VE-KD: a method for training smaller language models adapted to specific domains

Anonymous ACL submission

Abstract

We propose VE-KD, a novel method juggling knowledge distillation and vocabulary expansion to train efficient domain-specific language models. In comparison with traditional pre-training approaches, VE-KD provides competitive performance in downstream tasks while reducing model size and required computational resources. Our experiments with different biomedical domain tasks demonstrate that VE-KD performs well compared with models such as BioBERT (+1% at HoC) and PubMedBERT (+1% at PubMedQA), with about 96% reduced training time. Furthermore, it outperforms DistilBERT, and offers a significant improvement in document-level tasks. Investigation of vocabulary size and tolerance, which are hyperparameters of our method, provides insights for further model optimization. The fact that VE-KD consistently maintains its advantages even when the corpus size is small suggests that it is a practical approach for domain-specific language tasks, and is transferrable to different domains for broader applications.

1 Introduction

Language models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) have provided significant performance improvements in solving natural language processing (NLP) tasks, enabling many practical applications that increase productivity, understanding, and accessibility in diverse industries.

These traditional models still hold value in terms of cost-effectiveness and ease of deployment, even though large language models (LLMs) demonstrate remarkable few-shot capabilities in NLP tasks. One reason is that training or fine-tuning LLMs such as GPT-3 requires an immense amount of data and computational resources. Another reason is a growing demand for AI applications that run on local machines because some applications require independence from network connectivity or have concerns over information security and confidentiality when using LLM API services such as GPT-4.

Various industrial and academic fields include specialized terms and concepts which general language models might not fully understand. These potential gaps in understanding of general language models may result in less effective or even erroneous solutions, it is therefore vital to adapt language models to specific domains.

However, LLMs such as GPT-3 and GPT-4 are difficult to use because it is expensive and challenging to obtain high-quality labeled data for additional pre-training, or because domain knowledge must be added through the API. In contrast, general BERT models have the advantage of easy of fine-tuning and specialization in different domains.

In industrial applications, operational efficiency is often the primary concern. For example, high latency can be detrimental for applications that require real-time response or that process large amounts of input data, such as monitoring systems or predictive analytics. Larger models need more powerful and thus more expensive hardware setups, but typically have capacity constraints imposed to manage costs. This also limits the model size that can feasibly be executed. Therefore, reducing resource consumption by compressing a model improves its deployment adaptability.

Although the need for domain adaptation and model compression is particularly prominent in industrial applications within a specific domain, when considering the complexities inherent in these processes are considered, a simplistic sequential approach may not yield the best results. First, both tasks face the challenge of obtaining high-quality data. Second, using general methods such as domain-adaptation followed by distillation or distilling the domain-adapted model requires two-step training and hyperparameter tuning (Yao et al.,

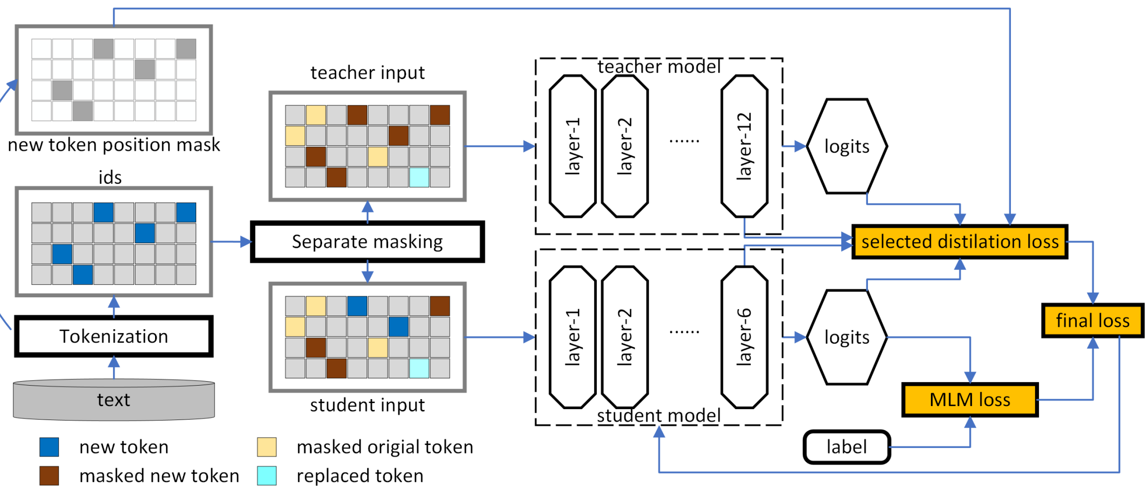


Figure 1: The architecture of VE-KD. New tokens and original tokens are processed separately during tokenization, masking and loss calculation. The student model soaks up two types of knowledge; one is common knowledge via original tokens and the other is domain-specific knowledge via new tokens.

2021), which makes the learning process difficult to optimize.

During the domain-adaptation phase in particular, such as secondary-stage unsupervised pre-training, there is a significant risk of losing general knowledge due to overlearning when a small corpus is used. Moreover, two-step training requires more computational resources and time, possibly requiring further iterations to achieve the most effective outcomes. Hence, a method that can proficiently perform domain adaptation and model compression simultaneously is distinctly necessary to overcome these issues.

In this paper, we propose VE-KD, a novel simple mechanism that can simultaneously perform domain adaptation and model compression from a teacher model such as BERT. We also show that our method significantly outperforms the teacher model on related tasks with corpus, with easy optimization and robustness, and lower computational resources and time.

2 Related Work

Large pre-trained models,like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), have become ubiquitous in NLP (Ramponi and Plank, 2020). In terms of a domain shifts, secondary-stage unsupervised pre-training on new domain has proven to be advantageous. Contextualized tokenizations are adapted to text from the target domain through masked language modeling, as introduced by Han and Eisenstein (2019), Gururangan et al. (2020). Lee et al. (2020) which executed continual pre-training to adapt the BERT model to the biomedical domain, by utilizing both the PubMed abstracts and PMC full-text resources. The use of contrastive learning also increases the representation ability for specific domains. Xu et al. (2023) investigated the use of contrastive learning to develop discriminative entity representations in the field of cross-domain named entity recognition.

However, many specialized domains contain unique words that are not included in the vocabulary of pre-trained language models. Gu et al. (2021) proposed a biomedical pre-trained model called PubMedBERT in which the vocabulary was constructed from scratch and the model was pre-trained from scratch. Furthermore, in many specialized domains, sufficiently large corpora may not be available to support pre-training from scratch. General domain vocabulary can be extended with in-domain vocabulary (Yao et al., 2021), to solve this out-of-vocabulary issue.

Knowledge distillation (KD) (Hinton et al., 2015) aims to transfer the knowledge from a large teacher model to a small student model. Existing knowledge distillation methods can be divided into three categories: response-based, feature-based, and relation-based (Gou et al., 2021). In this paper, we focus on task-agnostic knowledge distillation approaches, where a distilled smaller pre-trained model can be directly fine-tuned on downstream tasks.

DistilBERT (Sanh et al., 2019) uses soft labels and embedding outputs to supervise the student model. TinyBERT (Jiao et al., 2020) and MobileBERT (Sun et al., 2020) introduce self-attention distributions and hidden states for training the student model. MiniLM (Wang et al., 2020) avoids restrictions on the number of student layers and supervises the student model by using the self-attention distributions and value relation of the teacher's last transformer layer. AD-KD approach (Wu et al., 2023) explores the token-level rationale behind the teacher model based on Integrated Gradients (IG) and transfers attribution knowledge to the student model.

3 Methods

In this study, we propose VE-KD, a method for model distillation with extendable vocabulary,as shown in Figure 1. Unlike Adapt-and-Distill (Yao et al., 2021) which requires two-step training, our approach simultaneously lightens the model and resolves the adaptability issues of special domains, which have been a problem in general-purpose models pre-trained on large corpora, particularly when using smaller corpora.

In the knowledge distillation aspect of VE-KD, a larger BERT model serves as the teacher model, instructing a smaller student model layer-by-layer. Through the distillation process, the student model becomes able to mimic the behavior of the larger teacher model in general terms. Simultaneously, the vocabulary expansion aspect broadens the model's vocabulary to capture domain-specific terms, thereby enhancing the method's ability to adapt to domain-specific tasks.

3.1 Vocabulary Expansion

We add domain-specific terms (we call new tokens) through vocabulary expansion, which distinguishes between general and domain knowledge by separating the new tokens from the original tokens. By processing them separately such as through different masking and loss functions, we allow for simultaneous learning of domain knowledge from the corpus and general knowledge from the teacher model through two separate pathways.

The vocabulary of the student model V_s is expanded based on the teacher model's vocabulary V_t. We use tensor2tensor's WordPiece generation script[1] to perform the vocabulary expansion. Followed on from the research of Yao et al. (2021), we chose a vocabulary size of 60k.

[1]https://github.com/tensorflow/tensor2tensor

3.2 Tokenization and Separate Token Masking

The process of separating the two terms is accomplished through tokenization and token masking. Typically, model distillation necessitates that both the teacher and student models possess identical dictionaries. However, due to vocabulary expansion, new tokens emerge that cannot be incorporated into the teacher model.

As shown in Figure 1, we employ text tokenization with an expanded vocabulary V_s. There are new tokens that cannot be accommodated in the teacher model. To circumvent this, we designed a unique mask method as below.

We denote the input sequence as $x = [x_1, x_2, x_3, ..., x_n]$, where n is the sequence length and each x_i represents a token tokenized by expanded vocabulary V_s, Let us suppose that x_1 and x_3 are new tokens and thus not included in V_t, then we replace them with a [MASK] token as new input

$$x_{\text{input}} = [[\text{MASK}], x_2, [\text{MASK}] ..., x_n].$$

We simultaneously acquire the position information of new tokens $P_{\text{newtoken}}(i) = 1$ if $x_i \notin V_t$ else 0, and use to calculate the loss function.

In areas other than new tokens, similar to BERT's MLM (Masked Language Model) task, tokens are masked and swapped at random by the same rule. The tokens used for replacement are picked from the vocabulary of the teacher model.

3.3 Loss Functions

This section explains the mechanism of calculating the loss function by separating new tokens from general terms. In the right half of Figure 1, we input the two entries into the teacher model (t) and the student model (s), and obtain the hidden state vectors $H_{t,s}$ from the final layer and the token prediction logits $L_{t,s}$.

At the new token position, the output logits and the hidden vectors state of the teacher model confict with the student model because the student model has a bigger vocabulary and new knowledge. In order to learn the knowledge of the teacher model successfully, similarity calculations are only made within the scope of general terms (without the new token position). The new $H'_{t,s}$ and $L'_{t,s}$ are formulated as follows:

$$H'_{t,s} = \{H_{t,s}(i) | P_{\text{newtoken}}(i) = 0\},$$

$$L'_{t,s} = \{L_{t,s}(i) | P_{\text{newtoken}}(i) = 0\}.$$

Following DistilBERT (Sanh et al., 2019), the loss function is calculated using the following three measures such as cosine similarity, Kullback-Leibler divergence (KL), and mean squared error (MSE), which are defined as follows:

$$\mathcal{L}_{\text{Cosine}}(H'_t, H'_s) = \frac{H'_t \cdot H'_s}{\|H'_s\| \|H'_t\|},$$

$$\mathcal{L}_{\text{KL}}(L'_t, L'_s) = \sum_i L'_t(i) \log \frac{L'_t(i)}{L'_s(i)},$$

$$\mathcal{L}_{\text{MSE}}(L'_t, L'_s) = \frac{1}{n} \sum_{i=1}^{n} \left(L'_t(i) - L'_s(i)\right)^2.$$

By doing so, we facilitate the learning of the teacher model's knowledge.

Next, similar to BERT, we calculate the masked language model loss function $\mathcal{L}_{\text{MLM}}$ to estimate the masked words using the student model's Logits L_s and labels L_{label}.

The KD loss and MLM loss may be in conflict because of the new token even if the calculation range is split. Knowledge about general terms between the teacher model and student model maybe differ because the meaning or grammar of general terms around the new token maybe different. Since taking 100% of the knowledge from the teacher model may have adverse effects on creating new domain knowledge for the student model. We therefore use the tolerance to control the KD loss as

$$\mathcal{L'}_{\text{KD}}(i) = \max(\mathcal{W}_{\text{KD}} \times \mathcal{L}_{\text{KD}}(i) - \varepsilon, 0).$$

In this context, $\mathcal{L}_{\text{KD}}$ refers to each KD loss, $\mathcal{W}_{\text{KD}}$ represents the weight for each KD loss, and ε denotes the tolerance for the KD loss. This implies that after being multiplied by the weight, if the value is smaller than ε, the model will consider the KD loss to be zero and refrain from further optimization for lower loss. If a conflict arises, the student model will first optimize the MLM loss. This ensures that the student model learns the new domain knowledge in the vicinity of the teacher model, without straying too far from it.

The final loss $\mathcal{L}_{\text{final}}$ is obtained by calculating the sum of the above individual losses, namely

$$\mathcal{L}_{\text{final}} = \mathcal{L'}_{\text{Cosine}} + \mathcal{L'}_{\text{KL}} + \mathcal{L'}_{\text{MSE}} + \alpha \mathcal{L}_{\text{MLM}},$$

where α is the positive weight parameter for the loss in the MLM task and is used to control the intensity of learning new tokens.

4 Experiment Details and Results

In this section, we conduct our experiments in the biomedical domain.

4.1 Datasets

We collected a PubMed abstract corpus for distillation, and using BLURB[2] for performance evaluation.

For the biomedical domain, we gathered a small-scale corpus of 1.3GB from PubMed abstracts and compare it with PubMedBERT, which used a 21GB corpus for pre-training. We omitted any abstracts containing fewer than 128 words to reduce noise.

We evaluate downstream tasks by using 12 tasks of the BLURB benchmark (excluding BIOSSES, a sentence similarity task that employs the [CLS] token, which is not well trained with this method). This benchmark consists of five named entity recognition tasks (BC5-Chemical, BC5-Disease, NCBI-disease, BC2GM and JNLPBA), a PICO (population, intervention, comparison, and outcome) extraction task (EBM PICO), three relation extraction tasks (ChemProt, DDI and GAD), a document classification task (HoC), and two question answering tasks (PubMedQA and BioASQ). We adhere to the same fine-tuning method and evaluation metrics as those used by PubMedBERT following Yasunaga et al. (2022). We list the statistics of those tasks in Table 1.

Dataset	**Train**	**Dev**	**Test**
BC5-chem (2016)	5,203	5,347	5,385
BC5-disease (2016)	4,182	4,244	4,424
NCBI-disease (2014)	5,134	787	960
BC2GM (2008)	15,197	3,061	6,325
JNLPBA (2004)	46,750	4,551	8,662
EBM PICO (2018)	339,167	85,321	16,364
ChemProt (2010)	18,035	11,268	5,745
DDI (2013)	25,296	2,496	5,716
GAD (2004)	4,261	535	534
HoC (2016)	1,295	186	371
PubMedQA (2019)	450	50	500
BioASQ (2015)	670	75	140

Table 1: The numbers of instances included in BLURB biomedical NLP benchmark datasets we used.

[2]https://microsoft.github.io/BLURB/leaderboard.html

	BERT-base	DistilBERT$_{\text{PubMed}}$	VE-KD$_o$	VE-KD$_w$
NER				
BC5CDR-chem	89.25	88.81	89.64	**89.83**
BC5CDR-disease	81.44	78.94	**81.77**	81.65
NCBI-disease	85.67	84.07	86.46	**86.50**
BC2GM	**80.90**	79.94	79.68	80.03
JNLPBA	**77.69**	76.64	76.50	76.34
PICO extraction				
EBM PICO	**72.34**	71.22	72.15	72.08
Relation extraction				
ChemProt	**71.86**	70.77	69.13	69.28
DDI	**80.04**	74.20	74.80	76.69
GAD	**80.41**	78.29	76.57	77.82
Document classification				
HoC	80.20	80.76	82.38	**83.21**
Question answering				
PubMedqa	51.62	53.40	54.80	**55.80**
BioASQ	70.36	67.86	72.86	**75.71**
Average of all tasks	76.82	75.41	76.39	**77.08**

Table 2: Comparison with distillation models trained by the PubMed corpus, DistilBERT$_{\text{PubMed}}$: using the same method of DistilBERT, VE-KD$_o$ and VE-KD$_w$ are models trained by our method, where o indicates without tolerance and w with. Bold indicates the top-ranked performance, Bold and underline indicate the first best and the second best, respectively.

4.2 Implementation

We use the uncased version of BERT$_{\text{BASE}}$[3] (12 layers,768 hidden size) as the teacher model. We perform distillation of BERT to a small (6 layer 768 hidden state) student model[4] with vocabulary expansion. More specifically, we use a peak learn rate of 5e-4, batch size of 240, and train for steps. We warm up the learning rate in the first 10% of steps and then linearly decay it. Additionally, We perform distillation of BERT by the normal method using the same corpus and hyperparameters to a 6-layer distilBERT$_{\text{PubMed}}$.

For comparison, we choose the teacher model BERT as baseline. Additionally, we choose some 6-layer small BERT or distilled BERT for general purpose, such as BERT$_{\text{L6H768}}$[3](6 layers,768 hidden size), TinyBERT, MiniLM or DistilBERT$_{\text{wiki}}$. For comparison with domain adaptation ability, we fine tune these models using the PubMed corpus. Specifically, we use a peak learn rate of 5e-4, batch size of 80, and train for 100,000 steps. We warm up the learning rate in the first 10% of steps and then linearly decay it.

[3]https://github.com/google-research/bert.

[4]Our model and evaluate dataset is available at: https://github.com/pZvfkv3t8PA9vAc/VE-KD_a-method-for-training-smaller-language-models-adapted-to-specific-domains.

4.3 Comparison With BERT and DistilBERT Trained by the PubMed Corpus

The results for the performance comparison of the distillation model using the same PubMed corpus are shown in Table 2, which shows that VE-KD$_w$ outperforms teacher model BERT on 6 tasks, and has an improved performer of 0.3% on average. VE-KD$_w$ outperforms DistilBERT$_{\text{PubMed}}$ on 10 tasks, achieving an increased performance of 2% absolute on average. Moreover, we see a trend of significantly larger improvements on document-level tasks compared with BERT-base document classification (+3% on HoC) and question answering (+4% on PubMedQA, +5% on BioASQ). Compared with DistilBERT, document classification (+2% on HoC) and question answering (+2% on PubMedQA, +8% on BioASQ). A reasonable explanation for why the HoC, PubMedQA, and BioASQ tasks show a substantial increase in performance is that they were developed from PubMed abstracts, which may have a high degree of similarity to the corpus we employed for training VE-KD.

domain adaptation	BERT-small o	BERT-small w	TinyBERT o	TinyBERT w	MiniLM o	MiniLM w	DistilBERT wiki o	DistilBERT wiki w	DistilBERT PubMed o	DistilBERT PubMed w	VE-KD$_w$
NER											
BC5CDR-chem	88.64	**90.51**	87.98	90.34	88.93	90.13	88.81	90.34	88.97	89.83	89.83
BC5CDR-disease	80.27	**81.90**	79.20	80.60	80.04	80.24	78.94	80.60	80.84	80.74	81.65
NCBI-disease	85.53	85.54	84.16	84.77	83.81	84.37	84.07	84.77	86.05	84.52	**86.50**
BC2GM	79.64	**80.22**	79.56	80.17	80.09	80.18	79.94	80.17	79.96	79.83	80.03
JNLPBA	76.53	**77.27**	76.83	76.75	75.92	76.65	76.64	76.75	76.86	76.60	76.34
PICO extraction											
EBM PICO	71.09	72.21	70.41	72.31	71.29	**72.53**	71.22	72.31	71.56	72.16	72.08
Relation extraction											
ChemProt	69.74	69.97	69.87	70.09	69.50	70.64	70.77	70.09	69.68	**71.11**	69.28
DDI	75.91	**77.57**	75.01	75.95	74.91	76.92	74.20	75.95	75.96	75.48	76.69
GAD	78.79	79.60	76.87	78.98	79.05	**79.74**	78.29	78.98	76.66	79.53	77.82
Doc classification											
HoC	81.73	82.66	73.98	81.21	77.72	81.41	80.76	81.21	81.41	82.20	**83.21**
Question answering											
PubMedQA	50.40	51.80	54.00	51.80	52.60	54.60	53.40	51.80	50.00	53.80	**55.80**
BioASQ	75.71	**80.00**	**80.00**	67.86	67.14	76.43	67.86	67.86	62.86	72.14	75.71
Average of all tasks	76.16	**77.44**	75.66	75.90	75.08	76.99	75.41	75.90	75.07	75.98	77.08

Table 3: Comparison among small models, where o indicates without domain adaptation and w with. Bold and underline indicate the first best and the second best, respectively.

VE-KD did not perform as well in the relation extraction task as DistilBERT experiencing an average performance decrease of 3% compared with BERT-base. This might be attributable to the considerable divergence between the datasets used in tasks such as DDI and GAD (which were not built from the PubMed corpus), and the PubMed corpus we used to train VE-KD. Therefore, we postulate that the performance of VE-KD is significantly influenced by the gap between the training corpus and the downstream task.

4.4 Effect of Tolerance Setting

As Table 2 shows, VE-KD$_w$ with tolerance setting achieves a performance increase of 0.7% on average compared with the model without tolerance setting. We see a trend where the tolerance setting gives a huge improvement on document-level tasks such as document classification (+1% on HoC) and question answering (+1% on PubMedQA, +3% on BioASQ).

In the DDI task, VE-KD without tolerance shows a huge performance decline similar to that of DistilBERT when using the same corpus. However, when a tolerance setting is added to VE-KD, it achieves a performance increase of 2%. This result suggests that our method can partially offset performance loss caused by differences in data distribution between the training corpus and downstream task.

4.5 Compare with Same Layer Size Model

Table 3 shows the results of performance comparison versus the small model with the same layers and hidden state size as VE-KD. Compared with small models without domain adaptation, VE-KD$_w$ achieves the highest performance on average. Even after domain adaption, VE-KD$_w$ is still the second highest model just behind the BERT-small model. Compared with the DistilBERT$_{\text{PubMed}}$ which uses the same corpus, VE-KD also attains a 0.5% performance increase on average, and in particular obtains a 2% increase for PubMedQA tasks. Our results suggest that a vocabulary expansion distillation method using one-time training can achieve or exceed the performance of adaptation followed by distillation.

5 Analysis

In this section, we analyzed the impact of training time and various settings on performance.

5.1 Impact of Training Time

Pre-training and fine-tuning typically require substantial computational resources. We benchmark our model against BioBERT and PubMedBERT using the HoC, PubMedQA task. To facilitate a fair comparison, we equate the training time of BioBERT and PubMedBERT to the duration it would potentially take with the same computational resources as used in this study (8 A100 GPUs).

As shown in Table 4 for the HoC and PubMedQA task, VE-KD outperforms BERT in the HoC task after 3 hrs of training. Moreover, it surpasses BioBERT and PubMedBERT following 6 and 9 hrs of training, respectively. For the PubMedQA task, VE-KD outperforms BERT after 6 hrs of training, and PubMedBERT after 9 hrs of training. These observations highlight the efficiency of our method as it can match or surpass the performance of models pre-trained from scratch, all while leveraging less than 10% of the computational resources and corpus.

The training time for VE-KD is mostly analogous to the distillation phase time of the 'distil-then-adapt' method. Compared with VE-KD with fine-tuned DistilBERT, VE-KD achieves a higher score while requiring only about half of the training time.

Model	**Training Time**	**Corpus Words**	**HoC**	**PubMed QA**
VE-KD	3 hrs	0.2B	81.64	54.00
	6 hrs	0.2B	81.74	55.30
	9 hrs	0.2B	82.64	56.60
DistilBERT	9 hrs	0.2B	80.76	53.40
DistilBERT ft.	19 hrs	0.2B	82.38	53.80
BERT	0 hrs	3.3B	80.20	54.00
BioBERT	240 hrs	4.5B	81.54	60.24
PubMedBERT	240 hrs	3.1B	82.32	55.84

Table 4: Results with different model training, where ft indicates that the model is fine-tuned.

5.2 Impact of Vocabulary Size

To understand the impact of vocabulary size, we carry out several experiments using varying vocabulary sizes in the biomedical domain. We use the same experimental conditions with two types of models: with or without tolerance setting. Figure 2 shows the performance of the model for different vocabulary sizes.

We observe that both types models deliver the best results with a vocabulary size of 60k in our study. Interestingly, models with larger vocabularies of 70k and 80k do not exhibit better performance but instead exhibit a significant performance loss. A reasonable explanation for these results may be that a larger vocabulary set can include more complex but less frequent tokens, which cannot be sufficiently learned through continuous pre-training, especially in a small-scale corpus.

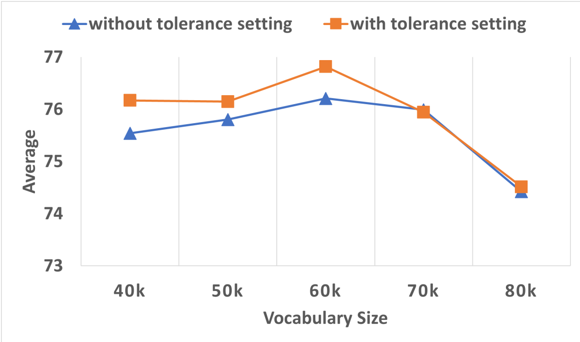


Figure 2: The average performance of VE-KD with different vocabulary size.

5.3 Impact of Tolerance

To understand the impact of tolerance, we conducted several experiments in which adjusting the tolerance is adjusted within a 60k vocabulary by utilizing HoC, PubMedQA, BioASQ, and averaging across all 12 tasks.

As shown in Figure 3, there is a noticeable change in performance between the model without tolerance setting, and each task as well as the average over the 12 tasks exhibits a peak performance when the tolerance is set to 0.5. We observe that as the tolerance increases up to 1.0 and 2.0, the performance continually decreases, compared with the model without tolerance setting. This implies that when the tolerance is excessively high, the instructional knowledge from the teacher model may not be effectively assimilated by the student model. Given that the current tolerance setting might be too restrictive for this method, we are considering modifying it to a softer approach in the future.

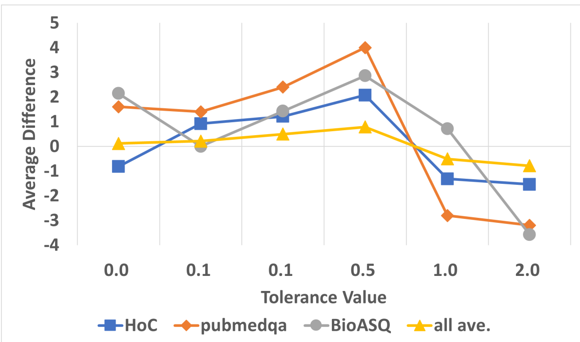
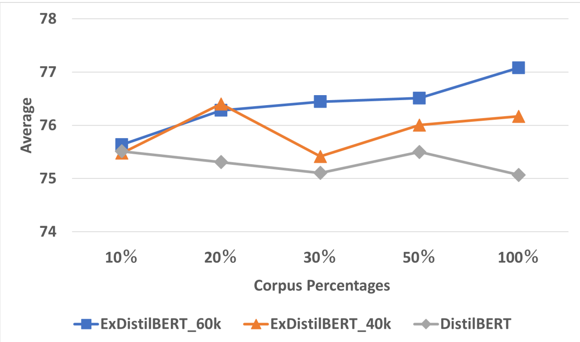


Figure 3: HoC, PubMedQA, BioASQ and the average performance of VE-KD with different tolerance.

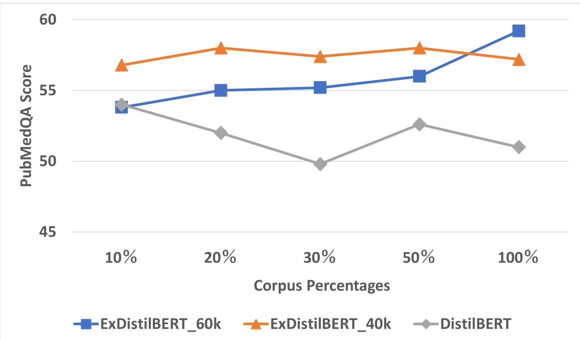
5.4 Smaller Corpus

To understand the potential of our method on smaller corpora, we carried out several experiments on VE-KD (with 40k and 60k vocabularies) and DistilBERT trained on varying percentage of the PubMed corpus.

Figure 4 shows the performance evaluation results for average score and the PubMedQA task. We observe that VE-KD_40k and VE-KD_60k trained on more than 20% of the corpus, and the 40k vocabulary model had larger fluctuations on average score than 60k at the same time. Interestingly, for the PubMedQA task, the model with 60k performs worse than the model with 40k Up until 100% of the dataset. One potential explanation for this is that the model with a 60k vocabulary has more parameters, implying that it requires additional training to achieve comparable performance. However, a model that implements a smaller vocabulary expansion may offer greater potential when applied to a small corpus.



(a) Average score of 12 tasks



(b) PubMedQA score

Figure 4: Performance on varying percentages of the PubMed corpus. VE-KD_40k and VE-KD_60k denote VE-KD with 40k and 60k vocabulary size.

5.5 Inference Speed and Model Size

We compare the parameter size and inference speed of VE-KD with BERT model and DistilBERT, and the results are shown in Table 5, Compared to BERT-base, the half layers DistilBERT and VE-KD are about 0.5 times faster. We find that vocabulary expansion delivers only marginal improvements on the model's inference speed, the same as the results of Yao et al. (2021).

For the model size of VE-KD, 40k and 60k vocabulary expansion gives about 8M and 22M parameters in the tokenization weights, respectively. The model lightening effect is thus smaller. For further model lightening, it might be necessary to have smaller size hidden dimension or less layers or number of attention heads.

Models	**#Params**	**Speedup**
BERT	110M	x1.00
DistilBERT	67M	x1.48
VE-KD_40k	75M	x1.50
VE-KD_60k	90M	x1.56

Table 5: Comparison of parameter's size and inference speed. The inference speed is test by EBM PICO task, and evaluated on single RTX 6000 GPU. VE-KD_40k and VE-KD_60k denote VE-KD with 40k and 60k vocabulary size.

6 Conclusion

In this paper, we proposed VE-KD, a novel method that merges vocabulary expansion and knowledge distillation. We also showed that our method achieves competitive performance on various downstream tasks, despite small model sizes and reduced computational resource requirements compared with standard domain-specific pre-training approaches. Our experimental results demonstrate that VE-KD is effective; that is to say, its performance is competitive with well-known models such as BioBERT and PubMedBERT, and its efficiency of pre-training is noteworthy. For document-level tasks in particular, it outperforms DistilBERT.

We then investigated the effects of vocabulary size and tolerance in detail and obtained insights that can help us configure more efficient models. Furthermore, VE-KD provides the benefits of consistency even when smaller corpus sizes were utilized. Due to its efficiency across various domain-specific language processing tasks, VE-KD sets the stage for further research in task-specific model optimization and application across diverse domains.

One limitation of our study is that we did not evaluate the model's generalization abilities on out-of-domain tasks, which could be crucial for certain applications. Further evaluation of them is part of our future work.

References

Simon Baker, Ilona Silins, Yufan Guo, Imran Ali, Johan Högberg, Ulla Stenius, and Anna Korhonen. 2016. Automatic semantic classification of scientific literature according to the hallmarks of cancer. *Bioinformatics*, 32(3):432–440.

Kevin G Becker, Kathleen C Barnes, Tiffani J Bright, and S Alex Wang. 2004. The genetic association database. *Nature genetics*, 36(5):431–432.

Nigel Collier, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Jin-Dong Kim. 2004. Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pages 73–78, Geneva, Switzerland. COLING.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4238–4248, Hong Kong, China. Association for Computational Linguistics.

María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. 2013. The ddi corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of biomedical informatics*, 46(5):914–920.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

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. *arXiv preprint arXiv:1907.11692*.

Benjamin Nye, Junyi Jessy Li, Roma Patel, Yinfei Yang, Iain J Marshall, Ani Nenkova, and Byron C Wallace. 2018. A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2018, page 197. NIH Public Access.

Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in NLP—A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Larry Smith, Lorraine K Tanabe, Rie Johnson nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M Friedrich, Kuzman Ganchev, et al. 2008. Overview of biocreative ii gene mention recognition. *Genome biology*, 9:1–19.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. MobileBERT: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170, Online. Association for Computational Linguistics.

Olivier Taboureau, Sonny Kim Nielsen, Karine Audouze, Nils Weinhold, Daniel Edsgärd, Francisco S Roque, Irene Kouskoumvekaki, Alina Bora, Ramona Curpan, Thomas Skøt Jensen, et al. 2010. Chemprot: a disease chemical biology database. *Nucleic acids research*, 39(suppl_1):D367–D372.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):1–28.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Siyue Wu, Hongzhan Chen, Xiaojun Quan, Qifan Wang, and Rui Wang. 2023. AD-KD: Attribution-driven knowledge distillation for language model compression. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8449–8465, Toronto, Canada. Association for Computational Linguistics.

Jingyun Xu, Changmeng Zheng, Yi Cai, and Tat-Seng Chua. 2023. Improving named entity recognition via bridge-based domain adaptation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3869–3882, Toronto, Canada. Association for Computational Linguistics.

Yunzhi Yao, Shaohan Huang, Wenhui Wang, Li Dong, and Furu Wei. 2021. Adapt-and-distill: Developing small, fast and effective pretrained language models for domains. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 460–470, Online. Association for Computational Linguistics.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, Dublin, Ireland. Association for Computational Linguistics.

A BLURB fine-tuning details

We apply the following fine-tuning hyperparameters to all models, including the baseline with same defaults training seed.

We set max_seq_length to 512 and choose learning rates from {1e-5, 2e-5, 3e-5, 5e-5, 6e-5}, batch sizes from {16, 32, 64} and fine-tuning epochs from 1–120.