LEX-BERT: ENHANCING BERT BASED NER WITH LEXICONS

Wei Zhu
Department of Computer Science
East China Normal University
Shanghai, China
52205901018@stu.ecnu.edu.cn

Daniel Cheung
AI4ALL
San Diego, California, US
albert_knows@163.com

ABSTRACT

In this work, we represent Lex-BERT, which incorporates the lexicon information into Chinese BERT for named entity recognition (NER) tasks in a natural manner. Instead of using word embeddings and a newly designed transformer layer as in FLAT, we identify the boundary of words in the sentences using special tokens, and the modified sentence will be encoded directly by BERT. Our model does not introduce any new parameters and are more efficient than FLAT. In addition, we do not require any word embeddings accompanying the lexicon collection. Experiments on MSRA and ZhCrossNER show that our model outperforms FLAT and other baselines.

1 INTRODUCTION

The pretrained language models (PLMs) including BERT Devlin et al. (2018) and its variants Yang et al. (2019); Liu et al. (2019) have been proven beneficial for many natural language processing (NLP) tasks, such as text classification, question answering Rajpurkar et al. (2018) and natural language inference (NLI) Bowman et al. (2015), on English, Chinese and many other languages. Despite its impressive performances in NER tasks, recently it has been proven that adding lexical information can improve the downstream performance by an significant margin (Li et al. (2020)). However, there is no work that incorporate word information in BERT without introducing additional structure. In our work, we propose Lexical BERT (Lex-BERT), a more convenient way of leveraging lexicons in BERT based NER models.

Our methods have three advantages over FLAT. First, we do not require the word information come with a word embedding model. Second, FLAT can not incorporate word type information. However, in many in-domain scenarios, the words collected have entity type information, which can be incorporated by our method. Third, our method is more efficient in terms of memory consumption and inference speed.

We conduct experiments on two NER benchmark datasets.The experiments show that our models can provide performance gains with less resource consumption.

2 METHOD

Now we elaborate on our methods. We provide two versions of Lex-BERT.

2.1 LEX-BERT V1

The first version of Lex-BERT identifies the spans of words in a sentence by inserting special tokens on the left and right of the words. The special tokens can not only mark the start and end positions of the word, but also provide entity type information into the sentence. As is shown in the Figure 1(a), we identify 口服(take orally) and 降(decrease) as medicine related verbs, 他汀(statins), 血脂(blood lipids), 药物(drug) as a drug, a medical examination index, and a medical concept term in the given

(a) Lex-BERT V1

(b) Lex-BERT V2

Figure 1: Two versions of Lex-BERT.

sentence 一般都是通过口服他汀类降血脂的药物 (Usually statins are taken orally to lower blood lipids).[1]

The inserted markers carry two types of information. First, if our dictionary comes with entity type information, we can insert it into sentence via markers. Here *v* is short for medicine related verbs, *d* for drug, *i* for examination index, *c* for medical concept term. Second, they indicate the start or end of a word in our lexical collections. Here the starting marker is in the format of [*x*] and the ending marker is like [/*x*], where x can be *v*, *d*, *i*, etc.

2.2 LEX-BERT V2

Now we present the second version of Lex-BERT (as in Figure 1(b)). In Lex-BERT V2, for a word w identified in the sentence, instead of inserting starting and ending markers at the surroundings of the word in the sentence, we append a single marker $[x]$ at the end of the sentence. Note that we tie the position embeddings of the marker with the starting token of the word:

$$P(w_{start}) = P([x]), \tag{1}$$

where $P(\cdot)$ denotes the position index of a token, and w_{start} is the first token of the word w.

In addition, we modify the attention matrix of BERT, which is depicted in Figure 2. The text tokens in the sentence will only attend to each other and not attend to the markers. In contrast, the marker tokens can attend to all the tokens in the input sequence.

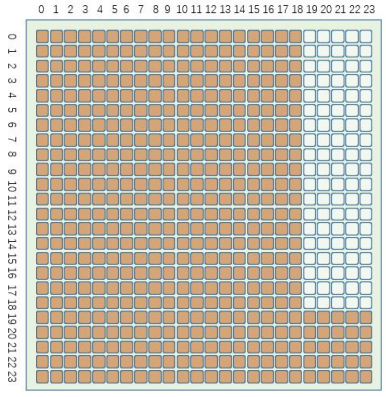

Figure 2: This figure depicts how we modify the attention mechanism of the original fully connected one in BERT.

[1]Note that "他汀类降血脂的药物" may not be in our dictionary, so we don't insert additional markers to its surroundings.

Dataset	Domain	Language	Train #	Dev #	Test #	Metrics
Ontonotes 4.0	multi-domain	zh	15.7k	4.3k	4.3	micro F1
ZhCrossNER	multi-domain	en	22k	5k	5k	macro F1

Table 1: Overview of used datasets in experiments.

model	Ontonotes	ZhCrossNER
BERT	80.14	69.74
FLAT	81.82	70.36
Lex-BERT V1	81.86	70.63
Lex-BERT V2	82.15	71.27

Table 2: Main results on the Chinese NER benchmark datasets. For each task and each model, experiments are repeated for 10 times, and the average and standard deviation of the scores are reported.

3 EXPERIMENTS

3.1 SETUP

In this article, all models use the Chinese BERT-wwm-ext Cui et al. (2020) as encoder. During the experiments, we mainly focus on the following hyper-parameters: batch size, learning rate, warm-up steps. Other BERT configurations remain the same with Cui et al. (2020). The optimizer is the AdamW Loshchilov & Hutter (2017), which is a modified version of Kingma & Ba (2015). Each model is run on a given task for 10 times and the average performance scores and standard deviations are reported for reproducibility. Each task is assigned 4 CPU cores and 1 Tesla V100 GPU card.

3.2 BENCHMARK TASKS

For downstream tasks, we investigate 2 NER tasks: (1) Chinese Ontonotes 4.0[2] (Ontonotes); (2) ZhCrossNER (Zhu (2020)), which consists of labeled sentences from multiple domains. Table 1. shows the statistics of the three datasets.

3.3 EXPERIMENTAL RESULTS

Table 3 reports the main experimental results. FLAT improves upon the BERT by adding lexicon information. Our Lex-BERT V1 provide slight improvements over FLAT. Our Lex-BERT V2 performs better, with 0.33% improvement on Ontonotes and 0.91% improvement on ZhCrossNER. The results show that our Lex-BERT are effective.

3.4 ANALYSIS OF EFFICIENCY

To demonstrate that our model is efficient, we compare the inference speed and GPU memory consumption of the three models compared in Table 3. The batch size is 16. In a batch, the sentences are padded to the max length in that batch, which should not exceed 512. Table show the relative memory size and relative inference speed. We can see that Lex-BERT are faster than FLAT and requires less speed

[2]https://catalog.ldc.upenn.edu/LDC2013T19

model	Memory consumption	Inference speed
BERT	1.0	1.0
FLAT	1.25	0.82
Lex-BERT V1	1.21	0.85
Lex-BERT V2	1.14	0.89

Table 3: The relative memory size and relative inference speed of the three models.

4 CONCLUSIONS

In this work, we propose two versions of Lex-BERT, which incorporate lexical information in BERT models without additional architecture designs or parameters. Experiments show that our model outperforms BERT based FLAT and are more efficient.

REFERENCES

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. *arXiv e-prints*, art. arXiv:1508.05326, August 2015.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 657–668, Online, November 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.findings-emnlp.58.

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

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICML*, 2015.

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. FLAT: Chinese NER using flat-lattice transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6836–6842, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.611. URL https://www.aclweb.org/anthology/2020.acl-main.611.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv e-prints*, art. arXiv:1907.11692, July 2019.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. *arXiv e-prints*, art. arXiv:1711.05101, November 2017.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know What You Don't Know: Unanswerable Questions for SQuAD. *arXiv e-prints*, art. arXiv:1806.03822, June 2018.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv e-prints*, art. arXiv:1906.08237, June 2019.

Wei Zhu. A chinese cross-domain named entity recognition dataset. *ArXiv*, abs/2012, 2020.