaluate our method on ten datasets of three different sequence labeling tasks, including Chinese NER, Chinese Word Segmentation, and Chinese POS tagging. The statistics of the datasets is shown in Table 1.

Chinese NER. We conduct experiments on four benchmark datasets, including Weibo NER (Peng and Dredze, 2015, 2016), OntoNotes (Weischedel et al., 2011), Resume NER (Zhang and Yang, 2018), and MSRA (Levow, 2006). Weibo NER is a social media domain dataset, which is drawn

from Sina Weibo; while OntoNotes and MSRA datasets are in the news domain. Resume NER dataset consists of resumes of senior executives, which is annotated by Zhang and Yang (2018).

Chinese Word Segmentation. For Chinese word segmentation, we employ three benchmark datasets in our experiments, namely PKU, MSR, and CTB6, where the former two are from SIGHAN 2005 Bakeoff (Emerson, 2005) and the last one is from Xue et al. (2005). For MSR and PKU, we follow their official training/test data split. For CTB6, we use the same split as that stated in Yang and Xue (2012); Higashiyama et al. (2019).

Chinese POS Tagging. For POS-tagging, three Chinese benchmark datasets are used, including CTB5 and CTB6 from the Penn Chinese TreeBank (Xue et al., 2005) and the Chinese GSD Treebank of Universal Dependencies(UD) (Nivre et al., 2016). The CTB datasets are in simplified Chinese while the UD dataset is in traditional Chinese. Following Shao et al. (2017), we first convert the UD dataset into simplified Chinese before the POS-tagging experiments[3]. Besides, UD has both universal and language-specific POS tags, we follow previous works (Shao et al., 2017; Tian et al., 2020a), referring to the corpus with two tagsets as UD1 and UD2, respectively. We use the official splits of train/dev/test in our experiments.

4.2 Experimental Settings

Our model is constructed based on BERT_{BASE} (Devlin et al., 2019), with 12 layers of transformer, and is initialized using the Chinese-BERT checkpoint from huggingface[4]. We use the 200-dimension pre-trained word embedding from Song et al. (2018), which is trained on texts of news and webpages using a directional skip-gram model. The lexicon $\mathbf{D}$ used in this paper is the vocab of the pre-trained word embedding. We apply the Lexicon Adapter between the 1-st and 2-nd Transformer in BERT and fine-tune both BERT and pre-trained word embedding during training.

Hyperparameters. We use the Adam optimizer with an initial learning rate of 1e-5 for original parameters of BERT, and 1e-4 for other parameters introduced by LEBERT, and a maximum epoch number of 20 for training on all datasets. The max length of the sequence is set to 256, and the training batch size is 20 for MSRA NER and 4 for other datasets.

[3]The conversion tool we used is OpenCC.

[4]https://github.com/huggingface/transformers

Model	Weibo	Ontonotes	MSRA	Resume
Zhang and Yang (2018)*	63.34	75.49	92.84	94.51
Zhu and Wang (2019)	59.31	73.64	92.97	94.94
Liu et al. (2019)*	65.30	75.79	93.50	94.49
Ding et al. (2019)	59.50	75.20	94.40	-
Ma et al. (2020)* †	69.11	81.34	95.35	95.54
Li et al. (2020)* †	68.07	80.56	95.46	95.78
BERT	67.27	79.93	94.71	95.33
BERT+Word	68.32	81.03	95.32	95.46
ERINE	67.96	77.65	95.08	94.82
ZEN	66.71	79.03	95.20	95.40
LEBERT	**70.75**	**82.08**	**95.70**	**96.08**

Table 2: Results on Chinese NER.

Baselines. To evaluate the effectiveness of the proposed LEBERT, we compare it with the following approaches in the experiments.

- **BERT**. Directly fine-tuning a pre-trained Chinese BERT on Chinese sequence labeling tasks.
- **BERT+Word**. A strong model-level fusion baseline method, which inputs the concatenation of BERT vector and bilinear attention weighted word vector, and uses LSTM[5] and CRF as fusion layer and inference layer respectively.
- **ERNIE** (Sun et al., 2019a). An extension of BERT using a entity-level mask to guide pre-training.
- **ZEN**. Diao et al. (2020) explicitly integrate N-gram information into BERT through an extra multi-layers of N-gram Transformer encoder and pre-training.

Further, we also compare with the state-of-the-art models of each task.

4.3 Overall Results

Chinese NER. Table 2 shows the experimental results on Chinese NER datasets[6]. The first four rows (Zhang and Yang, 2018; Zhu and Wang, 2019; Liu et al., 2019; Ding et al., 2019) in the first block show the performance of lexicon enhanced character-based Chinese NER models, and the last two rows (Ma et al., 2020; Li et al., 2020)

[5]We also evaluated with other fusion layers, such as Transformer, but we found LSTM is consistently better.

[6]For a fair comparison, in Table 2, we use * denotes training the model with the same pre-trained word embedding as ours; † means the model is also initialized using the Chinese BERT checkpoint from huggingface and evaluated using the seqeval tool.

Model	PKU	MSR	CTB6
Yang et al. (2017)	95.00	96.80	95.40
Ma et al. (2018)	96.10	97.40	96.70
Yang et al. (2019)	95.80	97.80	96.10
Qiu et al. (2020)	96.41	98.05	96.99
Tian et al. (2020c)(with BERT)	96.51	98.28	97.16
Tian et al. (2020c)(with ZEN)	96.53	98.40	97.25
BERT	96.25	97.94	96.98
BERT+Word	96.55	98.41	97.25
ERINE	96.33	98.17	97.02
ZEN	96.36	98.36	97.13
LEBERT	**96.91**	**98.69**	**97.52**

Table 3: Results on Chinese Word Segmentation.

Model	CTB5	CTB6	UD1	UD2
Shao et al. (2017)	94.38	-	89.75	89.42
Zhang et al. (2018)	94.95	92.51	-	-
Tian et al. (2020a)(BERT)	96.77	94.82	95.51	95.46
Tian et al. (2020a)(ZEN)	96.86	94.87	95.52	95.49
Tian et al. (2020b)(BERT)	96.60	94.74	95.50	95.38
Tian et al. (2020b)(ZEN)	96.82	94.82	95.59	95.41
BERT	96.25	94.64	94.83	94.73
BERT+Word	96.77	94.75	95.39	95.41
ERINE	96.51	94.76	95.10	95.14
ZEN	96.60	94.70	95.15	95.05
LEBERT	**97.14**	**95.18**	**96.06**	**95.74**

Table 4: Results on Chinese POS Tagging.

		BERT	STOA (with BERT)
NER	Weibo	10.63%	5.31%
	Ontonote4	10.71%	3.97%
	MSRA	18.71%	5.28%
	Resume	16.06%	7.11%
CWS	PKU	17.60%	11.46%
	MSR	36.41%	23.84%
	CTB6	17.88%	12.68%
POS	CTB5	23.73%	11.46%
	CTB6	10.07%	6.95%
	UD1	23.79%	12.25%
	UD2	19.17%	6.17%

Table 5: The relative error reductions over different base models.

in the same block are the state-of-the-art models using shallow fusion layer to integrate lexicon information and BERT. The hybrid models, including existing state-of-the-art models, BERT + Word, and the proposed LEBERT, achieve better performance than both lexicon enhanced models and BERT baseline. This demonstrates the effectiveness of combining BERT and lexicon features for Chinese NER. Compared with model-level fusion models ((Ma et al., 2020; Li et al., 2020), and BERT+Word), our BERT-level fusion model, LEBERT, improves in F1 score on all four datasets across different domains, which shows that our approach is more efficient in integrating word and BERT. The results also indicate that our adapter-based method, LEBERT, with an extra pre-trained word embedding solely, outperforms those two lexicon-guided pre-training models (ERNIE and ZEN). This is likely because implicit integration of lexicon in ERNIE and restricted pre-defined n-gram vocabulary size in ZEN limited the effect.

Chinese Word Segmentation. We report the F1 score of our model and the baseline methods on Chinese Word Segmentation in Table 3. Yang et al. (2019) applied a lattice LSTM to integrate word feature to character-based CWS model. Qiu et al. (2020) investigated the benefit of multiple heterogeneous segmentation criteria for single criterion Chinese word segmentation. Tian et al. (2020c) designed a wordhood memory network to incorporate wordhood information into a pre-trained-based CWS model and showed good performance. Compared with those approaches, the models (BERT+Word and LEBERT) that combine lexicon features and BERT perform better. Moreover, our proposed LEBERT outperforms both model-level fusion baseline (BERT+Word) and lexicon-guided pre-training models (ERNIE and ZEN), achieving the best results.

Chinese POS Tagging. We report the F1 score on four benchmarks of Chinese POS tagging in Table 4. The state-of-the-art model (Tian et al., 2020a) jointly trains Chinese Word Segmentation and Chinese POS tagging using a two-way attention to incorporate auto-analyzed knowledge, such as POS labels, syntactic constituents, and dependency relations. Similar to BERT+Word baseline, Tian et al. (2020b) integrated character-Ngram features with BERT at model-level using a multi-channel attention. As shown in Table 4, hybrid models ((Tian et al., 2020b), BERT+Word, LEBERT) that combine words information and BERT outperform BERT baseline, indicating that lexicon features can further improve the performance of BERT. LEBERT achieves the best results among these approaches, which demonstrates the effectiveness of BERT-level fusion. Consistent with results on Chinese NER and CWS, our BERT adapter-based approach is superior to lexicon-guided pre-training methods (ERNIE and ZEN).

Our proposed model has achieved state-of-the-art results across all datasets. To better show the strength of our method, we also summarize the relative error reduction over BERT baseline

	Span F1		Type Acc	
	Ontonotes	UD1	Ontonotes	UD1
BERT	82.68	97.99	97.16	96.99
BERT+Word	83.38	98.09	97.24	97.51
LEBERT	84.16	98.47	97.84	97.72

Table 6: Span F1 and Type Acc of different models.

and BERT-based state-of-the-art models in Table 5. The results show that the relative error reductions are significant compared with baseline models.

4.4 Model-level Fusion vs. BERT-level Fusion

Compared with model-level fusion models, LEBERT directly integrates lexicon features into BERT. We evaluate those two types of models in terms of Span F1, Type Acc, and Sentence Length, choosing the BERT+Word as the model-level fusion baseline due to its good performance across all the datasets. We also compare with a BERT baseline since both LEBERT and BERT+Word are improved based on it.

Span F1 & **Type Acc**. Span F1 means the correctness of the span for an Entity in NER or a word in POS-tagging, while Type Acc denotes the proportion of full-correct predictions to span-correct predictions. Table 6 shows the results of three models on the Ontonotes and UD1 datasets. We can find that both BERT+Word and LEBERT perform better than BERT in terms of Span F1 and Type Acc on the two datasets. The results indicate that lexicon information contributes to span boundary detection and span classification. Specifically, the improvement of Span F1 is larger than Type Acc on Ontonotes, but smaller on UD1. Compared with BERT+Word, LEBERT achieves more improvement, demonstrating the effectiveness of lexicon feature enhanced via BERT-level fusion.

Sentence Length. Figure 5 shows the F1-value trend of the baselines and LEBERT on Ontonotes dataset. All the models show a similar performance-length curve, decreasing as the sentence length increase. We speculate that long sentences are more challenging due to complicated semantics. Even lexicon enhanced models may fail to choose the correct words because of the increased number of matched words as the sentence become longer. The F1-score of BERT is relatively low, while BERT+Word achieves better performance due to the usage of lexicon information. Compared with BERT+Word, LEBERT performs better and shows more robustness when sentence length increases, demonstrating the more effective use of lexicon information.

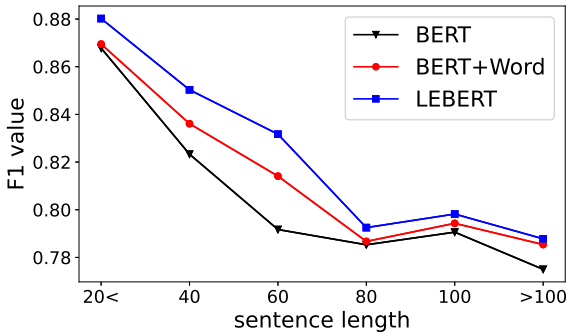


Figure 5: F1-value against the sentence length.

Layer	***one***				
	1	3	6	9	12
F1	82.08	81.43	81.24	81.10	80.64
Layer	***multi***			***all***	
	1,3	1,3,6	1,3,6,9	all	
F1	81.54	81.28	81.23	78.54	

Table 7: Results of variations of LEBERT with Lexicon Adapter applied at different layers of BERT model. ***one***, ***multi***, ***all*** mean applying LA after one layer, multiply layers, all layers of Transformer in BERT.

Case Study. Table 8 shows examples of Chinese NER and Chinese POS tagging results on Ontonotes and UD1 datasets respectively. In the first example, BERT can not determine the entity boundary, but BERT+Word and LEBERT can segment it correctly. However, the BERT+Word model fails to predict the type of the entity "呼伦贝尔盟 (Hulunbuir League)" while LEBERT makes the correct prediction. This is likely because fusion at the lower layer contributes to capturing more complex semantics provided by BERT and lexicon. In the second example, the three models can find the correct span boundary, but both BERT and BERT+Word make incorrect predictions of the span type. Although BERT+Word can use the word information, it is disturbed by the irrelevant word "七八 (Seven and Eight)" predicting it as NUM. In contrast, LEBERT can not only integrate lexicon features but also choose the correct word for prediction.

4.5 Discussion

Adaptation at Different Layers. We explore the effect of applying the Lexicon Adapter (LA) between different Transformer layers of BERT on Ontonotes dataset. Different settings are evalu-

#1 Example of Chinese NER								
Sentence (truncated)	内蒙古呼伦贝尔盟 (Hulunbuir League, Inner Mongolia)							
Matched Words	内蒙, 内蒙古, 内蒙古呼伦贝尔, 蒙古, 呼伦, 呼伦贝尔, 呼伦贝尔盟, 贝尔 Inner Mongolia, Inner Mongolia, Inner Mongolia Hulunbuir, Mongolia, Hulun, Hulunbuir, Hulunbuir League, Buir							
Characters	内	蒙	古	呼	伦	贝	尔	盟
Gold Labels	B-GPE	I-GPE	E-GPE	B-GPE	I-GPE	I-GPE	I-GPE	E-GPE
BERT	B-GPE	I-GPE	I-GPE	I-GPE	I-GPE	I-GPE	I-GPE	E-GPE
BERT+Word	B-GPE	I-GPE	E-GPE	B-ORG	I-ORG	I-ORG	I-ORG	E-ORG
LEBERT	B-GPE	I-GPE	E-GPE	B-GPE	I-GPE	I-GPE	I-GPE	E-GPE
#2 Example of Chinese POS Tagging								
Sentence (truncated)	乱七八糟的关系 (Messy Relationship)							
Matched Words	乱七八糟, 七八, 八糟, 关系 Mess, Seven and Eight, Bad News, Relationship							
Characters	乱	七	八	糟	的	关	系	
Gold Labels	B-ADJ	I-ADJ	I-ADJ	E-ADJ	S-PART	B-NOUN	E-NOUN	
BERT	B-ADJ	I-NUM	I-NUM	E-ADJ	S-PART	B-NOUN	E-NOUN	
BERT+Word	B-ADJ	I-NUM	I-NUM	E-ADJ	S-PART	B-NOUN	E-NOUN	
LEBERT	B-ADJ	I-ADJ	I-ADJ	E-ADJ	S-PART	B-NOUN	E-NOUN	

Table 8: Examples of tagging result.

ated, including applying LA after *one*, *multiple*, and *all* layers of Transformer. As for *one* layer, we applied LA after $k \in \{1, 3, 6, 9, 12\}$ layer; and $\{1, 3\}$, $\{1, 3, 6\}$, $\{1, 3, 6, 9\}$ layers for *multiple* layers. *All* layers represents LA used after every Transformer layer in BERT. The results show in Table 7. The shallow layer achieves better performance, which can be due to the fact that the shallow layer promotes more layered interaction between lexicon features and BERT. Applying LA at multi-layers of BERT hurts the performance and one possible reason is that integration at multi-layers causes over-fitting.

Tuning BERT or Not. Intuitively, integrating lexicon into BERT without fine-tuning can be faster (Houlsby et al., 2019) but with lower performance due to the different characteristics of lexicon feature and BERT (discrete representation vs. neural representation). To evaluate its impact, we conduct experiments with and without fine-tuning BERT parameters on Ontonotes and UD1 datasets. From the results, we find that without fine-tuning the BERT, the F1-score shows a decline of 7.03 points (82.08 $\rightarrow$ 75.05) on Ontonotes and 3.75 points (96.06 $\rightarrow$ 92.31) on UD1, illustrating the importance of fine-tuning BERT for our lexicon integration.

5 Conclusion

In this paper, we proposed a novel method to integrate lexicon features and BERT for Chinese sequence labeling, which directly injects lexicon information between Transformer layers in BERT using a Lexicon Adapter. Compared with model-level fusion methods, LEBERT allows in-depth fusion of lexicon features and BERT representation at BERT-level. Extensive experiments show that the proposed LEBERT achieves state-of-the-art performance on ten datasets of three Chinese sequence labeling tasks.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and suggestions. Moreover, We sincerely thank Dr. Zhiyang Teng for his constructive collaboration during the development of this paper, and Dr. Haixia Chai, Dr. Jie Yang, and my colleague Junfeng Tian for their help in polishing our paper.

References

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. 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 1538–1548, Hong Kong, China. Association for Computational Linguistics.

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

Jason Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4(0):357–370.

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.

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

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

L. Gan and Y. Zhang. 2020. Investigating self-attention network for chinese word segmentation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2933–2941.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *CoRR*, abs/1901.05287.

Tao Gui, Ruotian Ma, Qi Zhang, Lujun Zhao, Yu-Gang Jiang, and Xuanjing Huang. 2019a. Cnn-based chinese ner with lexicon rethinking. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4982–4988. International Joint Conferences on Artificial Intelligence Organization.

Tao Gui, Qi Zhang, Haoran Huang, Minlong Peng, and Xuanjing Huang. 2017. Part-of-speech tagging for Twitter with adversarial neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2411–2420, Copenhagen, Denmark. Association for Computational Linguistics.

Tao Gui, Yicheng Zou, Qi Zhang, Minlong Peng, Jinlan Fu, Zhongyu Wei, and Xuanjing Huang. 2019b. A lexicon-based graph neural network for Chinese NER. 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 1040–1050, Hong Kong, China. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Shohei Higashiyama, Masao Utiyama, Eiichiro Sumita, Masao Ideuchi, Yoshiaki Oida, Yohei Sakamoto, and Isaac Okada. 2019. Incorporating word attention into character-based word segmentation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2699–2709, Minneapolis, Minnesota. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*.

Yuting Hu and Suzan Verberne. 2020. Named entity recognition for Chinese biomedical patents. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 627–637, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Chen Jia, Yuefeng Shi, Qinrong Yang, and Yue Zhang. 2020. Entity enhanced BERT pre-training for Chinese NER. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6384–6396, Online. Association for Computational Linguistics.

Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., USA.

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

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

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020. FLAT: Chinese NER using flat-lattice transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6836–6842, Online. Association for Computational Linguistics.

Wei Liu, Tongge Xu, Qinghua Xu, Jiayu Song, and Yueran Zu. 2019. An encoding strategy based word-character LSTM for Chinese NER. 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 2379–2389, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhangxun Liu, Conghui Zhu, and Tiejun Zhao. 2010. *Chinese Named Entity Recognition with a Sequence Labeling Approach: Based on Characters, or Based on Words?* Springer Berlin Heidelberg.

Ji Ma, Kuzman Ganchev, and David Weiss. 2018. State-of-the-art Chinese word segmentation with Bi-LSTMs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4902–4908, Brussels, Belgium. Association for Computational Linguistics.

Ruotian Ma, Minlong Peng, Qi Zhang, Zhongyu Wei, and Xuanjing Huang. 2020. Simplify the usage of lexicon in Chinese NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5951–5960, Online. Association for Computational Linguistics.

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

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*, volume 32, pages 2746–2757. Curran Associates, Inc.

Hwee Tou Ng and Jin Kiat Low. 2004. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based? In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 277–284, Barcelona, Spain. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

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

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

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer.

Xipeng Qiu, Hengzhi Pei, Hang Yan, and Xuanjing Huang. 2020. A concise model for multi-criteria Chinese word segmentation with transformer encoder. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2887–2897, Online. Association for Computational Linguistics.

Yan Shao, Christian Hardmeier, Jörg Tiedemann, and Joakim Nivre. 2017. Character-based joint segmentation and POS tagging for Chinese using bidirectional RNN-CRF. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 173–183, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Mo Shen, Wingmui Li, HyunJeong Choe, Chenhui Chu, Daisuke Kawahara, and Sadao Kurohashi. 2016. Consistent word segmentation, part-of-speech tagging and dependency labelling annotation for Chinese language. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 298–308, Osaka, Japan. The COLING 2016 Organizing Committee.

Yan Song, Shuming Shi, Jing Li, and Haisong Zhang. 2018. Directional skip-gram: Explicitly distinguishing left and right context for word embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 175–180, New Orleans, Louisiana. Association for Computational Linguistics.

Weiwei Sun and Hans Uszkoreit. 2012. Capturing paradigmatic and syntagmatic lexical relations: Towards accurate Chinese part-of-speech tagging. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 242–252, Jeju Island, Korea. Association for Computational Linguistics.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019a. 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. 2019b. Ernie 2.0: A continual pre-training framework for language understanding. *arXiv preprint arXiv:1907.12412*.

Yuanhe Tian, Yan Song, Xiang Ao, Fei Xia, Xiaojun Quan, Tong Zhang, and Yonggang Wang. 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*, pages 8286–8296, Online. Association for Computational Linguistics.

Yuanhe Tian, Yan Song, and Fei Xia. 2020b. 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*, pages 2073–2084, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2020. K-adapter: Infusing knowledge into pre-trained models with adapters.

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

Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. 2005. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering*, 11(2):207.

Haiqin Yang. 2019. BERT meets chinese word segmentation. *CoRR*, abs/1909.09292.

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

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

Jie Yang, Yue Zhang, and Shuailong Liang. 2019. Subword encoding in lattice LSTM for Chinese word segmentation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2720–2725, Minneapolis, Minnesota. Association for Computational Linguistics.

Yaqin Yang and Nianwen Xue. 2012. Chinese comma disambiguation for discourse analysis. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 786–794, Jeju Island, Korea. Association for Computational Linguistics.

M. Zhang, N. Yu, and G. Fu. 2018. A simple and effective neural model for joint word segmentation and pos tagging. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(9):1528–1538.

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.

Xiaoyan Zhao, Min Yang, Qiang Qu, and Yang Sun. 2020. *Improving Neural Chinese Word Segmentation with Lexicon-Enhanced Adaptive Attention*, page 1953–1956. Association for Computing Machinery, New York, NY, USA.

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