An Improved Mask Approach Based on Pointer Network for Domain Adaptation of BERT

An Improved Mask Approach Based on Pointer Network for Domain Adaptation of BERT

Pengkai Lu[1], Dawei Jiang[1] and Ying Li[2]

[1]College of Computer Science and Technology, Zhejiang University, Hangzhou, CN
[2]Netease Network Co., LTD., Hangzhou, CN
Email: lupengkai@zju.edu.cn

Abstract. Pre-trained BERT model has shown its amazing strength on downstream NLP tasks by fine-tuning. However, the results with fine-tuned BERT will decrease when the model is directly applied to a series of domain-specific tasks. he original fine-tuning method does not consider accurate semantics of tokens in a specific domain. Different from random selecting, we present a more efficient mask method which utilizes a pointer network to decide which tokens should be preferentially masked. The pointer network sorts tokens in a sentence by their recovery difficulty. Then we train a BERT model to predict top tokens that are replaced by [mask] in original sentences. We tested the new training approach on biomedical corpora. Experiments show that the new trained model outperforms the original BERT model in some domain-specific NLP tasks while consuming extra domain corpus.

1. Introduction

Recently, deep contextual language model is showed effective to learn universal language representations, and achieves state-of-the-art results in a series of Nature Language Processing tasks. Leveraging the advantage of unsupervised pre-training has become especially important when annotations are difficult to obtain. BERT [1] is one of the recent language models that can be viewed as denoising autoencoders. One main task of training BERT model is to mask a small subset of the input sequence randomly and train a neural network to recover the masked tokens. Hence it can be used for learning bidirectional presentations.

Fine-tuning deep neural networks [1] is become popular as a supervised approach for domain adaptation in which a base network has been trained with the source data, and then the first n layers of the base network are fixed while the target domain labeled data is used to train the last few layers of the model. It has achieved amazing performance in downstream tasks ranging from sequence classification, sequence-pair classification to question answering, while requiring minimal task-specific architectural modification.

Compared to pre-training, the cost of fine-tuning is cheap. However, the performance of the fine-tuned BERT will decrease if there is a large domain shift between the source data and the target data. How to improve domain adaptation of the BERT model is an important but challenging task. In many practical situations, the source and target distributions may be very different, and in some cases, key target characteristics may not be supported in the source domain. In a specific domain corpus, there are many out-of-vocabulary words. Fortunately, BERT has adopted WordPiece embeddings, [2] which divides a word into server subwords with a 30,000 tokens vocabulary. However, contextual information of a word is deeply changed in a domain-specific corpus, such as biology and finance.

This can be very harmful when a fine-tuning-based approach is applied to token-level tasks, such as question answering, in which case the context must be merged from both directions.[3] considers a co-

training method which slowly adjusts its training set from the source according to the target source. At first, they select the target example that they are most confident of as their training data. Then, they choose training examples from unlabeled data based on their similarity. Similarity is assessed between the source domain and the open domain. As more training examples are included in the training set, specific features between two sets become more similar.

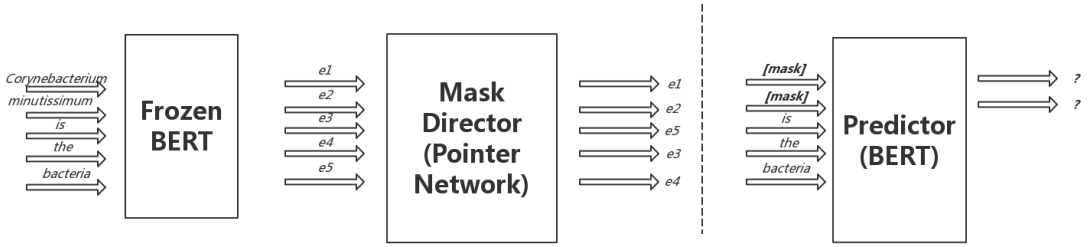

Figure 1. An overview of improved mask method directed by a pointer network.

Inspired by this, we propose a more intelligent sample-efficient mask method which utilizes a pointer network to decide which tokens should be preferentially masked. Pointer network [4] was first proposed to deal with some combinational optimization problems with an attention mechanism, such as sorting the elements of a given set. The order of the token representation plays an important role in the efficiency of continual pre-training. This method trains two neural networks, a mask director MD and a predictor P. Fig. 1 overviews our new training task settings. Firstly, we train a pointer network MD which takes token representation produced by original BERT as input and outputs order of the tokens. Secondly, we replace the top tokens with "[mask]" in the sentence. Thirdly, we train a pre-trained BERT model P to predict masked tokens as the original BERT model does. If P achieves a good performance, MD will get low scores. To achieve high scores, MD will be driven to output more accurate order which will improve recovery difficulty for P. Compared with direct fine-tuning, the advantage of our approach is that tokens of new domain will get more appropriate vector representation and since MD can give a proper option for masking, this extra process is efficient.

We prove the effectiveness of our method by comparing it against BERT on a biomedical corpora. Our method achieves better performance on some tasks while only consuming a small portion of the training budget.

2. Background

2.1. Masked Language Model

Word representation learning, which tries to summary distribution of a large amount of text, has always been fundamental work since it was proposed. Previous models such as Word2Vec [5], GloVe [6] focuse on predict words which are masked in advance from one or two direction.

ELMo [7] uses the combation of individually trained LSTMs on different directions to extract features for downstream tasks. While ELMo combines context from two LSTMs, BERT extracts context from two direction at the same time. It is dependent on a masked language model and pre-trained using bidirectional transformers [8]. The architecture of BERT is shown in Fig. 2. BERT uses a mask language model to predict randomly masked words in order. Therefore, it can be used to learn bidirectional representation. In addition, it achieves amazing performance on most NLP tasks while allowing minimal modifications to the task specific architecture. According to the authors of BERT, combining information from two-way representations, rather than one-way representations, is essential for encoding words in natural languages.

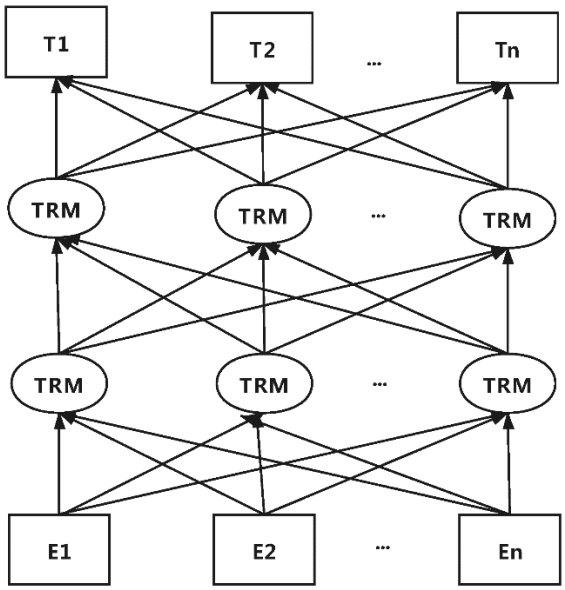

Figure 2. Architecture of BERT.

2.2. Pointer Network

Pointer network is a kind of neural network, which works with variable input and uses the attention mechanism [9, 10] to learn and predict the statistical probability of the output sequence, where the elements are discrete markers corresponding to the positions in the input sequence. That is to say, the pointer network uses its dimension which is dynamic and corresponds to the soft-max output of each input sequence, so the output space is constrained to observe the input sequence.

The structure of the pointer network is shown in Fig. 3. This sequence of sequence models is primarily comprised of two components- an encoder and a decoder. The encoder represents input as a vector which contains compressed hidden information, and the decoder transforms this vector into output. The attention mechanism enables the decoder to query the entire output of the encoder. The attention mechanism effectively extracts meaningful information from the input[11]. It calculates corresponding weights for each element of the input sequence, telling the decoder network which parts of input are important. This method allows the decoder to concentrate more on meaningful information in the input that is relevant to the final output, improving the result. LSTM, one type of recurrent neural network, compared with feed-forward neural networks, has the characteristics of cyclic connections, which makes it more efficient to model sequences [12].

3. Method

We put forward a sample-efficient training task to promote performance of pre-trained BERT in domain-specific NLP tasks. Specifically, instead of applying fine-tuning directly to downstream tasks, we continue to train BERT with domain corpus efficiently in advance. Our proposed model takes tokens in one sentence as input, and generates an ordered token sequence according to tokens' recovery difficulty. Instead of random selection, our language model training method replaces some tokens according to their ranking, i.e. we train a BERT model to predict top tokens which are replaced by "[mask]" in original sentences. This improved method avoids training data waste resulting from random masking, so it promotes utilization ratio of target domain corpus.

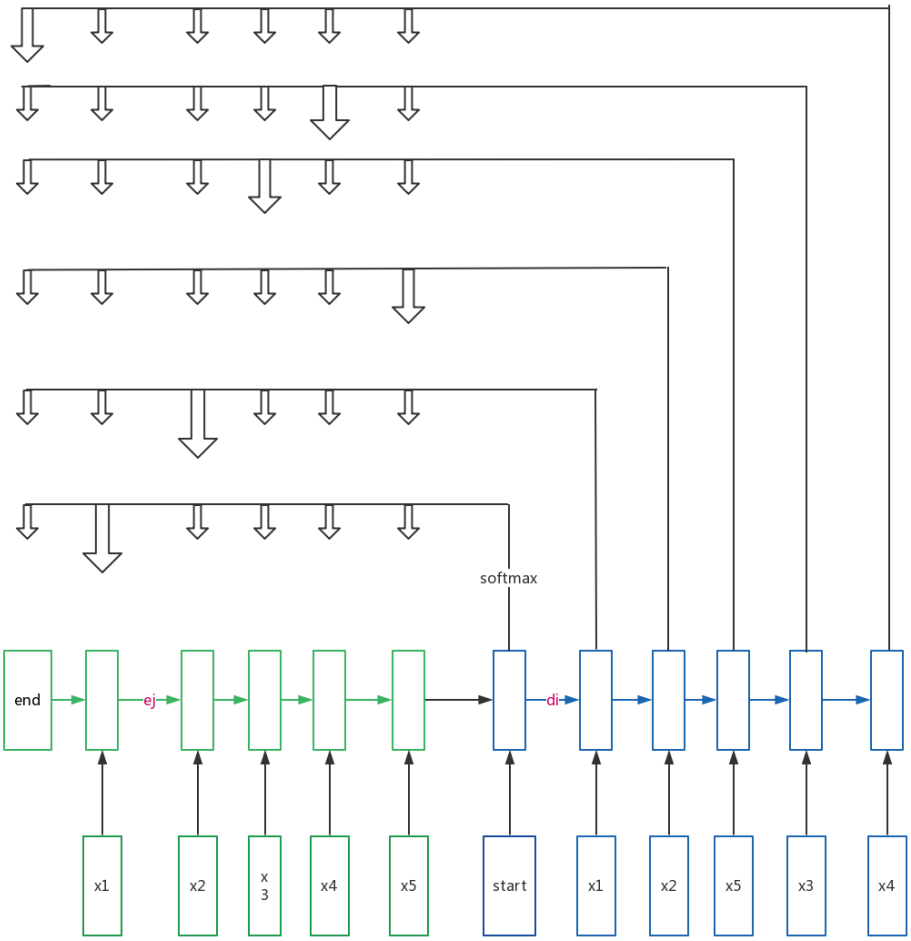

Figure 3. Architecture of the pointer network.

3.1. Mask Director

Before training a pointer network, we use original BERT to encode every token in target domain sentences. Then the token sequence is fed into the decoder of the pointer network, and the token is fed into each time step until the sequence ends. The end of the sequence is marked with a special end marker. Then, the mask controller switches to the decoding mode, and in each time step, an element is generated in the output sequence of the decoder until the end mark appears. Before that, the whole process is over.

The probability of a specific order of given sentences $P(o|s)$ could be formalized as

$$P(o|s) = \prod_{i=1}^{n} P(o_i|o_{i-1}, \dots, o_1, s) \tag{1}$$

The probability $P(o_i|o_{i-1}, \dots, o_1, s)$ is calculated by the pointer network:

$$P(o_i|o_{i-1}, \dots, o_1, s) = softmax(u^i)_{O_i} \tag{2}$$

$$u_j^i = v^\top tan\, h\left(W^\top \begin{bmatrix} e_j \\ d_i \end{bmatrix}\right), j = (1, \dots, n) \tag{3}$$

Where $e_j, d_i \in R^h$ are outputs of encoder and decoder of the pointer network. $v \in R^h$ and $W \in R^{(2h)\times h}$.

Encoder The encoding representation of the pointer network could be formalized as:

$$e_j = LSTM\left(Enc\left(s_{o_j}\right), e_{j-1}\right), j = (1, \ldots, n) \tag{4}$$

Where $Enc\left(s_{o_j}\right)$ indicates the encoding of sentence s_{o_j}.

Decoder Similarity, the decoding representation of the pointer network is formalized as:

$$d_i = LSTM\left(Enc\left(s_{o_i}\right), d_{i-1}\right), i = (1, \ldots, n) \tag{5}$$

Assuming that we have m training examples $(x_i, y_i)_{i=1}^{m}$, where x_i indicates a sequence of sentences with a specific permutation of y_i, and y_i is in gold order o^* which is based on the performance of predictor. The goal is to minimize the loss function $l(\theta)$:

$$l(\theta) = -\frac{1}{m}\sum_{i=1} log\ P\left(y_i|x_i;\theta\right) + \frac{\lambda}{2}|\theta|_2^2 \tag{6}$$

Where $P(y_i|x_i;\theta) = P(o^*|s = x_i;\theta)$ and λ is a hyper-parameter of regularization term. θ indicates all trainable parameters.

3.2. Predictor

The initial parameter of the predictor is set up as the original BERT pre-trained on English Wikipedia and BooksCorpus [13].

The tokens with top ranking are replaced with a [MASK] token. Then the predictor learns to maximize the likelihood of masked tokens and outputs tokens with the highest likelihood from vocabulary. In the settings of experiments, top 15% of all tokens are masked in each input sequence. The predictor tries to predict masked tokens as the original BERT does. We don't back-propagate the predictor's loss through the mask director. That the predictor makes a correct prediction about one masked token means the mask director makes one error. The mini-batch stochastic gradient method is used to update the parameters of the mask director. We take turns training the mask director and the predictor. After training, we throw out the mask director and fine-tune the predictor on downstream tasks.

4. Experiments

In this part, we quantitatively study the proposed method and compare the performance of the proposed method with the original BERT model which is only pre-trained on the general domain corpus. We give BERT the weight of pre-training, which is pre-trained on English Wikipedia and BookCorpus..

4.1. Settings

*W*e use the original BERT code released by its author to realize our training method. The corpus for training our method is full texts of 1M papers randomly selected from Semantic Scholar which uses amazing AI and engineering technology to understand the semantics of scientific literature, so as to help scholars discover relevant research paper [14]. We set a maximum sentence length of 128 tokens as the original BERT does. All released code is converted to be compatible with PyTorch by using the Transformers library.

4.2. Tasks and Datasets

Tasks We tested our method on five large public datasets across three core NLP tasks: named entity recognition(NER), relation extraction(RE) and dependency parsing (DEP). Named entity recognition is a fundamental NLP task that recognizes numerous proper nouns from texts. Many complex models were proposed to solve the problem [15]. However, with BERT, you only need to fine-tune the output layer based on the representations encoded by pre-trained BERT to achieve better results. Relation extraction is a task that classifies relations of named entities. Sentence classifier of original BERT can

be used to do this by using a [CLS] token. Dependency parsing is a task of analyzing the syntactic structure of a sentence. F1 scores on the three tasks are reported.

Datasets We use NCBI Disease [16] dataset and JNLPBA [17] dataset for NER task. GAD [18] and ChemPort [19] are selected for RE tasks and they contain two types of relations, which are gene-disease relations and protein-chemical relations. For DP task, we choose GENIA [20] dataset.

4.3. Results

Table 1. F1 Scores of BERT and our method on five datasets across three fundamental NLP task.

Task	Dataset	SOTA	BERT	Our Method
NER	NCBI Disease	89.36	86.37	87.29
	JNLPBA	78.58	75.92	76.02
RE	GAD	83.93	80.34	81.35
	ChemPort	76.46	73.85	74.51
DEP	CENIA	92.84	91.35	91.55

Table 1 summarizes the F1 scores of the experiment results. We observe that our method outperforms the original BERT on all the datasets. Our method achieves a higher F1 score on the three tasks than BERT. However, our method is still worse than the SOTA method. The SOTA model for NCBI-disease, GAD and ChemPort is BioBERT [21], which is a BERT model trained on 18B tokens from biomedical papers. The SOTA model for JNLPBA is a BiLSTM-CRF ensemble trained on multiple NER datasets [22]. The SOTA model for GENIA uses part-of-speech features [23, 24].

5. Related Work

Recent work on domain-specific pre-trained models includes BioBERT [21], ClinicalBERT [25,26] and SciBERT [27]. BioBERT is trained on PubMed abstracts and PMC full text articles, ClinicalBERT is trained on clinical text from the MIMIC-III database [28], and SciBERT is trained on full text of biomedical and computer science papers from the Semantic Scholar corpus [14]. The main idea of them is training BERT model on domain data and does not improve the pre-training method efficiency.

6. Conclusion

In this paper, we proposed one improved training method to promote domain adaptation of the pre-trained BERT language model. We illustrate the utilization of examples masked by the mask director to update BERT model. We show that by adding a step of using pointer network to sort tokens in a sentence for mask tasks, we can achieve better results than directly fine-tuning pre-trained BERT on the downstream tasks. Compared to random masking, our method is more sample-efficient, which only consumes a small portion of domain data.

7. Reference

[1] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.*NAACL*, 2019, pp4171-4186.
[2] Wu Y, Schuster M, Chen Z, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint*, arXiv:1609.08144
[3] Chen M, Weinberger K Q, Blitzer J. Co-training for domain adaptation. *Advances in neural information processing systems*, 2011, pp2456-2464.
[4] Vinyals O, Fortunato M and Jaitly N. Pointer networks. *Advances in Neural Information Processing Systems*, 2015, pp2692-2700.
[5] Mikolov T, Sutskever I, Chen K, Corrado G S and Dean J. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing*, 2013, pp3111-3119.
[6] Pennington J, Socher R and Manning C D. Glove: Global vectors for word representation. *EMNLP, 2014*, pp 1532-1543.

[7] Peters M E, Neumann M, Iyyer M, et al. *NAACL*, 2018 pp2227-2237.
[8] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017, pp5998-6008.
[9] Chorowski J, Bahdanau D, Serdyuk D, et al. Attention-based models for speech recognition. *Advances in Neural Information Processing Systems*, 2015, pp 577-585.
[10] Luong T, Pham H and Manning C D. Effective approaches to attention-based neural machine translation. *EMNLP*, 2015, pp1412-1421.
[11] Hochreiter S and Schmidhuber J. Long short-term memory. *Neural Computation* 9 pp1735-1780.
[12] Sak H, Senior A W and Beaufays F. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *CoRR*, URL http://arxiv.org/abs/1402.1128.
[13] Zhu Y, Kiros R, Zemel R S et al. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *ICCV*, 2015, pp19-27.
[14] Ammar W, Groeneveld D, Bhagavatula C et al. Construction of the literature graph in semantic scholar. *NAACL*, 2018, pp84-91.
[15] Giorgi J M and Bader G D. Transfer learning for biomedical named entity recognition with neural networks. *Bioinform*, 34, pp4087-4094.
[16] Dogan R I, Leaman R and Lu Z. NCBI disease corpus: A resource for disease name recognition and concept normalization. *J. Biomed. Informatics*, 47, pp1-10.
[17] Collier N and Kim J. Introduction to the bio-entity recognition task at JNLPBA. *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*. URL https://www.aclweb.org/anthology/W04-1213/.
[18] Bravo A, Gonzalez J P. Queralt-Rosinach N, et al. Extraction of relations between genes and diseases from text and large-scale data analysis: implications for translational research. *BMC Bioinform*. 16, pp55:1-55:17.
[19] Kringelum J, Kjrul S K, Brunak S, et al. Chemprot-3.0: a global chemical biology diseases mapping. *Database* 2016 URL https://doi.org/10.1093/database/bav123.
[20] Kim J, Ohta T, Tateisi Y, et al. GENIA corpus - a semantically annotated corpus for bio-textmining. *Proceedings of the Eleventh International Conference on Intelligent Systems for Molecular Biology*, pp180-182.
[21] Lee J, Yoon W, Kim S, et al. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinform*, 36, pp1234-1240.
[22] Yoon W, So C H, Lee J, et al. Collabonet: collaboration of deep neural networks for biomedical named entity recognition. *BMC Bioinform*, 20-S, pp55-65.
[23] Nguyen D Q and Verspoor K. From POS tagging to dependency parsing for biomedical event extraction. *BMC Bioinform*, 20, pp72:1-72:13.
[24] Dozat T and Manning C D. Deep biaffine attention for neural dependency parsing. *ICLR*, 2015, URL https://openreview.net/forum?id=Hk95PK9le.
[25] Alsentzer E, Murphy J R, Boag W, et al. Publicly available clinical BERT embeddings. *CoRR* abs/1904.03323, URL http://arxiv.org/abs/1904.03323.
[26] Huang K, Altosaar J and Ranganath R. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *CoRR* abs/1904.05342, URL http://arxiv.org/abs/1904.05342.
[27] Beltagy I, Lo K and Cohan A. Scibert: A pretrained language model for scientific text. *EMNLP, 2019* pp 3613-3618.
[28] Johnson A E W, Pollard T J, Shen L, et al. Mimic-iii, a freely accessible critical care database. *Scientic Data*, 3.