VART: Vocabulary Adapted BERT Model for Multi-label Document Classification

Zhongguang Zheng
Fujitsu R&D Center
zhengzhg@fujitsu.com

Lu Fang
Fujitsu R&D Center
fanglu@fujitsu.com

Yiling Cao
Fujitsu R&D Center
caoyiling@fujitsu.com

Jun Sun
Fujitsu R&D Center
sunjun@fujitsu.com

Abstract

Large scale pre-trained language models (PTLMs) such as BERT have been widely used in various natural language processing (NLP) tasks, since PTLMs greatly improve the downstream task performances by fine-tuning the parameters on the target task datasets. However, in many NLP tasks, such as document classification, the task datasets often contain numerous domain specific words which are not included in the vocabulary of the original PTLM. Those out-of-vocabulary (OOV) words tend to carry useful domain knowledge for the downstream tasks. The domain gap caused by OOV words may limit the effectiveness of PTLM. In this paper, we present VART, a concise pre-training method to adapt BERT model by learn OOV word representations for multi-label document classification (MLDC) task. VART employs an extended embedding layer to learn the OOV word representations. The extended layer can be pre-trained on the task datasets with high efficiency and low computational resource. The experiments for MLDC task on three datasets from different domains with different sizes demonstrate that VART consistently outperforms the conventional PTLM adaptation methods such as fine-tuning, task adaption and other pre-trained model adaptation methods.

1 Introduction

Pre-trained language models (PTLMs) such as GPT (Radford and Narasimhan, 2018) and BERT (Devlin et al., 2018), which are trained on massive unlabeled datasets, can effectively encode rich knowledge into huge parameter spaces. Therefore, by fine-tuning the PTLM parameters, the encoded knowledge is able to benefit a wide range of downstream natural language processing (NLP) tasks (Dai et al., 2021; Liang et al., 2020; Adhikari et al., 2019; Wang et al., 2019; Zhu et al., 2020; Gao et al., 2019).

However, applying PTLMs to the specific domain tasks always faces the domain gap problem. Conventionally, PTLMs are trained on a large volume of general domain datasets with a fixed vocabulary extracted from the datasets. When applying such general domain PTLMs on a specialized domain dataset, e.g., patent documents, the domain gap becomes an important factor that hinders the performance of PTLMs. One of the causes of the domain gap is the domain words which are not included in the PLTM vocabulary. Although a PTLM is capable to handle the out-of-vocabulary (OOV) words by splitting each OOV word into multiple in-vocabulary sub-words, for instance, the word “chalcogenide” commonly seen in the patent document will be split into five sub-words including “ch”, “##al”, “##co”, “##gen” and “##ide” with the vocabulary of BERT-base-cased model. As a result, the representation of “chalcogenide” is divided into five embedded vectors. Consequently, the information of “chalcogenide” which is intuitively preferable as an integral representation would be ignored in the downstream tasks. Since those OOV words tend to carry rich domain knowledge, the information loss potentially limits the effectiveness of PTLMs for tasks such as multi-label document classification (MLDC), which is known as a fundamental and essential NLP task and has been widely applied in specific domain tasks such as clinical code prediction (Scheurwegs et al., 2017; Mullenbach et al., 2018) for electronic health record (EHR) texts and biomedical document classification (Du et al., 2019; Baker and Korhonen, 2017). The datasets for domain-specific MLDC tasks always contain a large number of domain OOV words which challenge the use of PTLMs.

In order to bridge the domain gap for PTLMs, a plenty of researches have been conducted. Some of the prior researches have focused on fine-tuning PTLMs for text classification (Sun et al., 2019), while other researches have addressed the issue by adapting the PTLMs to the target domain datasets by training PTLMs from scratch with a new vocabulary, such as SciBERT (Beltagy et al., 2019). However, the fine-tuning approach is focused on adapting the PTLM parameters to the target domain and leaves aside the OOV issue, while training PTLMs from scratch sets an extremely high demand of computational resource and it is time-consuming. To solve the domain OOV problem efficiently, there are recent researches focusing on extending the original PTLM vocabulary with target domain words, such as exBERT (Tai et al., 2020), which complements the original BERT model with another BERT model for learning the OOV representations. However, exBERT model still faces the problems of efficiency and training complexity.

Inspired by the work of exBERT, we propose VART, a **V**ocabulary **A**dapted BE**RT** model which adapts the original BERT model by extending the vocabulary with the target domain vocabulary. Specifically, we just extend the embedding layer of BERT model to learn the OOV word representations while inheriting other BERT layers and then pre-train the model on the downstream task datasets. Comparing with exBERT, the main contributions of this work are summarized as follows:

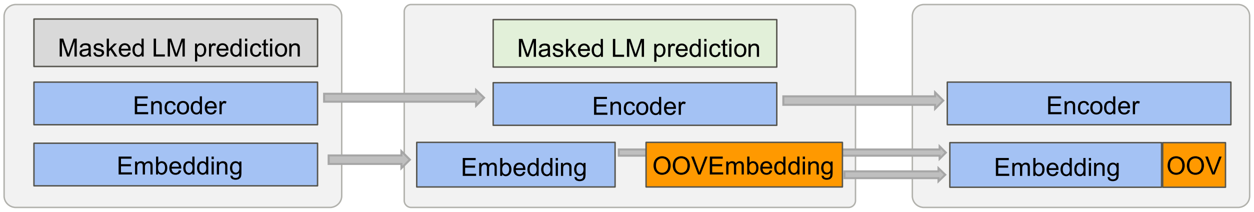
- We demonstrate a concise training method to extend the BERT vocabulary with domain OOV words. VART overcomes the problems that remained in exBERT by boosting the efficiency in both pre-training and fine-tuning phases while saving computational resources.

- Extensive experiments are conducted on three datasets with different sizes and domains for MLDC task. Although smaller in model size, VART consistently outperforms exBERT and other baseline methods even on an extremely small scale task dataset.

2 Related Work

Fine-tuning PTLMs. The most common conventional approach for utilizing PTLMs is fine-tuning. Generally, fine-tuning is performed by replacing the output layer of a PTLM with other layers which are specified according to the downstream tasks. The parameters of the original PTLM are preserved and tuned on the task datasets. Various fine-tuning methods of BERT specially on document classification task are investigated in (Sun et al., 2019), such as studying the effectiveness of different BERT layers in the fine-tuning phase. Besides, a multi-task learning mechanism is also used to fine-tune the BERT model. Rietzler et al. (2019) fine-tunes the BERT model for sentiment classification task. Different adaptation scenarios such as in-domain, cross-domain and joint-domain are studied in the experiments. Gururangan et al. (2020) focuses on dataset selection for further pre-training the RoBERTa (Liu et al., 2019) model. Various dataset selection strategies, such as domain dataset selection (domain-adaptive pre-training, DAPT) and task dataset selection (task-adaptive pre-training, TAPT) are proposed. The experiment results demonstrate that an adapted PTLM is beneficial to various downstream NLP tasks.

Domain specific PTLMs. In order to further improve the performances on domain specific tasks, such as biomedical domain, researches focusing on training domain specific PTLMs have been proposed in recent years. SCIBert (Beltagy et al., 2019) leverages pre-training on large multi-domain scientific publications with a new in-domain vocabulary. The experiment shows that SCIBert outperforms general domain BERT in the scientific domain tasks. Moreover, the in-domain vocabulary is proven helpful in the experiment. Gu et al. (2020) pre-trains the domain BERT model from scratch with a customized vocabulary on the PubMed articles. BioBERT (Lee et al., 2019) inherits the vocabulary and the model



(a) Original BERT

(b) Pre-training B_{ext} with V_{ext} on target dataset

(c) VART

Figure 1: Overview of adapting BERT model by extending the vocabulary. We will reuse the original BERT encoder layer and further pre-train it with an extended vocabulary V_{ext} from the target dataset. While in the pre-training phase, we train the OOVEmbedding layer and MLM Prediction layer from scratch. Finally, the VART model is constructed by merging the two embedding layers into one.

parameters from original BERT model and then is further pre-trained on a large volume dataset that mainly consisted of PubMed articles. Similarly, Lee and Hsiang (2020) obtains a BERT model in patent domain by fine-tuning original BERT model on over two million patent documents.

Extending PTLM vocabulary. It is known that training PTLMs from scratch requires powerful computational resources and is time-consuming. Recently proposed exBERT (Tai et al., 2020) introduces a general training method to extend the original BERT from the general domain to a specific domain. The exBERT model preserves the original BERT model and vocabulary. Meanwhile, another smaller (or full size) BERT model is used to learn the information of domain OOV words. However, there are mainly two problems of this dual BERT model structure. Firstly, exBERT is much larger than a single BERT model so that it is highly inefficient in pre-training phase. For alleviating the problem, the author proposes tradeoffs such as shrinking the size of the extra BERT model and fixing the parameters of the original BERT model in the pre-training phase. However, the smaller size BERT model may be inadequate to learn document representations, while the original BERT model still faces the domain gap if the parameters are fixed. Secondly, it is non-trivial to combine the outputs from the two BERT encoders. For this reason, a weighting block comprised of a fully-connected layer and sigmoid activator is used to combine the two encoder outputs. This increases the training complexity.

The fine-tuning PTLMs method focuses on the model parameter adaptation and leaves aside the OOV issue. Training domain specific PTLMs from scratch is computational expensive and time-consuming. Inspired by exBERT, we propose VART to adapt the original BERT model to the target domain by learning the representations of domain OOV words. In order to solve the remained problems in exBERT, we use only one single BERT model with a minor modification for the adaptive pre-training. Comparing with exBERT, our model boosts the efficiency in both pre-training and fine-tuning phases without sacrificing the performances for MLDC task.

3 Methodology

In this section, we introduce VART model which is pre-trained with an extra OOV list from the target dataset illustrated in Figure 1. In the first place, we would like to define the PTLM adaptation task in this paper. Given an original pre-trained BERT model B with a vocabulary V_{in} and a training dataset D_t for MLDC in a specific domain, we expand V_{in} to V_{ext} with an OOV list extracted from D_t at first, and then design an extended BERT model B_{ext}, which is inherited from B and is further pre-trained with V_{ext}. Finally, VART is derived from B_{ext} and is fine-tuned for downstream MLDC task. In the rest of this paper, the term “BERT” refers to “BERT-base-cased”[1] model from Huggingface[2] repository and is implemented by Huggingface transformers[3] (Wolf et al., 2020).

[1] https://huggingface.co/bert-base-cased

[2] https://huggingface.co/

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

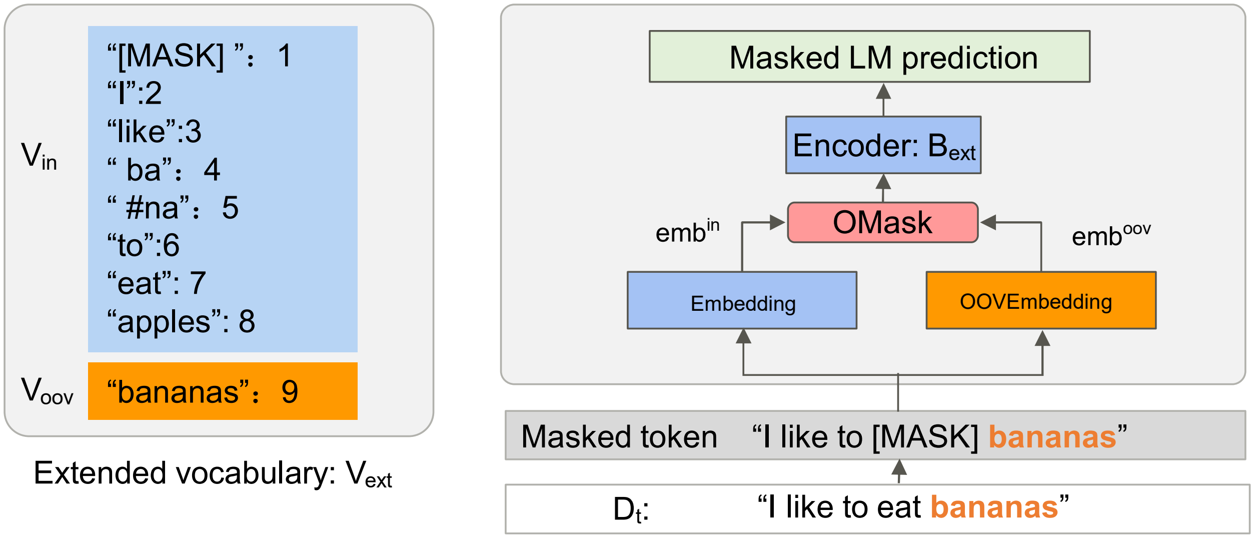


Figure 2: Overview of further pre-training B_{ext} with extended vocabulary and target dataset D_t. The $OMask$ (OOV Mask) selects and combines embeddings emb^{in} from the original Embedding layer for the existing vocabularies and emb^{oov} from OOVEmbedding layer for the new vocabularies.

OOV Extraction. Given a training dataset D_t of a specific task (MLDC for this paper), we first extract a word list V_t from D_t via WordPiece (Wu et al., 2016) algorithm, then an OOV vocabulary V_{oov} is constructed by selecting all the terms in V_t but not in V_{in}. Afterwards, V_{ext} is obtained by appending V_{oov} to V_{in}. As the example shown in Figure 2, we extract a new word "bananas" from D_t and append the word to V_{in}.

Pre-training VART. The structure of original BERT model can be categorized mainly into three components shown in Figure 1(a). The embedding layer encodes the input tokens into a real-valued vector. The encoder is a stack of Transformer (Vaswani et al., 2017) blocks consist of multiple self-attention heads and learns the final representation of the input sequence. The task layer generates the output for self-supervised tasks such as masked language model (MLM) and next sentence prediction (NSP).

In order to learn the representations for the OOV words, we simply complement an extra embedding layer to the original BERT model B while preserving the original embedding layer to encode the in-vocabulary words. As the example shown in Figure 2, by applying $OMask$, only the new word "bananas" is encoded by the OOVEmbedding layer. Afterwards, it is handy to use $OMask$ for combining the embedded vectors emb^{in} and emb^{oov}. This process is described in the following Equations, where w_i is id of the i-th word in the input sequence.

$$\text{emb}_i^{\text{in}} = \text{Embedding}((1-\text{OMask}_i)\cdot w_i) \quad (1)$$

$$\text{emb}_i^{\text{oov}} = \text{OOVEmbedding}(\text{OMask}_i \cdot w_i) \quad (2)$$

$$\text{emb}_i = (1-\text{OMask}_i)\text{emb}_i^{\text{in}} + \text{OMask}_i\text{emb}_i^{\text{oov}} \quad (3)$$

The encoder of B_{ext} is initialized with the encoder in B. For the task layer, we only select MLM to pre-train B_{ext} in this work. Considering that the vocabulary size of V_{ext} is enlarged, the magnitude of prediction vector from the task layer should be increased to match the size of V_{ext}. Therefore, we create a new task layer with the proper dimension for pre-training B_{ext} on D_t. Considering that the extra embedding layer and the task layer in B_{ext} need to be trained from scratch, while the encoder layer is inherited from B, it is plausible to set different learning rates for those layers in the pre-training process. Afterwards VART model is obtained by concatenating the two embedding layers after pre-training B_{ext}(see Figure1(c)). At this point, we adapt B to the target dataset D_t by extending the vocabulary.

The final transformation brings another merit of VART in that our model is converted into a standard

Table 1: Datasets overview. The column "Len" denotes the average document length of each dataset. "Labels" is the actual label number of the dataset.

Dataset	**Train**	**Test**	**Len**	**Labels**	**Domain**
CEI	2.4k	1.2k	277	103	Clinical
EU-Leg	45k	6k	538	4193	Law
Patent	90k	10k	66	9152	Patent

BERT model from the non-standard structured B_{ext}. A standard structured BERT model is much more friendly to people who would deploy VART in their own applications since they do not need to import any extra libraries in their project reducing the risk of library conflicts.

4 Experiment

4.1 Datasets

We conduct experiments on three datasets for the MLDC task with different sizes and domains. An overview of all the datasets is shown in Table 1.

CEI dataset[4] (Larsson et al., 2017) is annotated for chemical exposure assessments[5]. The dataset contains 3.7k abstracts from PubMed documents and is categorized by experts into 32 classes denoting chemical exposure information.

EU-Leg dataset (Chalkidis et al., 2019) comprises 57k legislative documents from EURLEX[6]. The documents are annotated with multiple concepts from EUROVOC[7] and contain about 4.3k labels in total.

Patent dataset[8] (Huang et al., 2019) is collected from USPTO[9] containing 100k patent documents, including titles and abstracts. The hierarchical annotation category contains almost 9k labels.

4.2 Implementation

Table 2: Learning rate settings for pre-training different models. "BERT$_{tapt}$" denotes the task adapted training for BERT model. The subscript of exBERT and VART models denotes the dataset on which the learning rate is applied.

Models	**Original Layers**	**Extended Layers**
BERT$_{tapt}$	4e-05	–
exBERT$_{other}$	2e-05	1e-04
exBERT$_{CEI}$	4e-05	1e-04
VART$_{other}$	4e-05	1e-04
VART$_{EU\_Leg}$	2e-05	2e-04

Pre-training. We use the exBERT library[10] and modify the training script to support the VART model. The BERT-base-cased model, which contains 12 Transformer layers with 12 self-attention heads and 768 hidden dimension, is selected as the original BERT model. The maximum input sequence length is set to 512 and only the masked language model task is chosen to pre-train all the models.

For the extended encoder in exBERT model, we inherit the best settings in the author's work with hidden size 252 and feed-forward layer size 1024 (about 33% of the original BERT model size). Different learning rates are set for the original BERT layers and the extended layers in both exBERT and VART models. The detailed learning rate settings are listed in Table 2. We pre-train all the models for 50 epochs on the CEI dataset, 40 epochs on the Patent dataset and 10 epochs on the EU-Leg dataset in considering of training efficiency. The models saved after the final epoch will be used for the MLDC task. The batch size is set to 4 on all the datasets. The Adam (Kingma and Ba, 2015) optimizer is applied to tune the parameters. All the experiments hereafter are conducted on our in-house servers with GeForce GTX 1080 Ti/2080 Ti GPUs.

Fine-tuning. In the fine-tuning process, the hidden state h of the first token [CLS] is considered as the document representation and is followed by a fully connected layer (task layer) which predicts the final labels. Different learning rates are set to the pre-trained models and the task layer respectively. 2e-05 is set to all the pre-trained models, while 2e-04 is set to the task layer on CEI dataset and 1e-04 is set on Patent and EU-Leg datasets. Moreover, a

[4] https://figshare.com/articles/dataset/Corpus_and_Software/4668229

[5] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5336247/

[6] https://eur-lex.europa.eu/

[7] https://publications.europa.eu/en/web/eu-vocabularies

[8] https://drive.google.com/open?id=1So3unr5p_vlYq31gE0Ly07Z2XTvD5QlM

[9] https://www.uspto.gov/

[10] https://github.com/cgmhaicenter/exBERT

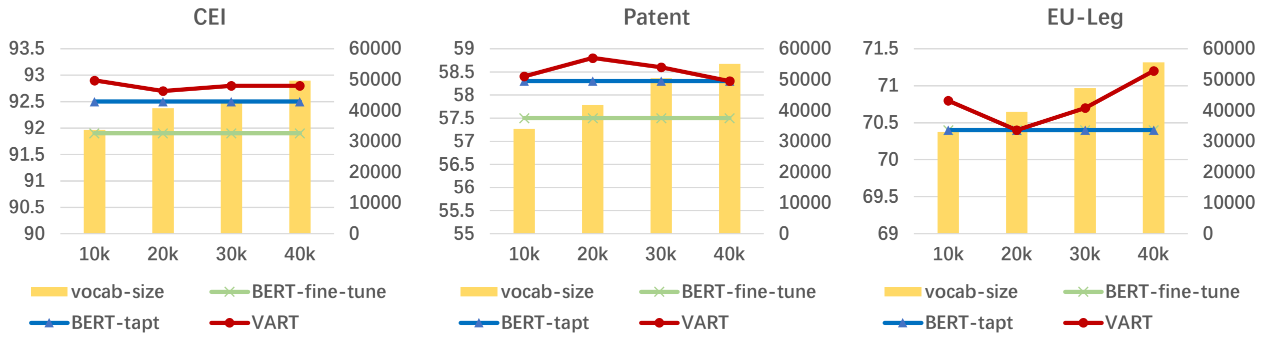


Figure 3: Visualization of results from different vocabularies. The line chart denotes the micro-F1 score corresponding to the left vertical axis and the histogram shows the vocabulary size corresponding to the right vertical axis.

Table 3: Overall experiment results. "Vocab Size" denotes the vocabulary size for training exBERT and VART models. For BERT models, the vocabulary size is 28,996.

Model	**CEI**	**EU-Leg**	**Patent**
BiGRU-LWAN	–	69.8%	–
HARNN	–	–	54.1%
BERT$_{fine\_tune}$	91.9%	70.4%	57.5%
BERT$_{tapt}$	92.6%	70.4%	58.3%
exBERT$_{ext}$	92.1%	**71.5%**	58.4%
exBERT$_{whole}$	92.8 %	70.8%	58.6%
VART	**92.9%**	71.2%	**58.8%**
Vocab Size	33,710	55,558	41,718

learning rate decay mechanism is adopted to boost the performance in the form of Equation 4, where α_0 is the initial learning rate and $decay\_rate$ is set to 0.9. We run 20 epochs for fine-tuning on all the datasets and the binary cross-entropy loss is used to train the classifier. Micro-F1 score is adopted as the evaluation metric.

$$\alpha' = \frac{1}{1 + decay\_rate \times epoch\_num} \alpha_0 \quad (4)$$

4.3 Baseline Methods

Networks without pre-trained models. HARNN (Huang et al., 2019) is a hierarchical attention-based RNN model. BiGRU-LWAN (Chalkidis et al., 2019) adopts a label-wise attention mechanism on the basis of a Bi-GRU layer. The two models carried out experiments on the same datasets as in this work and without pre-trained models. We cite the reported results on Patent and EU-Leg datasets from the works directly.

Table 4: Comparison of different vocabularies. Scores in bold denote the results better than Bert$_{tapt}$. The best scores are marked with †. The number in brackets presents the vocabulary size.

Settings	**CEI**	**EU-Leg**	**Patent**
Bert$_{fine-tune}$	91.9% (28,996)	70.4% (28,996)	57.5% (28,996)
Bert$_{tapt}$	92.6% (28,996)	70.4% (28,996)	58.3% (28,996)
VART$_{10k}$	**92.9**†**%** (33,710)	**70.8%** (33,000)	**58.4%** (34,008)
VART$_{20k}$	**92.7%** (40,726)	70.4% (39,507)	**58.8%**† (41,718)
VART$_{30k}$	**92.8%** (42,659)	**70.7%** (47,254)	**58.6%** (50,456)
VART$_{40k}$	**92.8%** (49,673)	**71.2%**† (55,558)	58.3% (55,075)

Fine-tune Only. The original BERT model is directly fine-tuned for the downstream MLDC task.

Task Adaptation. We follow the task adaptation (TAPT) method described in (Gururangan et al., 2020) to pre-train the BERT model on each training dataset with the vocabulary unchanged at first, and then fine-tune the adapted BERT model for the MLDC task.

exBERT. We inherit the best settings for the extended encoder in the author's work with hidden size 252, feed-forward layer size 1024, 12 attention heads and 12 hidden layers (about 33% of the orig-

Table 5: Efficiency and computational resource cost comparison of the pre-trained models.

		BERT$_{tapt}$	**exBERT**	**VART**
Parameter size		110M	147M	122M
FLOPs$_{pre\_train}$	CEI	55.2B	75B	60.6B
	Patent		74.6B	60.2B
	EU-Leg		80.1B	65.6B
FLOPs$_{fine\_tune}$	CEI	43.5B	57.9B	43.5B
	Patent			
	EU-Leg			
GPU usage		1	2	1

inal BERT model size). The exBERT model is pre-trained based on the same vocabulary with VART model. Moreover, there are two pre-training modes for exBERT: training the extended model only and training the whole model. We test the both pre-training modes in this work.

4.4 Experiment Results and Analysis

The overall experiment results are listed in Table 3. From the results, we can observe that the fine-tuning method greatly improves the performances with respect to the networks without BERT model. This confirms the effectiveness of BERT model, not surprisingly. Task adapted BERT model further improves the performance of fine-tuning method, which demonstrates that the adaptation of PTLMs is essential to improve the performance for specific tasks. VART achieves best scores on CEI and Patent datasets. Although exBERT produces the best result on EU-Leg dataset with training extended encoder mode, the result from VART on EU-leg is close to exBERT$_{ext}$ and is better that other baselines including exBERT$_{whole}$.

From the results we can also see the problems of exBERT. Firstly, the two training modes perform differently on different datasets. Moreover, the results of the two modes are also significantly different. On the CEI dataset, exBERT$_{whole}$ is significantly better than exBERT$_{ext}$, while the result is opposite on EU-Leg dataset. This indicates that it is difficult for exBERT to balancing the performances of the original BERT encoder and the extended encoder. On the contrary, VART model can be easily trained as a result of its simple structure.

Secondly, the training efficiency hinders the utilization of exBERT. Table 5 lists the comparison of the training efficiency and the computational resource cost between exBERT and VART. We can see that, although exBERT yields equivalent results on CEI and Patent datasets and produces the best result on EU-Leg dataset, VART is more efficient than exBERT in both pre-training and fine-tuning. For instance, as to the real running time, it took about 22 hours with VART on the Patent dataset comparing to 36 hours with exBERT$_{ext}$ and 40 hours with exBERT$_{whole}$. In the fine-tuning process, it took about 19 hours with VART model on EU-Leg dataset comparing to 40 hours with exBERT. Besides, VART has about 17% fewer parameters than exBERT without sacrificing performances. As to the computational resource cost, VART only requires 1 GPU for both pre-training and fine-tuning, while exBERT needs 2 GPUs under the same setting. All the evidences demonstrate that VART is able to further improve the performance on basis of conventional PTLM adaptation methods with a high efficiency and a low computational resource cost.

4.5 Impact of Vocabulary Size

We further conduct experiments to test VART model with different vocabulary sizes. API `BertWordPieceTokenizer` is used for extracting vocabularies from the target datasets. By setting the parameter `vocab_size` with different values, we can control the extracted vocabulary size. For each training dataset, we extract four vocabularies by setting `vocab_size` with 10k, 20k, 30k and 40k separately. The settings for pre-training and fine-tuning with different vocabularies remain the same as Section 4.2.

The detailed results of different vocabulary sizes are listed in Table 4, which is also visualized in Figure 3. From the results we can observe that VART model produces better results than BERT$_{tapt}$ model in most cases. This indicates that extending the vocabulary is beneficial to the MLDC task.

On the other hand, we could arrive at a similar conclusion with Tai et al. (2020) that increasing the vocabulary size may not always produce better results. The best score coming from the largest vocabulary can be only seen on the EU-Leg dataset. On the CEI and Patent datasets, the best scores are achieved with 10k and 20k vocabularies instead of using their largest vocabularies.

We hypothesis that the vocabulary size and the dataset size should to be proportional. Since larger vocabularies from smaller datasets may contain more low frequency words which provide less information for the MLDC task. However, the relationship between the vocabulary size and the training sample number is worth studying in the further.

5 Conclusion

We introduced VART, a concise pre-training method to extend the BERT model with domain OOV words. With a minor modification of adding an extra embedding layer to the original BERT model, we can adapt the BERT model to the target task datasets. Comparing with the conventional method such as exBERT, our approach maximizes the use of general domain BERT model with much higher efficiency, such as less pre-training time and lower computational resource requirements. Experiment results indicate that our approach leverages the domain gap of PTLMs for MLDC tasks. Since our approach is a general solution for adapting the BERT model, in the future we would like to examine VART in other NLP tasks.

References

Alec Radford and Karthik Narasimhan. 2018. *Improving language understanding by generative pretraining*. https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. *Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding*. https://arxiv.org/abs/1810.04805

Zhihao Dai, Zhong Li, and Lianghao Han. 2021. *BoneBert: A BERT-based Automated Information Extraction System of Radiology Reports for Bone Fracture Detection and Diagnosis. Advances in Intelligent Data Analysis XIX*, pp. 263–274. https://doi.org/10.1007/978-3-030-74251-5_21

Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. *BoneBert: BOND: BERT-Assisted Open-Domain Named Entity Recognition with Distant Supervision. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp 1054—1064.

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. *DocBERT: BERT for Document Classification*. https://arxiv.org/abs/1904.08398

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. *Multi-passage BERT: A Globally Normalized BERT Model for Open-domain Question Answering*. https://arxiv.org/abs/1908.08167

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. *Incorporating BERT into Neural Machine Translation*. https://arxiv.org/abs/2002.06823

Gao Zhengjie, Feng Ao, Song Xinyu, and Wu Xi. 2019. *Target-Dependent Sentiment Classification With BERT*. *IEEE Access*, vol. 7, pp. 154290–154299. 10.1109/ACCESS.2019.2946594

Elyne Scheurwegs, Boris Culea, Kim Luyckx, Léon Luyten, and Walter Daelemans. 2017. *Selecting relevant features from the electronic health record for clinical code prediction*. Journal of Biomedical Informatics, vol. 74, pp. 92–103. https://doi.org/10.1016/j.jbi.2017.09.004

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. *Explainable Prediction of Medical Codes from Clinical Text*. https://arxiv.org/abs/1802.05695

Jingcheng Du, Qingyu Chen, Yifan Peng, Yang Xiang, Cui Tao, and Zhiyong Lu. 2019. *ML-Net: multi-label classification of biomedical texts with deep neural networks. Journal of the American Medical Informatics Association*, vol. 26, pp. 1279—1285. https://doi.org/10.1093/jamia/ocz085

Simon Baker and Anna Korhonen. 2017. *Initializing neural networks for hierarchical multi-label text classification*. BioNLP, https://doi.org/10.17863/CAM.12418

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. *How to Fine-Tune BERT for Text Classification?*. China National Conference on Chinese Computational Linguistics, pp 194–206. https://arxiv.org/abs/1905.05583

Iz Beltagy, Arman Cohan, and Kyle Lo. 2019. *Scibert: Pretrained contextualized embeddings for scientific text*. https://arxiv.org/abs/1903.10676

Wen Tai, H. T. Kung, Xin Dong, Marcus Comiter, and Chang-Fu Kuo. 2020. *exBERT: Extending Pre-trained Models with Domain-specific Vocabulary Under Constrained Training Resources*. *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp 1433–1439. https://aclanthology.org/2020.findings-emnlp.129

Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. 2019. *Adapt or Get Left Behind: Domain Adaptation through BERT Language Model Finetuning for Aspect-Target Sentiment Classification*. https://arxiv.org/abs/1908.11860

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. *Don't Stop Pretraining: Adapt Language Models to Domains and Tasks*. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp 8342–8360. https://arxiv.org/abs/2004.10964

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. *RoBERTa: A robustly optimized BERT pretraining approach*. https://arxiv.org/abs/1907.11692

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. *Domain specific language model pretraining for biomedical natural language processing*. https://arxiv.org/abs/2007.15779

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. *Biobert: pre-trained biomedical language representation model for biomedical text mining*. https://arxiv.org/abs/1901.08746

Jieh-Sheng Lee and Jieh Hsiang. 2020. *Patent classification by fine-tuning BERT language model*. https://doi.org/10.1016/j.wpi.2020.101965

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. *Google's neural machine translation system: Bridging the gap between human and machine translation*. https://arxiv.org/abs/1609.08144

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp 38–45. https://www.aclweb.org/anthology/2020.emnlp-demos.6

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. *Attention is all you need. Advances in Neural Information Processing Systems*, pp 5998-–6008.

Kristin Larsson, Simon Baker, Ilona Silins, Yufan Guo, Ulla Stenius, Anna Korhonen, and Marika Berglund. 2017. *Text mining for improved exposure assessment. PLoS One*.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Ion Androutsopoulos . 2019. *Large-Scale Multi-Label Text Classification on EU Legislation. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp 6314–6322.

Wei Huang, Enhong Chen, Qi Liu, Yuying Chen, Zai Huang, Yang Liu, Zhou Zhao, Dan Zhang, and Shijin Wang. 2019. *Hierarchical Multi-label Text Classification: An Attention-based Recurrent Network Approach*. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp 1051—1060.

Diederik P. Kingma and Jimmy Ba. 2015. *Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations, Conference Track Proceedings*.

Wei Huang, Enhong Chen, Qi Liu, Yuying Chen, Zai Huang, Yang Liu, Zhou Zhao, Dan Zhang, and Shijin Wang. 2019. *Hierarchical Multi-label Text Classification: An Attention-based Recurrent Network Approach*. *Proceedings of the 28th ACM International Conference on Information, Knowledge Management*, pp 1051–1060.