Investigation of Improving The Pre-Training And Fine-Tuning of BERT Model For Biomedical Relation Extraction

Peng Su (psu@udel.edu)
University of Delaware
K. Vijay-Shanker
University of Delaware

Research Article

Keywords: Deep Learning, Transformer, BERT, Text Mining, Biomedical Relation Extraction

Posted Date: October 26th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-640112/v1

Version of Record: A version of this preprint was published at BMC Bioinformatics on April 4th, 2022. See the published version at https://doi.org/10.1186/s12859-022-04642-w.

RESEARCH

Investigation of Improving the Pre-training and Fine-tuning of BERT Model for Biomedical Relation Extraction

Peng Su* and K. Vijay-Shanker

*Correspondence: psu@udel.edu
Department of Computer and Information Science, University of Delaware, Newark, USA
Full list of author information is available at the end of the article

Abstract

Background: Recently, automatically extracting biomedical relations has been a significant subject in biomedical research due to the rapid growth of biomedical literature. Since the adaptation to the biomedical domain, the transformer-based BERT models have produced leading results on many biomedical natural language processing tasks. In this work, we will explore the approaches to improve the BERT model for relation extraction tasks in both the pre-training and fine-tuning stages of its applications. In the pre-training stage, we add another level of BERT adaptation on sub-domain data to bridge the gap between domain knowledge and task-specific knowledge. Also, we propose methods to incorporate the ignored knowledge in the last layer of BERT to improve its fine-tuning.

Results: The experiment results demonstrate that our approaches for pre-training and fine-tuning can improve the BERT model performance. After combining the two proposed techniques, our approach outperforms the original BERT models with averaged F1 score improvement of 2.1% on relation extraction tasks. Moreover, our approach achieves state-of-the-art performance on three relation extraction benchmark datasets.

Conclusions: The extra pre-training step on sub-domain data can help the BERT model generalization on specific tasks, and our proposed fine-tuning mechanism could utilize the knowledge in the last layer of BERT to boost the model performance. Furthermore, the combination of these two approaches further improves the performance of BERT model on the relation extraction tasks.

Keywords: Deep Learning; Transformer; BERT; Text Mining; Biomedical Relation Extraction

Background

In recent years, biomedical text mining has become an urgent problem since the manual curation lags far behind the explosive growth of biomedical literature. Relation extraction (RE) is an important task in biomedical text mining that aims to identify the relations between biomedical entities mentioned in the text. Substantial techniques have been developing to extract biomedical relations such as protein-protein interactions (PPI [1]), drug-drug interactions (DDI [2]), chemical protein interactions (ChemProt [3]). After the extraction of different relations, we can convert the unstructured literature to structured information, which is a crucial step towards natural language understanding applications like automated reasoning [4], machine translation [5], question answering [6], etc.

Extraction of various relations in biomedical domain has attracted tremendous attentions and many different methods have been proposed [7–11]. Recently, the emerging of language models dominates the RE field with their superior performance [12–15]. Applying language models on relation extraction problem includes two steps: the pre-training and the fine-tuning. In the pre-training step, a vast amount of unlabeled data can be utilized to learn a language representation. The fine-tuning step is to learn the knowledge in task-specific (labeled) datasets through supervised learning. Among all the language models, BERT [14]—a language representation model based on bidirectional transformer [16], attracts lots of attentions from researchers in different fields.

BERT is designed to learn a universal and context-dependent language representation using the transformer [16]. Originally, BERT was proposed to learn the representation for general domain. To make the model generalize better in biomedical domain, several BERT adaptations for the biomedical domain have been proposed such as BioBERT [17], BlueBERT [18], SciBERT [19], and PubMedBERT [20]. BioBERT [17] further pre-trains the BERT model on PubMed abstracts and PMC full-length articles; BlueBERT [18] is further pre-trained on PubMed abstracts and MIMIC-III clinical notes [21]; A collection of 1.14M articles from Semantic Scholar [22] are used to pre-train SciBERT model [19]. While the first three biomedical BERT models are pre-trained on the basis of the original BERT, PubMedBERT [20] is pre-trained with whole-word masking from scratch using PubMed abstracts.

In this work, we propose and evaluate two approaches to improve relation extraction with BERT-style models. We first consider the approach for the pre-training phase and later discuss the other that is related to the fine-tuning phase. Our goal to improve the pre-training of BERT model is motivated by the results of the experiments we conducted to investigate differing performances of several BERT-based models for the biomedical domain. Different models such as BioBERT [17], SciBERT [19] and BlueBERT [18] are developed to adapt to the biomedical domain. One of the primary differences between them is the corpora used in the pre-training for domain adaptation. Those pre-trained BERT models are suppose to have similar performance on similar applications since they were pre-trained on similar biomedical data. However, our experiments reveal that those models have significantly different performance on the same set of RE tasks (Table 2). Therefore, we hypothesize that the text used in the pre-training for domain adaptation can have a significant impact on the downstream applications.

In order to leverage the pre-training for a specific task, we introduce another level of adaptation to adjust the domain adaptation to specific sub-domains in this work. We call this part sub-domain adaptation. To fulfill this task, we add one more pre-training step on sub-domain data. For example, for a RE task like the extraction of drug-drug interactions (DDI), we investigate whether adding more drug-specific text can help over general biomedical domain knowledge. After the sub-domain adaptation, we expect that the pre-trained BERT model will generalize better on the specific tasks. There are several studies using the labeled training data in the pre-training of language model for better generalization [23, 24]. However those methods are not feasible for our tasks since we also need the context of the training sentences (for Next Sentence Prediction in pre-training of BERT). Thus, we propose a new approach to acquire more general pre-training data for our tasks.

Our motivation to improve the fine-tuning of BERT is from the following observations. All previous work utilizing BERT-based models for classification tasks (including relation extraction) employ a standard way of fine-tuning using the classification token (CLS) alone among the last layer. Thereby all information contained in other final layer nodes is completely ignored during the fine-tuning process. However, the ignored knowledge in the last layer of BERT model is utilized for other tasks like sequence tagging. From this point of view, the BERT model is fine-tuned with one less layer for classification tasks. This drives us to design a new mechanism of using all the information in the last layer for the classification tasks.

In addition, our investigation of improving fine-tuning mechanism is partially driven by the insights in [25], in which the authors find that the BERT model learns the representation similar to traditional natural language processing (NLP) pipelines. Based on the findings in [25], the basic syntactic aspects of the text appear to be learned in the lower layers, while high-level semantic information appears in the higher layers of BERT. Since relation extraction tasks are concerned with the semantic relations between entities, we were curious about whether the upper (including top) layers contain important information about RE tasks. During this investigation, we employ the edge probing technique [26] to measure how much relevant information (about relation extraction tasks) the last layer contains. The results illustrate that the last layer of BERT model contains useful information, but it is unused in the original fine-tuning method of only using classification (CLS) token. To incorporate the unused information from the last layer into fine-tuning, we explore two different methods: recurrent neural network (RNN) and attention mechanism [27]. The summarized knowledge will be concatenated with the classification token as the model output in an improved fine-tuning mechanism. We call it fine-tuning with information summarization of the last layer (SLL fine-tuning).

To demonstrate the effectiveness of our proposed approaches in the pre-training and fine-tuning process, we evaluate them on three extensively studied relation extraction tasks in biomedical field: protein-protein interactions (PPI [1]), drug-drug interactions (DDI [2]), and chemical-protein interactions (ChemProt [3]). The experiment results illustrate that both sub-domain adaptation and the proposed fine-tuning mechanism can boost the model performance on all the tasks. In addition, the combined approach outperforms previous methods on the three benchmark datasets.

In summary, the contributions of this work are:

- Proposing a sub-domain adaptation to improve the pre-training of BERT model for specific tasks;
- Utilizing the edge probing technique to explore the ignored knowledge in the last layer of BERT model;
- The SLL fine-tuning mechanism is proposed to utilize all the available knowledge in the last layer to boost the BERT model performance;
- State-of-the-art performance is achieved on three relation extraction benchmark datasets;

Dataset	Instance #	Train	Dev	Test
PPI(AIMed)	5,834	-	-	-
DDI	33,508	22,233	5,559	5,716
ChemProt	45,048	18,035	11,268	15,745

Table 1: Datasets statistics for PPI, DDI, and ChemProt. For the AIMed dataset of PPI, there are only two labels: Positive and Negative. The ChemProt corpus is labeled with five positive classes (CPR:3, CPR:4, CPR:5, CPR:6, CPR:9) and the negative class. Similarly, the DDI dataset contains four positive labels (ADVICE, EFFECT, INT, MECHANISM) and one negative label.

Results and discussion

In this work, we experiment with two types of BERT model adapted for the biomedical domain: (1) the BERT model adapted from general domain using biomedical text (e.g., BioBERT [17], BlueBERT [18], SciBERT [19]); (2) the BERT model adapted from scratch using biomedical text (e.g., PubMedBERT [20]). Our study starts with the comparison of BERT models from the first type. Then we experiment with both types of BERT models on the proposed approaches. We choose BioBERT from the first type of BERT models as it generally outperforms SciBERT and BlueBERT on our tasks (Table 2). For the second type of BERT model, we use PubMedBERT in our experiments.

We verify the effectiveness of the proposed approaches on three benchmark datasets and we show the statistics of these datasets in Table 1. We use the AIMed corpus [28] for the PPI task. For the ChemProt and DDI tasks, we use the datasets in [3] and [2], respectively. We employ the the same spilt of training, development, and test sets with the PubMedBERT model [20] during the model evaluation. For the AIMed corpus, standard sets of training and test are not available, so we apply 10-fold cross-validation during evaluation. We employ precision (P), recall (R) and F1-score (F) to evaluate the model on the PPI task since it is a binary classification problem. However, the models for ChemProt and DDI tasks will be evaluated with micro precision (P), recall (R) and F1 score (F) on the non-negative classes since they are multi-class classification problem.

In this section, we first discuss the effect of pre-training corpora on the BERT models. Next, the experiment results on sub-domain adaptation of BERT models are demonstrated. Then, we present the results of exploring the learned knowledge in the last layer and the SLL fine-tuning. Also, the model performance of combining the proposed approaches for the pre-training and fine-tuning is provided for the three tasks. Finally, we perform an analysis on the learned attention weights from our SLL fine-tuning.

Impact of corpus on domain adaptation of different BERT models

We begin with a study of the impact of the corpora used to adapt BERT-based models to the biomedical domain. Specifically, we experiment with the three well-known models: BioBERT [17], BlueBERT [18] and SciBERT [19]. Considering these three models have the same architecture, the primary difference between them is the corpus used to adapt to the biomedical domain during pre-training. These three models are suppose to have similar performance, but the results are contrary to our

Model	PPI	DDI	ChemProt
BioBERT	81.0	79.0	75.3
SciBERT	78.8	78.7	74.4
BlueBERT	71.9	76.8	71.2

Table 2: Performance (F1 score) of BERT model on the ChemProt, DDI, and PPI datasets.

Model	PPI			DDI			ChemProt		
	P	R	F	P	R	F	P	R	F
BlueBERT	69.3	75.0	71.9	76.2	77.4	76.8	70.9	71.5	71.2
BlueBERT (-M)	**76.6**	**83.1**	**79.6**	**80.0**	**78.5**	**79.2**	**74.7**	**75.8**	**75.2**

Table 3: Performance of BlueBERT model on the PPI, ChemProt, and DDI tasks before and after removing MIMIC-III from the domain adaptation data. -M: Subtract the MIMIC-III clinical notes.

expectation as they are summarized in Table 2. BlueBERT shows performance drop-off compared to the other two on our three datasets, especially on the PPI set. Thus, the text used in pre-training for domain adaptation appears to have a surprisingly significant influence on the performance of BERT models on downstream tasks. A noticeable difference with BlueBERT, when compared to the other two models, is the inclusion of clinical notes text in the domain adaptation process, which differs considerably from the text used for pre-training other two models. Since all three evaluation sets are irrelevant to the clinical domain, we conduct an experiment to see if the removal of this extraneous material from pre-training adaptation would impact the results. The results, shown in Table 3, suggest an improvement in BlueBERT's performance (BlueBERT almost achieves the same results as the other two). This observation leads us to conjecture that task-related data in the pre-training might yield better generalization of BERT models on downstream tasks.

Sub-domain adaptation

While all BERT models try to use a large corpus of text from the biomedical domain to obtain the domain-adapted versions, our results indicate that the differences between the domain adaptation text might have a noticeable impact on individual tasks, as each task requires its own specific knowledge. Therefore, we wish to further consider the pre-training data for the pre-training phase.

Given our investigation involving several RE tasks, we consider the approach of utilizing task-related data to further adjust the adaptation for the domain underlying the tasks. Specifically, we extract data (unlabeled text) for three sub-domains from PubMed for our tasks: protein/gene (P/G) domain, drug (D) domain and chemical+protein (CP) domain. These three sets of abstracts are used for sub-domain adaptation setup to produce three differently adapted BERT models: BERT(+P/G), BERT(+D) and BERT(+CP), respectively.

Table 4 presents the results of our experiments on sub-domain adaptation. The results are reported for both BioBERT and PubMedBERT, which have already been adapted to the biomedical domain. Their performances are shown in the first row and the fifth row, and serve as baselines to compare with those models that have additional pre-training for sub-domain adaptation. The other results in Table 4 shows

Model	PPI			DDI			ChemProt		
	P	R	F	P	R	F	P	R	F
BioBERT	79.0	83.3	81.0	79.9	78.1	79.0	74.3	**76.3**	75.3
BioBERT (+P/G)	**82.5**	**83.7**	**83.0**	76.1	77.6	76.9	76.5	74.2	75.3
BioBERT (+D)	81.5	80.9	81.2	**81.9**	78.4	**80.1**	**76.7**	74.4	75.6
BioBERT (+CP)	81.3	83.7	82.4	78.7	**79.0**	78.8	76.6	76.1	**76.4**
PubMedBERT	80.1	84.3	82.1	82.6	81.9	82.3	78.8	75.9	77.3
PubMedBERT (+P/G)	**81.2**	**85.5**	**83.3**	83.7	80.5	82.0	**80.5**	75.5	77.9
PubMedBERT (+D)	79.1	85.3	82.0	**84.1**	81.7	**82.9**	80.4	74.6	77.4
PubMedBERT (+CP)	79.6	84.7	82.0	81.1	**82.7**	81.9	79.4	**77.5**	**78.4**

Table 4: BERT performance after pre-training with sub-domain data. +P/G: add Protein/Gene-related PubMed abstracts as sub-domain data; +D: add Drug-related PubMed abstracts as sub-domain data; +CP: add protein-related and chemical-related PubMed abstracts as sub-domain data.

Fig1.png

Figure 1: Learned knowledge of training data in the layers of BioBERT. L is the total layers of BERT model. "Measurement of Knowledge" (Δ_{τ}^{L}) is defined in the Methods section.

the sub-domain adaptation can boost the model performance on related tasks. Note that not only are similar results obtained for BioBERT and PubMedBERT, but also for each task, the addition of sub-domain adaptation phase improves the performance. Furthermore, the maximum improvement is obtained when most relevant sub-domain text is used in this sub-domain adaptation phase. Specifically, the best results are obtained for the PPI task by using "protein/gene" related text. Also, the "drug" text leads to the highest performance for the DDI task. Both BioBERT and PubMedBERT obtain maximum benefits in the use of "chemical+protein" text for sub-domain adaptation on the ChemProt task. On the other hand, adding drug related text does not help for the PPI task. Also, the addition of "protein/gene" text actually hurts the performance of both models when used for the DDI task.

All the experiment results indicate that the pre-trained BERT models generalize differently on different tasks, and it is beneficial to add another level of adaptation on task-specific data. In this work, we have demonstrated that using entity-related data in the sub-domain adaptation helps the model generalization on relation extraction tasks.

Learned knowledge in the last layer of BERT

In this subsection, we focus on improving the fine-tuning process of BERT. Using the technique of probing classifier, we show that the last layer of BERT model contains useful information that can be exploited to improve performance on downstream tasks (Fig. 1). Specifically, we compare the performance of probing classifiers using all L layers and first (L-1)-th layers to measure the knowledge captured in the last layer (not present in the previous layers). We can see that the probing classifier using the information in all L layers performs better on all three tasks, which means many instances are predicted correctly by adding the knowledge in the last (L-th) layer, but not using the knowledge in the first (L-1)-th layers. Thus it is beneficial to

Model	PPI			DDI			ChemProt		
	P	R	F	P	R	F	P	R	F
BioBERT	79.0	83.3	81.0	79.9	78.1	79.0	74.3	**76.3**	75.3
BioBERT_SLL_LSTM	80.2	84.0	82.0	80.5	78.5	79.5	77.6	74.4	76.0
BioBERT_SLL_biLSTM	80.2	82.7	81.4	80.8	78.5	79.6	**77.9**	73.9	75.9
BioBERT_SLL_Att	**80.7**	**84.4**	**82.5**	**81.6**	**79.4**	**80.5**	77.5	75.1	**76.3**
PubMedBERT	80.1	84.3	82.1	82.6	81.9	82.3	78.8	75.9	77.3
PubMedBERT_SLL_LSTM	79.8	**85.6**	82.6	82.6	**82.8**	82.7	**78.9**	77.0	**77.9**
PubMedBERT_SLL_biLSTM	80.5	82.6	81.7	82.6	81.4	82.0	78.5	76.5	77.5
PubMedBERT_SLL_Att	**81.3**	85.0	**83.1**	**84.3**	82.7	**83.5**	78.3	**77.6**	**77.9**

Table 5: Performance of BERT model on PPI, DDI, and ChemProt. BioBERT/PubMedBERT_SLL_LSTM: model of summarizing the outputs of the last layer using LSTM; BioBERT/PubMedBERT_SLL_biLSTM: model of summarizing the outputs of the last layer using biLSTM; BioBERT/PubMedBERT_SLL_Att: model of summarizing the outputs of the last layer using attention mechanism.

utilize those information during fine-tuning of BERT. In addition, the information in the last layer is automatically generated during training, we can just utilize it without extra cost.

SLL fine-tuning: utilizing the summarized information in the last layer

Having shown that the last layer contains some useful information, which is not exploited by current transformer-based models for RE, we consider two different methods to utilize it and incorporate the summarized information in a refined fine-tuning process. In Table 5, we provide the model performance using different methods: RNN models (both LSTM and biLSTM) and attention mechanism. Clearly, the method of applying attention mechanism on the outputs of last layer (BERT_SLL_Att) obtains the best performance on the three tasks. Specifically, the attention-based method on BioBERT model achieves F1 score improvement of 1.5%, 1.5%, and 1.0% on PPI, DDI, and ChemProt, respectively. Similarly, we also observe F1 improvements with the PubMedBERT model for the three tasks.

Even though the LSTM and biLSTM methods show less improvement compared to the use of attention mechanism, they improve the model performance in most of the cases. We also observe that LSTM and biLSTM have very similar performance, with LSTM being slightly better in general. Since the training of BERT considers the context of words from both directions, we hypothesize that the position information is already encoded in the word representation and biLSTM will not provide extra information (which partially explains the experiment results). Therefore, we will only utilize LSTM when considering RNN model for the experiments in the following subsections.

Combining sub-domain adaptation and SLL fine-tuning mechanism

The two proposed techniques can be combined as they are for two independent stages of BERT model. As shown in Table 6, combining the sub-domain adaptation and the proposed fine-tuning mechanism can further boost the model performance on all the three tasks. The first two rows for BioBERT and PubMedBERT are just repetitions of the model performance from Table 5 and serve as the baselines for

Model	PPI			DDI			ChemProt		
	P	R	F	P	R	F	P	R	F
BioBERT	79.0	83.3	81.0	79.9	78.1	79.0	74.3	76.3	75.3
BioBERT_SLL_Att	80.7	84.4	82.5	81.3	80.1	80.7	76.5	**77.1**	76.8
BioBERT_SLL_Att (+P/G)	**83.1**	**84.7**	**83.8**	80.4	79.7	80.0	78.4	75.1	76.7
BioBERT_SLL_Att (+D)	81.5	84.5	82.9	**82.6**	**81.2**	**81.9**	76.8	74.7	75.7
BioBERT_SLL_Att (+CP)	82.5	84.2	83.3	81.7	77.0	79.3	**78.9**	75.2	**77.0**
PubMedBERT	80.1	84.3	82.1	82.6	81.9	82.3	78.8	75.9	77.3
PubMedBERT_SLL_Att	81.3	85.0	83.1	84.3	82.7	83.5	78.3	77.6	77.9
PubMedBERT_SLL_Att (+P/G)	81.1	**87.1**	**84.0**	83.6	80.6	82.1	**79.8**	77.0	78.4
PubMedBERT_SLL_Att (+D)	**81.4**	84.5	82.9	84.9	**83.2**	**84.0**	79.5	75.9	77.7
PubMedBERT_SLL_Att (+CP)	**81.4**	85.7	83.4	**85.0**	81.4	83.2	79.7	**77.7**	**78.7**

Table 6: BERT performance after combining sub-domain adaptation and the refined fine-tuning mechanism. BioBERT/PubMedBERT: original BERT model; BioBERT/PubMedBERT_SLL_Att: model of summarizing the outputs of the last layer using attention mechanism. +P/G: add Protein/Gene-related PubMed abstracts as sub-domain data; +D: add Drug-related PubMed abstracts as sub-domain data; +CP: add protein-related and chemical-related PubMed abstracts as sub-domain data.

Model	PPI			DDI			ChemProt		
	P	R	F	P	R	F	P	R	F
BioBERT	79.0	83.3	81.0	79.9	78.1	79.0	74.3	76.3	75.3
BioBERT_SLL_Att	80.7	**84.4**	82.5	**81.3**	80.1	**80.7**	**76.5**	**77.1**	**76.8**
BioBERT_SLL_Att*	**82.3**	83.5	**82.8**	79.7	77.6	78.6	76.4	74.5	75.4
PubMedBERT	80.1	84.3	82.1	82.6	81.9	82.3	**78.8**	75.9	77.3
PubMedBERT_SLL_Att	**81.3**	85.0	**83.1**	**84.3**	**82.7**	**83.5**	78.3	77.6	**77.9**
PubMedBERT_SLL_Att*	80.0	**85.2**	82.4	82.5	80.9	81.7	75.7	**77.7**	76.7

Table 7: Model performance without using the [CLS] token in the last layer. BERT_SLL_Att*: models of fine-tuning with only the summarized information from attention mechanism (without [CLS] token).

the results of using combined techniques. In comparison with the original BioBERT model, the model with combined techniques can improve the F1 score with 2.8%, 2.9%, and 1.7% on the PPI, DDI, and ChemProt tasks, respectively. Similarly, we can also boost the PubMedBERT model performance with 1.9%, 1.7%, and 1.4% F1 score improvement on the three tasks, respectively. As far as we know, PubMedBERT is the state-of-the-art model for these three tasks. After further improving its performance, we now achieve the state-of-the-art performance on all three benchmark datasets.

More analysis: roles of classification ([CLS]) token in fine-tuning

We have shown that the proposed fine-tuning method outperforms original fine-tuning mechanism in Table 5. Both methods include the use of [CLS] token in the last layer, so it will be helpful to understand the role of [CLS] token. Here we experiment with the model of fine-tuning without [CLS] token to investigate the contribution of [CLS] token. In particular, we just drop the [CLS] output and utilize the summarized information from attention mechanism as the output of BERT model.

Fig2a.png

(a) PPI example (Label: Positive).

Fig2b.png

(b) DDI example (Label: EFFECT).

Fig2c.png

(c) ChemProt example(Label: CPR:3).

Figure 2: The visualization of attention weights in the last layer.

Here we experiment with both BioBERT and PubMedBERT models. In Table 7, the third and sixth row show the model performance of employing only the summarized outputs in the last layer (i.e., without [CLS] output). The four rows before (row 1, 2, 4, 5) are only repetitions of the model performance from Table 5. We observe that fine-tuning without using [CLS] token hurts the model performance on most cases and only BioBERT_SLL_Att* model performs slightly better on the PPI task. These results show that the [CLS] token in the last layer contains important information about our classification tasks.

In addition, the authors in [29] explore the contribution of [CLS] tokens from the intermediate layers of BERT, we also experiment with this method and observe worse model performance after incorporating the [CLS] outputs from the intermediate layers in all tasks. Thus, we will not present those results here. Our experiment results imply that it is better to use information from the last layer since the knowledge in the intermediate layers will be transferred to the latter layers during training.

Analysis of attention weights in the SLL fine-tuning

In the previous subsection, we illustrate that utilizing additional attention mechanism yields better performance. We conduct some preliminary experiments to understand whether the additional mechanism focuses on specific parts of the input text for our tasks. We first visualize the weight distribution (from the attention mechanism) on the words (tokens) of the sentences and inspect which words (tokens) the model is focusing on. In Fig. 2, we have chosen three examples from the three tasks (one example for each task) that were predicted incorrectly by the original BioBERT but are correctly classified by the BioBERT model with SLL fine-tuning. In Fig. 2, the darkness of the color represents the attention weight, which means the words with darker color have larger attention weights.

Fig. 2 demonstrates that the attention mechanism is assigning relatively large weights on "informative" words (which express the relationship of the entities), which are usually called "trigger words" for the relations in the NLP field. It appears that the attention mechanism is learning to focus on the "trigger words" in our tasks when making its predictions. For instance, in the example of PPI relation (Fig. 2a), the attention mechanism is assigning larger weights on the entity token (i.e., @PROTEIN$) and the trigger word (i.e., "activates"). The attention mechanism, likewise, assigns larger weights on the trigger words ("inhibit" and "mediate") in the examples of DDI and ChemProt, respectively. Considering the original BioBERT model misclassified these sentences, those examples also show that the attention mechanism is able to learn the knowledge (especially about trigger words) of relation expression.

Previously, we demonstrate that the attention mechanism assigns large weights to relation expression words (trigger word and entity token) through examples. Next, we will consider the attention weights in the corpus level. Specifically, we investigate the distribution of the attention weights on all the words in the dataset. Given that the entities and the trigger words are usually near to each other for the relation expression, we only consider the three words around the entities for the weights distribution. In particular, we see the average of the weights as the global attention weight for the word. In Table 8, we give some examples (words that have large attention weight) from the dataset of the tasks. In the table, most of the words can be considered as "trigger words" of the relations. Thus, we can say that the attention mechanism is able to learn some key knowledge of relation expression.

Task	Word Stem
PPI	activ(ate), complex, associ(ate), interact, human protein, bind, domain, specif(y), receptor
DDI	concomitantli, combin(e), concomit(ant), increas(e), use concurr(ent), decreas(e), inhibit, receiv(e), administ(er)
ChemProt	phosphoryl(ate), attenu(ate), stimul(ate), deriv(e), regul(ate) novel, metabol(ize), reduc(e), induc(e), inhibit

Table 8: The top 10 words with large learned attention weight in PPI, DDI, and ChemProt corpora. For the calculation of global attention weight, we use Porter's stemmer [30] to obtain the word stem for each word since words might in different forms in the sentence. For example, the stem of "activate" is "activ", and the words like "activation" and "activates" share the same word stem.

Methods

In this work, we see relation extraction as a classification problem. Specifically, when a sentence and two entity mentions are given, we have to tell if the sentence expresses a specific relation between the two entities. Here we employ the BERT model to solve the RE problem, and we will design approaches to improve the pre-training and fine-tuning of BERT model.

In this section, we first give some background knowledge about BERT model. Then we describe our approach of improving the pre-training of BERT using sub-domain adaptation. Next, we introduce the basics of utilizing edge probing to explore the learned knowledge in different layers of BERT model. We then discuss the SLL fine-tuning of incorporating the knowledge from the last layer. At last, we give the details of our experiment setup and data pre-processing.

Introduction of BERT

BERT [14] is language representation model using bidirectional transformer [16]. Trained with "masked language model" technique, BERT is able to learn the context of a word from both left and right direction of the text (sentence). To apply the BERT model on a specific task, it needs two steps: 1) pre-training on unlabeled text to obtain general knowledge of a domain; 2) fine-tuning on labeled data to gain the specific knowledge of a task.

Fig3.png

Figure 3: BERT model training process with sub-domain adaptation.

Pre-training of BERT: In this stage, BERT can learn a general language representation via two well-designed tasks: masked language model (MLM) and next sentence prediction (NSP). The MLM technique randomly replaces a portion of tokens with a special token ([MASK] token), and lets the language model predict the replaced tokens. In the original BERT, MLM only selects subwords to mask. BioBERT, SciBERT and BlueBERT follow this convention of subwords masking. The PubMedBERT utilizes whole-word masking (WWM) instead, which enforces the mask of the whole word in MLM if one or more of its subwords are chosen. In NSP, the model is trained to predict whether a sentence is followed another sentence in the original text given a sentence pair. The pre-training of BERT usually utilize a great quantity of unlabeled data. In addition, BERT is originally designed for the general domain, and it is pretrained on two datasets: English Wikipedia and BooksCorpus.

BERT for general domain might generalize poorly on a specific domain since every domain has its unique knowledge. BERT can not gain such knowledge without pre-training on the data for specific domain. For instance, the general BERT model represents the biomedical entity names badly since its pre-training datasets are lack of those names. Thus we need to adapt the BERT model to biomedical domain before applying it to biomedical tasks (a.k.a., pre-training the BERT model on biomedical data). There are two ways of adapting BERT model: (1) continual pre-training using biomedical data from pre-trained BERT for general domain (e.g., BioBERT [17]); (2) pre-training BERT on biomedical data from scratch (e.g., PubMedBERT [20]). Obviously, the difference between these two types of adapted models is whether to use the general domain data before the biomedical domain adaptation. In this paper, we experiment with those two types of adapted BERT models for the biomedical domain.

Fine-tuning of BERT: The task-agnostic representation (from pre-training) can be fine-tuned for diverse downstream tasks via supervised training on labeled datasets. For a specific task, we just need to attach a custom layer on the last layer of BERT. Next, the BERT parameters can be fine-tuned like supervised learning using the labeled data of a specific task. For our classification tasks, we can build a classifier by appending a softmax layer on the output of [CLS] token. The full name of [CLS] is "classification", and it is designed for the purpose of classification in BERT. In the input for BERT, the sequence always starts with [CLS] token.

Improving pre-training using sub-domain data

We can see from Table 2 that the different pre-training data lead to significantly different performance on our focused tasks. Inspired by this observation, we explore whether task-related pre-training data could improve the BERT model. Specifically, we investigate whether adding an extra pre-training step using task-related data after the domain adaptation pre-training to help model generalization. We call it

Fig4.png

Figure 4: Probing classifier architecture. We freeze the parameters of BERT model during the training of probing classifier. Through the learned α, we can know the relevance between each layer and the task. Also, we can tell which layer learns the knowledge for a specific instance by building a series of probing classifier $\{P_\tau^l\}_{l=1}^L$. For the relation extraction instance "RFX5 interacts with histone deacetylase 2", if the probing classifier P_τ^l predicts the interacting relationship between proteins "RFX5" and "histone deacetylase 2" correctly using the information of the first l layers, but P_τ^{l-1} does not predict correctly using the information of the first (l-1) layers. We can say that the knowledge about this instance is learned in the l-th layer.

sub-domain adaptation as we are seeking to tailor the model further for specific tasks.

In sub-domain adaptation, we use the following way to extract the sub-domain data that are related to a specific RE task. We assume that the text containing some type(s) of entity in a relation (e.g., protein for the PPI relation) is relevant to the specific RE task. Since PubMed is our main source of pre-training data, we extract the PubMed abstracts containing specific types of entity as the sub-domain data. Specifically, for the PPI task, we extract abstracts from PubMed via the query "Protein OR Gene" and we obtain 7,729,611 abstracts for protein/gene domain using this query. Similarly, the DDI task involves the drug entities and so we use the PubMed query "Drug" and 5,714,799 abstracts are extracted as pre-training data for drug domain. For the ChemProt task, we use Pubtator [31] to extract the PubMed abstracts that contain protein/gene and chemical entities, and 3,375,380 abstracts are used as the sub-domain data.

The new training process of BERT model is illustrated in Fig. 3. The first box represent the standard creation of BERT models for biomedical domain. The rightmost box also is the standard fine-tuning for the RE tasks. Our experiment here involves the inclusion of sub-domain adaptation in the middle box of Fig. 3.

Improving the fine-tuning of BERT

As we discussed before, the BERT model for relation extraction classification problem only utilizes the classification ([CLS]) token when it is fine-tuned on the training dataset. If the last layer contains useful information about the task, we should incorporate it to boost the model performance during fine-tuning. In this subsection, we first introduce the edge probing technique and then utilize it to verify the usefulness of the information in the last layer of BERT. Then, we propose two methods to incorporate the information from the last layer in a refined fine-tuning mechanism.

Introduction of edge probing

Edge probing is proposed in [26] to measure the quality of encoded information about linguistic structure in a pre-trained encoder (BERT in our case). Specifically, edge probing aims to evaluate how well the word is represented in each position, and what knowledge is learned about the structural information of the sentence. For instance, through edge probing, we can know if the constituent information (like noun phrase or verb phrase) of the words is encoded in the representation from the encoder. To achieve this goal, edge probing converts this problem into classification tasks and builds probing classifiers using the word representations and the expected labels. Based on the performance of the probing classifier, we can measure how well the structural information about that word in the sentence is encoded. Let us take the constituent type as an example, the phrase "is a global brand" in the sentence "The important thing about Disney is that it [is a global brand]." is a verb phrase (VP), so the probing classifier should predict the type "VP" when it is given the representation of "is a global brand". Usually, we use multi-layer perceptron (MLP) for probing classifier. Also, the parameters of encoder are frozen during the training of probing classifier. For more detailed description of probing classifier on different type of relations, we refer readers to the paper [26].

Originally, the edge probing is to investigate the role of words in the sentence in [26]. In [25], the scalar mixing technique [13] is combined with edge probing to explore the encoded information from different layers of the encoder. In particular, through this approach, we can tell which layer(s) are most relevant to a specific task. In formal, let $h^l = [h^l_0,\ h^l_1, ...,\ h^l_N]$ be the word representation after l-th layer of the encoder, then we can build a probing classifier (Fig. 4) using the first l layers:

$$h_{i,l} = \gamma_l \sum_{k=0}^{l} \alpha_l^k h_i^k,\ i = 1, ..., N$$

$$P_\tau^l = \text{MLP}([h_{0,l}, h_{1,l}, ..., h_{N,l}])$$

where $\alpha_l = softmax([\alpha^{(0)}, \alpha^{(1)}, ..., \alpha^{(l)}])$ and N is the sequence length. During the training, the parameters γ_l and α_l will be jointly learned. After the training of P_τ^l, we can use the learned parameters α_l to estimate the contribution of each layer for our task. If the weight $\alpha^{(i)}$ for the i-th layer is high, we can say that the i-th layer contains more relevant information about our task. In this work, we use the framework of edge probing to measure the learned knowledge in the layers of BERT model.

Investigation of the knowledge in the last layer

In the previous subsection, we demonstrate that the contribution of each layer in BERT can be learned through the weights for layers in a probing classifier. However, those weights are irrelevant to the training data distribution, which means we can not tell how many layers BERT needs to predict a specific instance correctly. This yields a question: what knowledge about the training data is only learned in the l-th layer, but not learned in the first (l-1) layer(s)?

Similar to the exploration in [25], the above problem can be addressed by training a series of probing classifier $\{P_\tau^l\}_{l=1}^L$, where L is the total layers of the encoder. The classifier P_τ^l utilizes all the information from the first l layers, while the knowledge in the first l-1 layers is employed in the classifier P_τ^{l-1}. Therefore, the difference of

Fig5a.png

(a) RNN on the last layer

Fig5b.png

(b) Attention on the last layer

Figure 5: Model architectures after incorporating all outputs from the last layer. In (a), we show both LSTM (only black in the RNN box) and biLSTM (both black and grey line in the RNN box).

performance score (F1 score) Δ^l_τ on test data between P^l_τ and P^{l-1}_τ stands for the learned knowledge only from the l-th layer:

$$\Delta^l_\tau = Score(P^l_\tau) - Score(P^{l-1}_\tau)$$

After calculating $\{\Delta^l_\tau\}^L_{l=1}$, we can acquire the distribution of the learned knowledge from the training data in each layer.

Our purpose is to investigate whether the last layer contains useful information for our tasks, so here we build the probing classifiers using the fine-tuned BioBERT model [17]. During training, we freeze the weights of BioBERT and only adjust the parameters of the probing classifiers. We show the score Δ^L_τ (measurement of knowledge) of the last layer of BioBERT for the datasets of our three tasks in Fig. 1. We can see that the last layer (12-th layer for the $\text{BERT}_{\text{base}}$ model) contains useful information in all cases and it is necessary to incorporate the discarded knowledge during fine-tuning.

Refined fine-tuning for the BERT model

We have shown that the fine-tuning method of only using classification token (CLS) discards some useful information in the last layer. Also, during the pre-training of BERT, the [CLS] token is only utilized in the next sentence prediction (NSP) task. This means that the [CLS] token might not encode the information about the interaction between entities because it is not designed to gain this type of information. In this subsection, we describe the mechanism of incorporating all the available information of the last layer in the BERT fine-tuning process. The proposed mechanism is called SLL fine-tuning: fine-tuning with information summarization in the last layer of BERT.

Our method includes two steps: 1) summarizing the ignored outputs in the last layer; 2) concatenating the summarized knowledge to the [CLS] output as final output from BERT. In step 1, we explore two types of methods of summarizing the information of the last layer: recurrent neural network (LSTM [32] and biLSTM [33]) and attention mechanism. In Fig. 5, we show the BERT model architectures after appending these two methods on the outputs of the last layer.

Formally, let H be the dimension of hidden states and L be the layer number of BERT model, then all the information in the last layer can be represented:

$$h^L_{all} = \{h^L_{CLS},\ h^L_1,\ h^L_2,\ ...,\ h^L_N,\ h^L_{SEP}\}$$

where h^L_{CLS} and h^L_{SEP} are the classification token output and separation token output, respectively. Previously, only h^L_{CLS} is used for classification problem during fine-tuning. Here we first summarize the discarded information in the last layer, i.e., $h^L = \{h_1^L,\ h_2^L,\ ...,\ h_N^L\}$, (the sentences separation token h_{SEP}^L is ignored here) using the RNN sequence model and attention mechanism:

1 The first choice of model to summarize a sequence is the recurrent neural network. Among RNN models, we choose LSTM because it handles the "long-term dependencies" better. Also, we utilize biLSTM to summarize the sequence in both forward and backward directions. We take the output of last LSTM unit as the representation of our sequence in LSTM. As for biLSTM, we just concatenate the outputs of first unit in backward direction and the last unit in forward direction as the final representation:

$$O = \begin{cases} LSTM(h_i^L) & (LSTM) \\ LSTM(\overrightarrow{h_i}^L) \oplus LSTM(\overleftarrow{h_i}^L) & (biLSTM) \end{cases}$$

2 Attention mechanism is another option of summarizing sequence data, which assigns a weight to each component. In particular, we employ the additive attention to summarize the sequence:

$$[\alpha_i] = softmax(h^L K)$$

$$O = \sum_{i=1}^{N} \alpha_i h_i^L$$

where $K^{H \times 1}$ are trainable parameters.

After the summarization, the output from RNN/attention mechanism is concatenated to the [CLS] output, and this combined output is the final output of BERT:

$$h = h_{CLS}^L \oplus O.$$

Then, we put a softmax layer on top of this representation to calculate the probability distribution on the labels:

$$p = softmax(W_f h + b_f)$$

where $W_f^{C \times H}$, $b_f^{C \times 1}$ are trainable parameters. C denotes the number of categories (classes) of our tasks.

In all the fine-tuning mechanisms we described so far, they all involve the use of [CLS] token. It is necessary to investigate the role of [CLS] token and measure its contribution on the classification tasks. Thus we experiment with the fine-tuning process in which [CLS] token is removed from the final output. In formal, we only use $h = [O]$ as the final output of the BERT model for the input. Then, the same softmax layer $p = softmax(W_f h + b_f)$ is utilized for predicting the relation type. The experiments using these two different fine-tuning methods can help interpret the roles of the [CLS] output and sentence outputs when the BERT model is applied on classification task.

Task	Label	Sentence examples
PPI	Positive	Nuclear protein @PROTEIN$ is a coactivator for the transcription factor @PROTEIN$.
	Negative	Their order of selection was @PROTEIN$ effusion, @PROTEIN$ serum, TNFalpha-effusion, and C3 effusion.
DDI	EFFECT	@DRUG$ may increase the ototoxic potential of other drugs such as aminoglycoside and some @DRUG$.
	MECHANISM	Cimetidine: @DRUG$ increases @DRUG$ plasma levels.
ChemProt	CPR:6	We conclude that @CHEMICAL$ and BAAM are competitive slowly reversible @PROTEIN$ antagonists on rat left atria.
	CPR:9	@PROTEIN$ plays a role in purine salvage by catalyzing the direct conversion of adenine to @CHEMICAL$.

Table 9: Pre-processed examples for the three tasks.

Combining the techniques for pre-training and fine-tuning

We have proposed techniques to improve the BERT model in both the pre-training and fine-tuning stage, so a natural idea is to combine these two techniques. For the new BERT model, we add an extra step for sub-domain adaptation in the pre-training and utilize the SLL mechanism in fine-tuning stage. In this way, we take the full advantage of the knowledge in the sub-domain data and the learned information in the last layer of BERT model.

Experiment setup

For the sub-domain adaptation, we train the BERT with 100K steps on the sub-domain data using maximum sequence length of 128, learning rate of 2e-5, and batch size of 192 in our experiments. In the sub-domain pre-training for BioBERT, we follow its settings in masked language model task and only utilize subwords masking. In contrast, we employ whole word masking in the pre-training of PubMedBERT.

For the BioBERT model fine-tuning, we use maximum sequence length of 128, training epoch of 10, learning rate of 2e-5, and batch size of 32. For the fine-tuning of PubMedBERT model, maximum sequence length of 256, training epoch of 10, learning rate of 2e-5, and batch size of 16 are utilized. In the edge probing experiments, we use a two-layer perceptron as the probing classifier, and 1,024 as the hidden layer unit number. During training of probing classifiers, we use learning rate of 2e-5 and training epoch of 4.

Data pre-processing

As illustrated before, one relation extraction instance includes two components: the text (sentence) and the entities in it. With the aim of making BERT model recognize the position of the biomedical entities, we follow the standard pre-processing step for relation extraction to replace the entity names with some predefined tags. For example, the protein names are replaced with the tag “@PROTEIN$”. Similarly, we replace the drug names and the chemical names with “@DRUG$” and “@CHEMICAL$”, respectively. In Table 9, we give some pre-processed instances from the corpora.

Conclusion

In this paper, we proposed two techniques to improve the pre-training and fine-tuning of BERT model. We first utilized the sub-domain data in the pre-training phase of BERT model to help its generalization on different tasks. Then we proposed a refined fine-tune process to utilize all the knowledge in the last layer of BERT model. In addition, we explored the combination of those two techniques. We demonstrated that the proposed methods are effective on three widely studied relation extraction tasks. Furthermore, the experiment results showed that the proposed methods achieve better performance on all three tasks and we achieved state-of-the-art performance on three relation extraction benchmark sets. In the future, we will apply our method on other relation extraction tasks in biomedical domain.

Acknowledgements

We would like to thank Ziqing Luo for the help of proofreading the manuscript and the suggestions. Also, Peng Su's PhD studies were supported by a supplement of NIH grant U01GM125267. In addition, portions of this manuscript were presented at the proceeding of 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) [34].

Funding

Not applicable.

Abbreviations

BERT: Bidirectional Encoder Representations from Transformers
RE: Relation Extraction
PPI: Protein-protein Interactions
DDI: Drug-drug Interactions
ChemProt: Chemical-protein Interactions
P/G: Protein or Gene
D: Drug
CP: Chemical and Protein
NLP: Natural Language Processing
RNN: Recurrent Neural Network
LSTM: Long Short-term Memory
BiLSTM: Bi-directional Long Short-term Memory
SLL: Summarization in the last layer
MLM: Masked Language Model
NSP: Next Sentence Prediction
CLS: Classification
Att: Attention mechanism
LL: Last Layer
P: Precision
R: Recall
F: F1 score
SOTA: State-of-the-art

Availability of data and materials

The data and code are publicly available at: https://github.com/udel-biotm-lab/BERT-RE

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Consent for publication

Not applicable.

Authors' contributions

PS and KVS designed the project; PS conducted the experiments and wrote the initial manuscript draft. PS and KVS performed the result analysis and finalized the manuscript. Both authors read and approved the final manuscript.

Authors' information

Affiliation
Biomedical text mining lab, University of Delaware, Newark, USA

Author details

Department of Computer and Information Science, University of Delaware, Newark, USA.

References

1. Krallinger, M., Leitner, F., Rodriguez-Penagos, C., Valencia, A.: Overview of the protein-protein interaction annotation extraction task of biocreative ii. Genome biology **9**(2), 4 (2008)
2. Herrero-Zazo, M., Segura-Bedmar, I., Martínez, P., Declerck, T.: The ddi corpus:f an annotated corpus with pharmacological substances and drug–drug interactions. Journal of biomedical informatics **46**(5), 914–920 (2013)
3. Krallinger, M., Rabal, O., Akhondi, S.A., *et al.*: Overview of the biocreative vi chemical-protein interaction track. In: Proceedings of the Sixth BioCreative Challenge Evaluation Workshop, vol. 1, pp. 141–146 (2017)
4. Van Harmelen, F., Lifschitz, V., Porter, B.: Handbook of Knowledge Representation. Elsevier, ??? (2008)
5. Macherey, K., Och, F.J., Ney, H.: Natural language understanding using statistical machine translation. In: Seventh European Conference on Speech Communication and Technology (2001)
6. Hirschman, L., Gaizauskas, R.: Natural language question answering: the view from here. natural language engineering **7**(4), 275 (2001)
7. Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04), pp. 423–429 (2004)
8. Sierra, G., Alarcón, R., Aguilar, C., Bach, C.: Definitional verbal patterns for semantic relation extraction. Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication **14**(1), 74–98 (2008)
9. Sahu, S.K., Anand, A.: Drug-drug interaction extraction from biomedical texts using long short-term memory network. Journal of biomedical informatics **86**, 15–24 (2018)
10. Zhang, H., Guan, R., Zhou, F., Liang, Y., Zhan, Z.-H., Huang, L., Feng, X.: Deep residual convolutional neural network for protein-protein interaction extraction. IEEE Access **7**, 89354–89365 (2019)
11. Su, P., Li, G., Wu, C., Vijay-Shanker, K.: Using distant supervision to augment manually annotated data for relation extraction. BioRxiv, 626226 (2019)
12. Dai, A.M., Le, Q.V.: Semi-supervised sequence learning. In: Advances in Neural Information Processing Systems, pp. 3079–3087 (2015)
13. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
14. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
15. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
16. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
17. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**(4), 1234–1240 (2020)
18. Peng, Y., Yan, S., Lu, Z.: Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. arXiv preprint arXiv:1906.05474 (2019)
19. Beltagy, I., Cohan, A., Lo, K.: Scibert: Pretrained contextualized embeddings for scientific text. arXiv preprint arXiv:1903.10676 (2019)
20. Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., Poon, H.: Domain-specific language model pretraining for biomedical natural language processing. arXiv preprint arXiv:2007.15779 (2020)
21. Johnson, A.E., Pollard, T.J., Shen, L., Li-wei, H.L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L.A., Mark, R.G.: Mimic-iii, a freely accessible critical care database. Scientific data **3**, 160035 (2016)
22. Ammar, W., Groeneveld, D., Bhagavatula, C., Beltagy, I., Crawford, M., Downey, D., Dunkelberger, J., Elgohary, A., Feldman, S., Ha, V., et al.: Construction of the literature graph in semantic scholar. arXiv preprint arXiv:1805.02262 (2018)
23. Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., Smith, N.A.: Don't stop pretraining: Adapt language models to domains and tasks. arXiv preprint arXiv:2004.10964 (2020)
24. Phang, J., Févry, T., Bowman, S.R.: Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. arXiv preprint arXiv:1811.01088 (2018)
25. Tenney, I., Das, D., Pavlick, E.: Bert rediscovers the classical nlp pipeline. arXiv preprint arXiv:1905.05950 (2019)
26. Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R.T., Kim, N., Van Durme, B., Bowman, S.R., Das, D., et al.: What do you learn from context? probing for sentence structure in contextualized word representations. arXiv preprint arXiv:1905.06316 (2019)
27. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
28. Bunescu, R., Ge, R., Kate, R.J., Marcotte, E.M., Mooney, R.J., Ramani, A.K., Wong, Y.W.: Comparative experiments on learning information extractors for proteins and their interactions. Artificial intelligence in medicine **33**(2), 139–155 (2005)
29. Song, Y., Wang, J., Liang, Z., Liu, Z., Jiang, T.: Utilizing bert intermediate layers for aspect based sentiment analysis and natural language inference. arXiv preprint arXiv:2002.04815 (2020)
30. Porter, M.F., *et al.*: An algorithm for suffix stripping. Program **14**(3), 130–137 (1980)
31. Wei, C.-H., Kao, H.-Y., Lu, Z.: Pubtator: a web-based text mining tool for assisting biocuration. Nucleic acids research **41**(W1), 518–522 (2013)
32. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
33. Graves, A., Fernández, S., Schmidhuber, J.: Bidirectional lstm networks for improved phoneme classification and recognition. In: International Conference on Artificial Neural Networks, pp. 799–804 (2005). Springer
34. Su, P., Vijay-Shanker, K.: Investigation of bert model on biomedical relation extraction based on revised fine-tuning mechanism. In: 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp.

2522–2529 (2020). IEEE

Figures

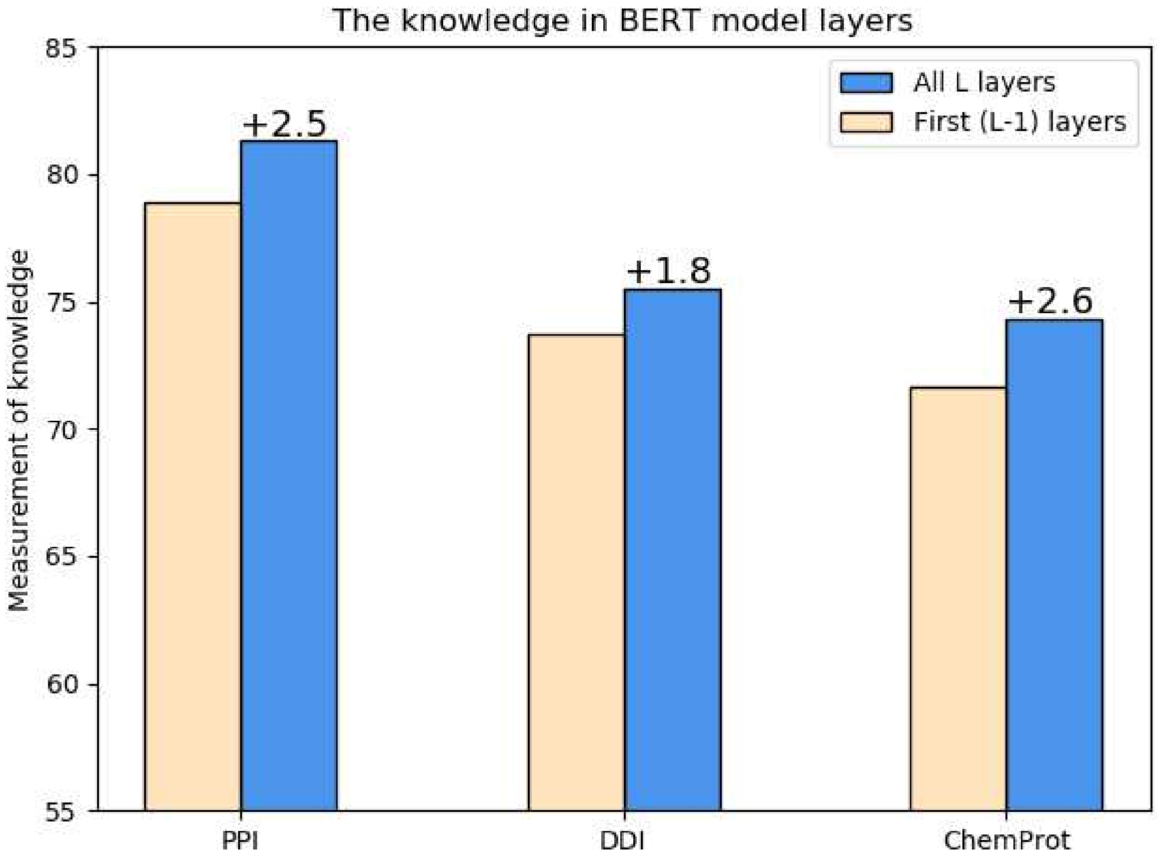

Figure 1: Learned knowledge of training data in the layers of BioBERT. L is the total layers of BERT model. "Measurement of Knowledge" (Δ_{τ}^{L}) is defined in the Methods section.

Figure 1

See image above for figure legend

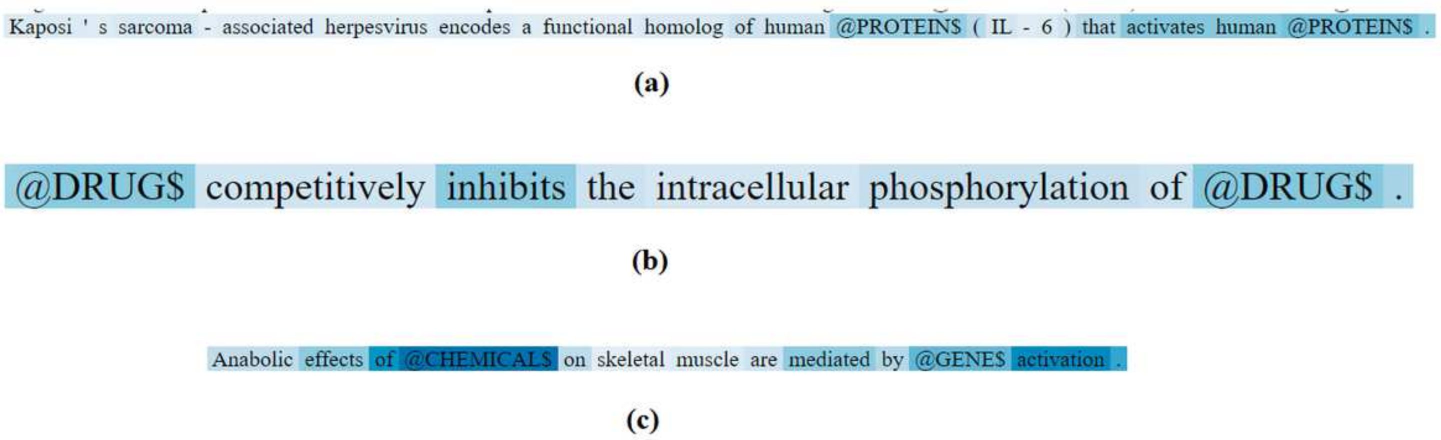
Kaposi ' s sarcoma - associated herpesvirus encodes a functional homolog of human @PROTEIN$ (IL - 6) that activates human @PROTEIN$.
(a)
@DRUG$ competitively inhibits the intracellular phosphorylation of @DRUG$.
(b)
Anabolic effects of @CHEMICAL$ on skeletal muscle are mediated by @GENE$ activation .
(c)

Figure 2

(a) PPI example (Label: Positive). (b) DDI example (Label: EFFECT). (c) ChemProt example(Label: CPR:3). The visualization of attention weights in the last layer.

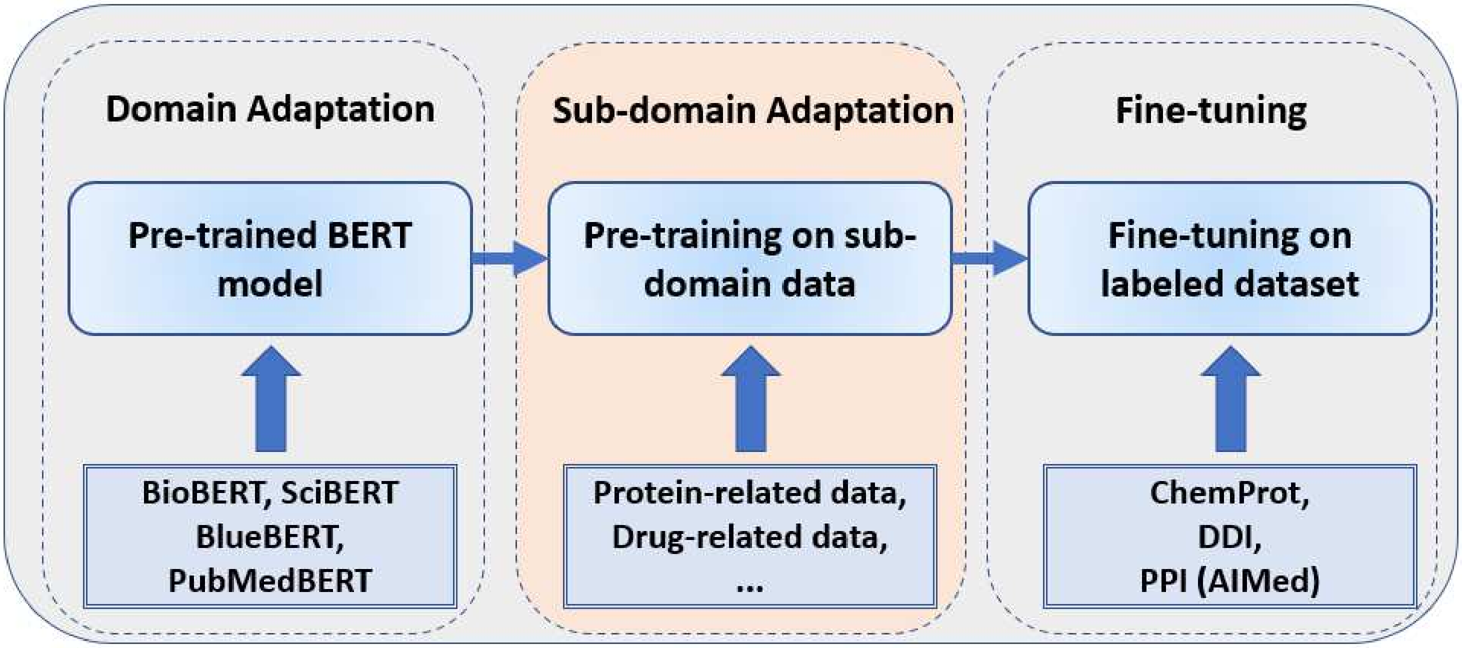
Domain Adaptation
Sub-domain Adaptation
Fine-tuning
Pre-trained BERT model
Pre-training on sub-domain data
Fine-tuning on labeled dataset
BioBERT, SciBERT BlueBERT, PubMedBERT
Protein-related data, Drug-related data, ...
ChemProt, DDI, PPI (AIMed)

Figure 3

BERT model training process with sub-domain adaptation.

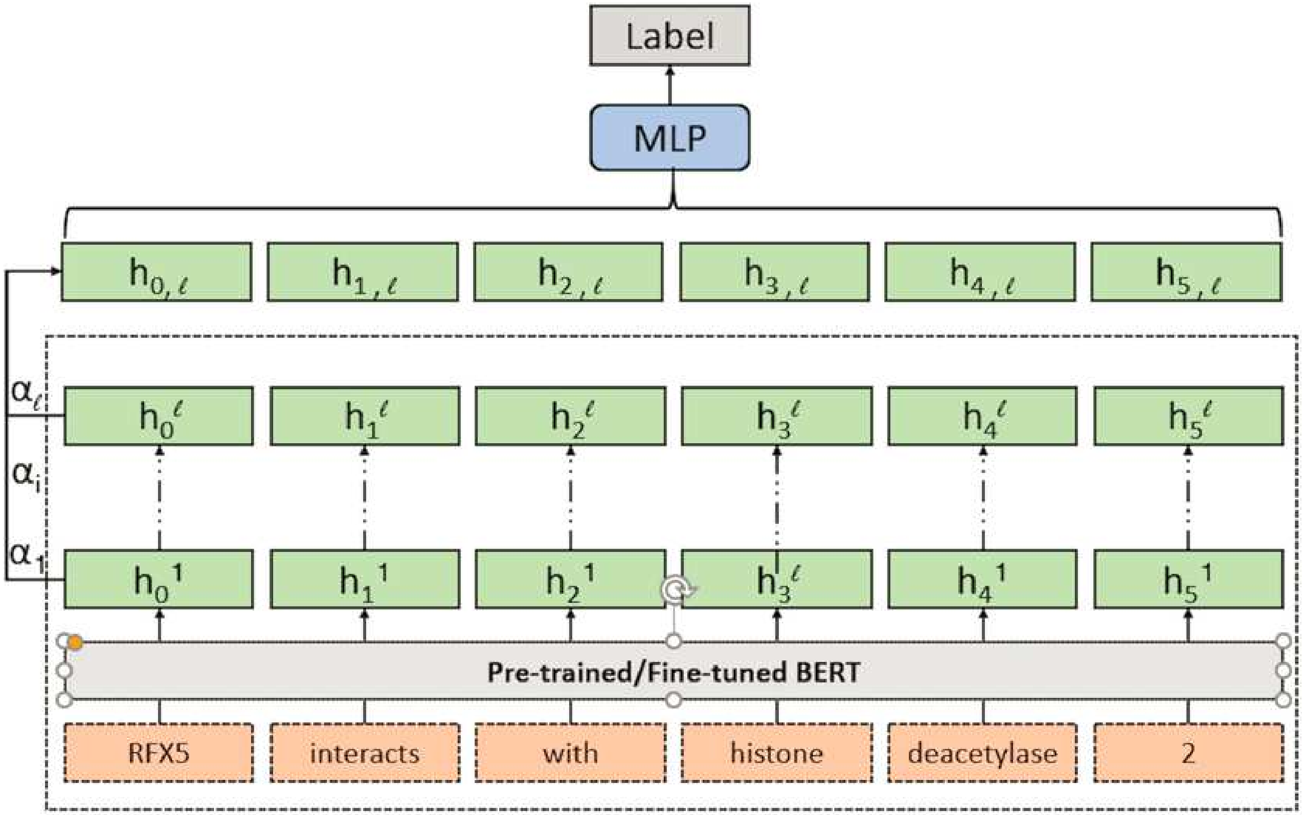

Figure 4: Probing classifier architecture. We freeze the parameters of BERT model during the training of probing classifier. Through the learned α, we can know the relevance between each layer and the task. Also, we can tell which layer learns the knowledge for a specific instance by building a series of probing classifier $\{P_\tau^l\}_{l=1}^L$. For the relation extraction instance "RFX5 interacts with histone deacetylase 2", if the probing classifier P_τ^l predicts the interacting relationship between proteins "RFX5" and "histone deacetylase 2" correctly using the information of the first l layers, but P_τ^{l-1} does not predict correctly using the information of the first (l-1) layers. We can say that the knowledge about this instance is learned in the l-th layer.

Figure 4

See image above for figure legend

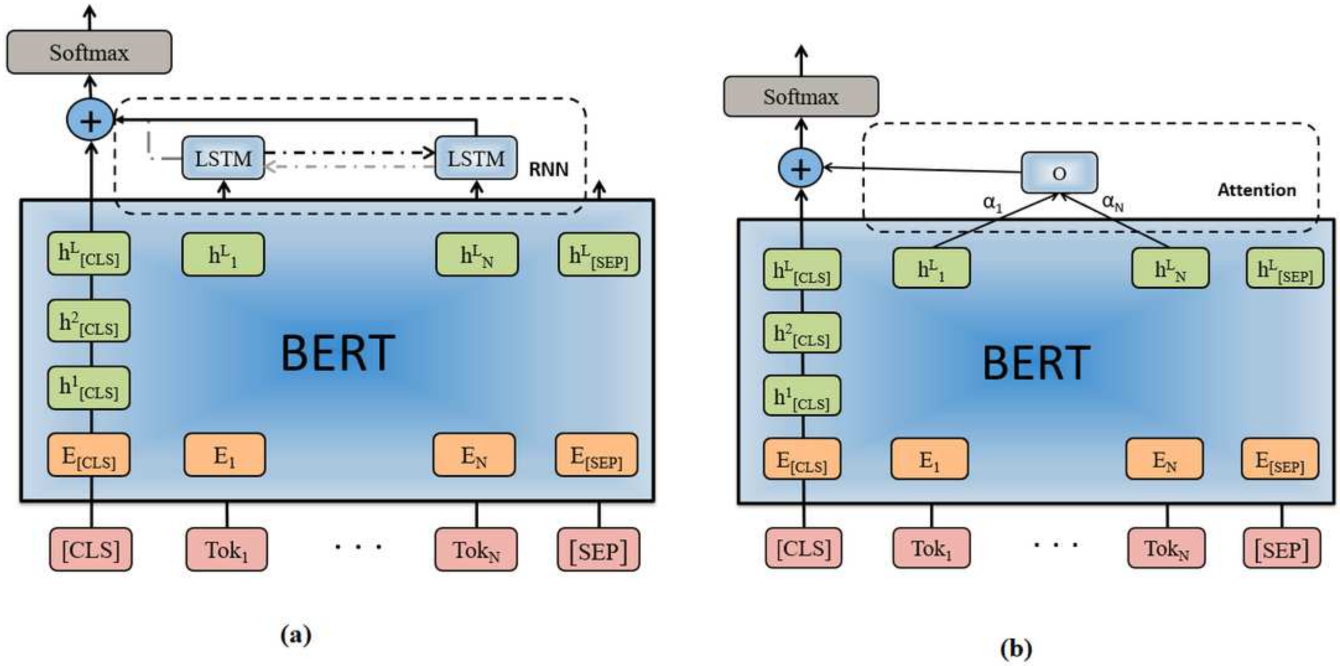

Figure 5

(a) RNN on the last layer (b) Attention on the last layer Model architectures after incorporating all outputs from the last layer. In (a), we show both LSTM (only black in the RNN box) and biLSTM (both black and grey line in the RNN box).