VE-KD: Vocabulary-Expansion Knowledge-Distillation for Training Smaller Domain-Specific Language Models

Anonymous ACL submission

Abstract

We propose VE-KD, a novel method that balances knowledge distillation and vocabulary expansion with the aim of training efficient domain-specific language models. Compared with traditional pre-training approaches, VE-KD exhibits competitive performance in downstream tasks while reducing model size and using fewer computational resources. Additionally, VE-KD refrains from overfitting in domain adaptation. 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% less training time. Furthermore, it outperforms DistilBERT and Adapt-and-Distill, showing 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 have increased 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 the growing demand for artificial intelligence (AI) applications that run on local machines because some applications require independence from network connectivity or have concerns about information security and confidentiality when using LLM-based application programming interface (API) services such as GPT-4.

Many industrial and academic fields use specialized terminology and concepts that general language models might not fully understand. These potential gaps in understanding may result in less effective or even erroneous solutions, making it essential 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 and 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. For example, BERT performs better in Named-entity recognition tasks compared with GPT-family models such as BioGPT (Luo et al., 2022).

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 realized. 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, given the complexities inherent in these processes,

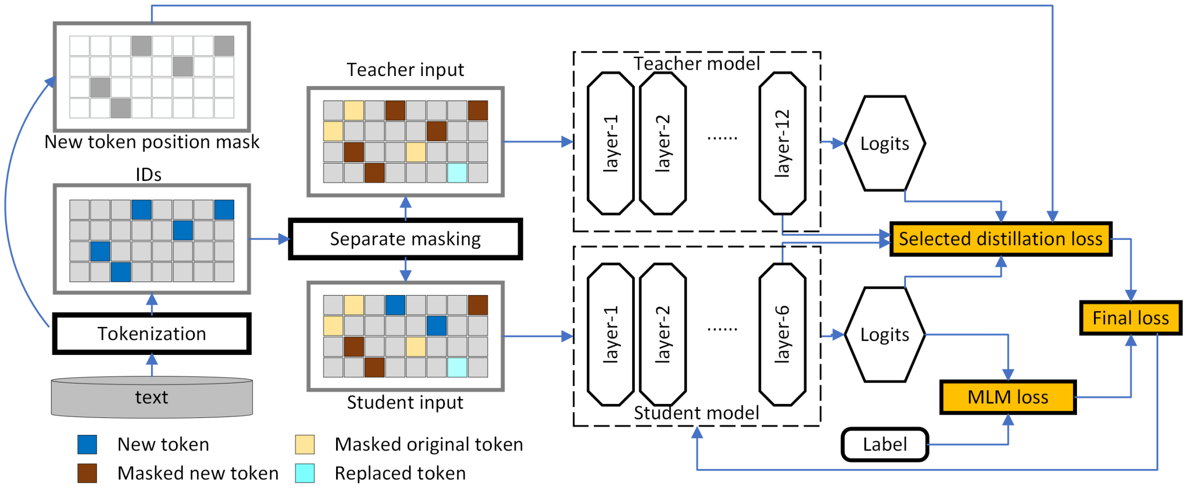


Figure 1: Architecture of VE-KD. New and original tokens are processed separately during tokenization, masking, and loss calculation. The student model soaks up two types of knowledge: common knowledge via original tokens and domain-specific knowledge via new tokens.

a simplistic sequential approach may not yield the best results. First, both tasks require high-quality data, which can be difficult to obtain. Second, using general methods such as domain-adaptation followed by distillation or distilling an already domain-adapted model requires two or more steps or training and hyperparameter tuning (Yao et al., 2021), which makes the learning process difficult to optimize.

During the domain-adaptation phase (secondary-stage unsupervised pre-training), there is a significant risk of losing general knowledge due to overfitting when a small corpus is used. Moreover, 2-step training requires more computational resources and time, possibly requiring further iterations to achieve the most effective outcomes. Therefore, a method that can proficiently perform domain adaptation and model compression simultaneously is needed to overcome these issues.

In this paper, we propose VE-KD, a novel method 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 a corpus, is robust and easy to optimize, and has lower requirements in terms of computational resources and time.

2 Related Work

Large pre-trained models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) have become ubiquitous in the NLP field (Ramponi and Plank, 2020). In terms of domain shifts, secondary-stage unsupervised pre-training on a 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) and Gururangan et al. (2020). Meanwhile, Lee et al. (2020) performed continual pre-training to adapt the BERT model to the biomedical domain, utilizing both 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 terms 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) in order 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. 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) includes self-attention distributions and hidden-states for training the student model. MiniLMv2 (Wang et al., 2021) 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. The AD-KD approach (Wu et al., 2023) explores the token-level rationale behind the teacher model based on Integrated Gradients and transfers attribution knowledge to the student model.

Several frameworks for general knowledge distillation using LLMs have also been proposed. GKD (Tan et al., 2023) is a general-knowledge distillation framework that supports distillation on larger-scale PLMs using various distillation methods and f-DISTILL (Wen et al., 2023) formulates sequence level knowledge distillation through minimization of a generalized f-divergence function. Hsieh et al. (2023) and Li et al. (2023) have proved that distillation using extracted rationales from black box LLMs is effective.

In this paper, we focus on task-agnostic knowledge distillation approaches, where a smaller distilled pre-trained model can be directly fine-tuned on downstream tasks.

3 Methods

In this study, we propose VE-KD, a knowledge distillation method with vocabulary expansion, as shown in Figure 1. Unlike Adapt-and-Distill (Yao et al., 2021), which requires 4-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. By continually distilling knowledge from the teacher model, VE-KD effectively avoids overfitting, a common issue that arises during the domain adaptation phase, especially when working with small corpora.

In the knowledge distillation aspect of VE-KD, a larger BERT model serves as the teacher model, instructing a smaller student model. Through the distillation process, the student model learns 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 (which 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 enable simultaneous learning of domain knowledge from the corpus as well as general knowledge from the teacher model via 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 vocabulary expansion. Building on the research of Yao et al. (2021), we chose a vocabulary size of 60,000 words.

3.2 Tokenization and Separate Token Masking

The process of separating two terms is accomplished through tokenization and token masking. Typically, model distillation necessitates that both the teacher and student models possess identical dictionaries. However, because of 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 the unique mask method shown 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 that has been 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. We replace them with a [MASK] token as new input, as follows:

$$x_{\text{input}} = [[\text{MASK}], x_2, [\text{MASK}] ..., x_n]. \quad (233)$$

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

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

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

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 as well as the token prediction logits $L_{t,s}$.

At the new token position, the output logits and the hidden-state vectors of the teacher model conflict with the student model because the student model has a bigger vocabulary and new knowledge. To learn the knowledge of the teacher model successfully, similarity calculations are made only 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 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 learning of the teacher model's knowledge.

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

The new token may lead to conflict between the KD loss and MLM loss even if the calculation range is split. . Knowledge about general terms may differ between the teacher model and student model because the meaning or grammar of general terms around the new token may differ. Because taking 100% of the knowledge from the teacher model might have adverse effects on creating new domain knowledge for the student model, we use tolerance to control the KD loss as follows:

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

Here, $\mathcal{L}_{\text{KD}}$ 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 0 and refrain from further optimization for lower loss. If a conflict arises, the student model will first optimize the MLM loss. Thereby ensuring that the student model learns the new domain knowledge 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, as follows:

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

Here, 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 used BLURB[2] (see Appendix A) for performance evaluation.

For the biomedical domain, we gathered a small corpus from PubMed (1.3GB) abstracts and compared it with PubMedBERT, which used a 21-GB corpus for pre-training. We omitted any abstracts containing fewer than 128 words in order 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). We adhere to the same evaluation and hyperparameter (see Appendix B) as those used by PubMedBERT following Yasunaga et al. (2022).

4.2 Implementation

We use the uncased version of $\text{BERT}_{\text{BASE}}$[3] (12 layers, 768 dimensions) as the teacher model and the baseline. We perform distillation of BERT to a small student model[4] (6 layers, 768 dimensions) with vocabulary expansion. The weights of the student model's layers is initialized with those of the

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

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

[4] Our models, evaluation data and training code are available at: https://github.com/pZvfkv3t8PA9vAc/VE-KD

	BERT-base	BERT-base_DA[5]	DistilBERT PubMed	Adapt-and-Distill	VE-KD
Layers Number	12	12	6	6	6
NER					
BC5CDR-chem	89.25	90.96	88.81	89.40	**89.83**
BC5CDR-disease	81.44	82.90	78.94	**82.25**	81.65
NCBI-disease	85.67	85.64	84.07	85.01	**86.50**
BC2GM	80.90	80.91	79.94	79.61	**80.03**
JNLPBA	77.69	77.20	76.64	**76.57**	76.34
PICO extraction					
ebmnlp	72.34	73.26	71.22	71.03	**72.08**
Relation extraction					
chemprot	71.86	72.64	70.77	68.16	**69.28**
DDI	80.04	80.64	74.20	**76.78**	76.69
GAD	80.41	79.40	78.29	**79.31**	77.82
Document classification					
HoC	80.20	81.37	80.76	81.64	**83.21**
Question answering					
Pubmedqa	51.62	56.20	53.40	54.00	**55.80**
BioASQ	70.36	66.43	67.86	72.86	**75.71**
Average of all tasks	76.82	77.30	75.41	76.38	**77.08**
Macro-average	74.79	75.41	73.74	74.68	**75.70**

Table 1: Comparison with distillation models trained by the PubMed corpus. DistilBERT$_{\text{PubMed}}$: using the same method with DistilBERT, Adapt-and-Distill: using the same method with Yao et al. (2021), VE-KD: using our method. Bold indicates the best performance of 6-layer models.

teacher model's layers 0, 2, 4, 7, 9, and 11. Additionally, we perform distillation of BERT by following the normal method which uses the same corpus and hyperparameters as a DistilBERT$_{\text{PubMed}}$ (Sanh et al., 2019) and a Adapt-and-Distill model (Yao et al., 2021) (Appendix C). The mid product of Adapt-and-Distill method, BERT-base_DA[5] obtained by domain adaptation from BERT-base is also the baseline.

Additionally, we chose some small (6 layers) BERT or distilled BERT models for general purposes, including BERT$_{\text{L6H768}}$[3](6 layers, 768 dimensions), TinyBERT, MiniLMv2 and DistilBERT$_{\text{wiki}}$. For comparison with domain adaptation ability, we additionally trained these models using the PubMed corpus with an MLM task.

We also performed distillation experiments across different dimensions, such as from BERT-large to a smaller student model (6 layers, 384 dimensions) as shown in Appendix D.

We made additional attempts, applying the VE-KD method to generative language models, such as GPT2 (Radford et al., 2019) and T5 (Raffel et al., 2020). Unfortunately, the results did not meet expectations, showing some performance drop compared to traditional distillation methods. Detailed data will be provided in the Appendix E.

4.3 Comparison with BERT, DistilBERT and Adapt-and-Distill

The results for the performance comparison of the distillation model using the same PubMed corpus are shown in Table 1. VE-KD outperformed teacher model BERT on 6 tasks, and showed an improved performance of 0.3% on average and 0.9% on Macro-average. VE-KD outperformed DistilBERT$_{\text{PubMed}}$ on 10 tasks, showed an increased absolute performance of 1.6% on average and 2.0% on Macro-average. VE-KD outperformed Adapt-and-Distill on 8 tasks, showed an increased absolute performance of 0.7% on average and 1.0% on Macro-average. VE-KD obtained the highest Macro-average score including BERT-base_DA[5] (+0.3%).

Moreover, we observed 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), as well as significant improvements in document classification (+2% on HoC) and question answering (+6% on PubMedQA, +13% on BioASQ) compared with DistilBERT, document classification (+2% on HoC) and question answering (+2% on PubMedQA, +3% on BioASQ) compared with Adapt-and-Distill. Document classification and question answering are tasks that require a deep understanding of sen-

[5]"DA" indicates that the model is under domain adaptation.

	$\text{BERT}_{\text{L6H768}}$		TinyBERT		MiniLMv2		DistilBERT wiki		DistilBERT PubMed		VE-KD
Domain Adaptation	*o*	*w*	*o*	*w*	*o*	*w*	*o*	*w*	*o*	*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
Macro-average	74.56	**75.91**	73.37	74.18	73.03	75.51	73.74	74.18	73.26	75.04	75.70

Table 2: Comparison of small models, where w indicates domain adaptation and o indicates no domain adaptation. Bold indicates the best performance, and underline indicates the second best.

tences. We propose that our method excels in tasks that focus on understanding the meaning of sentences, rather than tasks that involve token-level information extraction.

BERT-base_DA[5] which is the mid product of Adapt-and-Distill potentially suffered from overfitting due to small corpora. Compare with $\text{BERT}_{\text{L6H768}}$_DA[5] in Table 2, we observed that the larger model yields worse performance. This could be attributed to the fact that larger models require larger corpora to avoid overfitting. Consequently, when dealing with smaller corpora, the performance of larger models may be compromised. In contrast, VE-KD, offers improved stability because it does not require multiple phases like Adapt-and-Distill. This stability enables VE-KD to better address the challenges of overfitting in domain adaptation scenarios.

VE-KD did not perform as well in the relation-extraction task as the other 6-layer models did, experiencing an average performance decrease of 3% compared with BERT-base. This might be attributable to the 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 Comparison with Models Having the Same Layer Size

Table 2 shows the results of performance comparison versus the small model having the same layers and hidden-state size as VE-KD. Compared with small models without domain adaptation, VE-KD achieves the highest performance on average. Even after domain adaption, VE-KD is still the second-highest model just behind $\text{BERT}_{\text{L6H768}}$. Compared with $\text{DistilBERT}_{\text{PubMed}}$_DA[5], which uses the same corpus, VE-KD also attains a 1.1% performance increase on average and 0.7% on Macro-average, and in particular obtains a 2% increase for PubMedQA and 3.6% in BioASQ 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 analyze 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 and PubMedQA tasks. 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 used in this study (8 A100 GPUs).

As shown in Table 3 for the HoC and PubMedQA tasks, VE-KD outperforms BERT in the HoC task after 3 h of training and surpasses BioBERT and PubMedBERT after 6 and 9 h of training, respectively. For the PubMedQA task, VE-KD outperforms BERT and PubMedBERT after 6 and 9 h of training, respectively. These observations highlight the efficiency of our method, which can match or surpass the performance of models pre-trained from scratch, all while using 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 Adapt-and-Distill method. Compared with fine-tuned DistilBERT and BERT, VE-KD achieves a higher performance while using only about half the training time. In comparison with Adapt-and-Distil, VE-KD achieves a higher performance while using only about 15% the training time.

Model	Training Time	Corpus Words	HoC	PubMed QA
VE-KD	3 h	0.2B	81.64	54.00
	6 h	0.2B	81.74	55.30
	9 h	0.2B	**82.64**	<u>56.60</u>
DistilBERT	9 h	0.2B	80.76	53.40
DistilBERT_DA[5]	19 h	0.2B	81.21	53.80
BERT_DA[5]	25 h	0.2B	81.37	56.20
Adapt-and-Distil	62 h	0.2B	81.64	54.00
BERT	0 h	3.3B	80.20	51.62
BioBERT	240 h	4.5B	81.54	**60.24**
PubMedBERT	240 h	3.1B	<u>82.32</u>	55.84

Table 3: Results with different model training. Bold and underline indicate the first best and the second best, respectively.

5.2 Impact of Vocabulary Size

To understand the impact of vocabulary size, we conduct 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 of models deliver the best results with a vocabulary size of 60,000 words in our study. Interestingly, models with larger vocabularies of 70,000 and 80,000 words do not exhibit better performance but instead exhibit a significant performance loss. A reasonable explanation for these results is that a larger vocabulary set might potentially include more complex but less common tokens, which cannot be sufficiently learned through continuous pre-training, especially in a small-scale corpus.

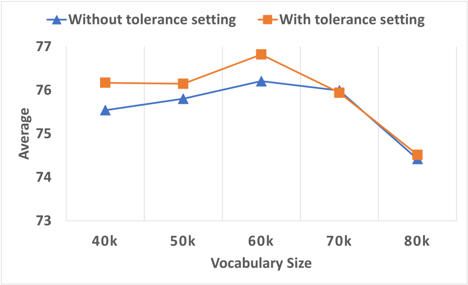


Figure 2: Average performance of VE-KD with different vocabulary sizes.

5.3 Impact of Tolerance

To understand the impact of tolerance, we conducted several experiments in which the tolerance is adjusted within a 60,000-word vocabulary by utilizing HoC, PubMedQA, and BioASQ and then averaged 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, and 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 as a softer approach in the future.

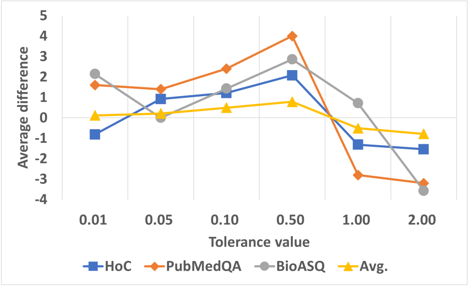
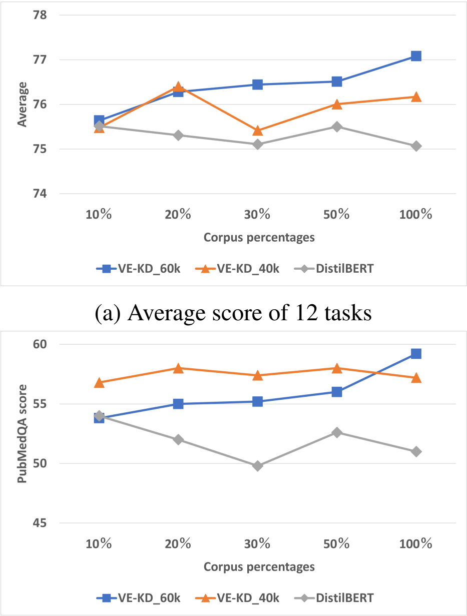


Figure 3: HoC, PubMedQA, BioASQ and the average performance of VE-KD with different tolerances.

5.4 Smaller Corpus

To understand the potential of our method on smaller corpora, we conducted several experiments on VE-KD (with 40,000 and 60,000-word vocabularies) and DistilBERT trained on various percentages of the PubMed corpus.

Figure 4 shows the performance-evaluation results for the average score and the PubMedQA task. We observe that when VE-KD_40k[6] and VE-KD_60k[6] trained on more than 20% of the corpus, the VE-KD_40k had larger fluctuations in average score compared with VE-KD_60k at the same time. Interestingly, for the PubMedQA task, VE-KD_60k performed worse than VE-KD_40k up until reaching 100% of the dataset. One potential explanation for this is that the VE-KD_60k 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.

5.5 Inference Speed and Model Size

We compare the parameter size and inference speed of VE-KD with the BERT model and DistilBERT, and the results are shown in Table 4. Compared with BERT-base, the half-layers of 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, in line with the results of Yao et al. (2021).

For the VE-KD_40k and VE-KD_60k yields about 8 million and 22 million parameters, respectively, in the tokenization weights. The model lightening effect is thus smaller. For further model lightening, it may be necessary to have smaller hidden dimensions, few layers, or lower numbers of attention heads.

Model	Parameters	Speedup
BERT	110 M	×1.00
DistilBERT	67 M	×1.48
VE-KD_40k	75 M	×1.50
VE-KD_60k	90 M	×1.56

Table 4: Comparison of parameter size and inference speed. The inference speed was tested by the EBM PICO task and evaluated on a single RTX 6000 GPU.

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. Our experimental results demonstrate that VE-KD is effective; Its performance is competitive with well-known models such as BioBERT and PubMedBERT, and its training efficiency is noteworthy. It outperforms DistilBERT and Adapt-and-Distill method, especially in document-level tasks. Furthermore, VE-KD is more robust compared to general domain adaptation. VE-KD using distillation mechanism which can avoid overfitting, especially work with small corpora.

We thoroughly investigated the effects of vocabulary size and tolerance and obtained insights that can help us configure more efficient models. Because of its efficiency across various domain-specific NLP tasks, VE-KD lays the groundwork 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. Another limitation is that we have not yet fully explored the applicability of the VE-KD method to other model structures. For instance, we are looking for more efficient and fitting methods of knowledge segmentation and aggregation for generative language models.

[6] VE-KD_40k and VE-KD_60k denote VE-KD with 40,000 and 60,000-word vocabulary sizes, respectively.

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.

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*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.

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.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. 2023. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2665–2679, Toronto, Canada. Association for Computational Linguistics.

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*.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6):bbac409.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

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.

Shicheng Tan, Weng Lam Tam, Yuanchun Wang, Wenwen Gong, Shu Zhao, Peng Zhang, and Jie Tang. 2023. GKD: A general knowledge distillation framework for large-scale pre-trained language model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 134–148, Toronto, Canada. Association for Computational Linguistics.

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, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021. MiniLMv2: Multi-head self-attention relation distillation for compressing pre-trained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2140–2151, Online. Association for Computational Linguistics.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. 2023. f-divergence minimization for sequence-level knowledge distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10817–10834, Toronto, Canada. Association for Computational Linguistics.

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 Fine-tuning Dataset

BLURB 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 5.

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 5: numbers of instances included in the BLURB biomedical NLP benchmark datasets we used.

B Hyperparameter Details

For all distillations, including the baseline with the same default training seed, we train for 5 epochs by using batch size of 240 and a peak learning rate of 5×10^{-4}, which is warmed up in the first 10% of steps and then is decayed linearly.

For all domain adaptation, including the baseline with same defaults training seed, we train for 100,000 steps by using batch size of 80 and a peak learn rate of 5×10^{-4}, which is warmed up in the first 10% of steps and then is decayed linearly.

For BLURB fine-tuning, including the baseline with same defaults training seed, we set max_seq_length to 512 and choose learning rates from 1×10^{-5}, 2×10^{-5}, 3×10^{-5}, 5×10^{-5}, 6×10^{-5}, batch sizes from 16, 32, 64 and epochs from 1 to 120.

C Experiment Using Adapt-and-Distill Method

To comparison with Adapt-and-Distill (Yao et al., 2021) method, we conducted a comparative experiment using same method. The Adapt-and-Distill method comprises four steps:

1. perform domain adaptation for the teacher model T to T'.
2. perform distillation from the teacher model T to the student model S,
3. perform domain adaptation for the student model S to S'.
4. perform distillation from the teacher model T' to S'', using the intermediate student model S' as initialization.

D Distillation with Different Dimensions

When the output dimensions of the teacher model and the student model differ, we add a learnable transformation $W_h \in \mathbb{R}^{d' \times d}$ to convert hidden-state vectors to the same dimensions as

$$H'_s = H'_s W_h.$$

But initialization from teacher to student is not available because of different dimensions. We randomly initial student model.

D.1 Distillation from BERT-large

To investigate the effect of this method on larger models, we also conducted a comparative experiment on BERT-large.

BERT-large has 24 layers of 1,024 hidden dimensions. We distilled it to 6 layers of 768 and 384 dimensions, respectively. using VE-KD method and DistilBERT method.

The results for the performance comparison using the same PubMed corpus are shown in Table 6, which shows that VE-KD$_{768}$ outperforms teacher model BERT-large only on 2 tasks. VE-KD$_{768}$ outperforms DistilBERT on 8 tasks, achieving an increased absolute performance of 1.5% on average. VE-KD$_{768}$ obtained the highest Macro-average score. VE-KD$_{384}$ exhibited a considerable drop in performance, possibly due to the simplicity of the transformation method that was used.

Compared with the performance of BERT-base, BERT-large is harder to distill, implying that it requires a larger corpus to achieve comparable performance.

	BERT-large	DistilBERT	VE-KD	VE-KD
Hidden dimension	1024	768	768	384
NER				
BC5CDR-chem	**90.45**	88.43	88.75	87.18
BC5CDR-disease	**82.17**	79.27	79.10	78.09
NCBI-disease	**85.57**	84.09	82.64	82.37
BC2GM	**81.23**	77.75	77.64	76.27
JNLPBA	**77.89**	75.26	75.27	72.96
PICO extraction				
ebmnlp	**72.23**	70.51	70.98	70.10
Relation extraction				
chemprot	**72.52**	65.73	68.28	64.08
DDI	**82.35**	70.28	72.96	67.97
GAD	75.00	**79.09**	78.99	76.68
Document classification				
HoC	78.37	80.54	**80.62**	77.14
Question answering				
Pubmedqa	50.80	**53.60**	52.80	50.00
BioASQ	67.14	65.71	**80.00**	62.86
Average of all tasks	**76.31**	74.19	75.67	72.14
Macro-average	73.93	72.67	**74.42**	70.52

Table 6: Comparison of models distilled from BERT-large. Bold indicates the best performance, and underline indicate the second best.

D.2 Distillation to Smaller Hidden Dimensions

We investigated the effect of this method on smaller models, and conducted a comparative experiment to 6 layers of 384 hidden dimensions from BERT-base, using VE-KD method, DistilBERT method, and Adapt-and-Distill method.

Table 7 shows the results for the performance comparison using the same PubMed corpus. Compared with BERT-base with 768 dimensions, the model with 384 dimensions presents a significant challenge to surpass. Both our 1-step VE-KD method and the 4-step Adapt-and-Distill method outperforms DistilBERT.

The Adapt-and-Distill method outperforms VE-KD by about 1% on average, and by 0.5% on Macro-average. This difference in performance could potentially be attributed to the initialization of the student model. Further investigation into the initialization process may help shed light on this performance difference between the two methods.

By incorporating an additional domain adaptation to VE-KD using same hyperparameter with third step of Adapt-and-Distill method, we obtain VE-KD_DA[5] (2-step), which achieved performance comparable to the Adapt-and-Distill model with less training time and fewer computational resources. It suggesting that VE-KD may be underfitting across different dimensions. Further exploration and analysis could provide insights into the underlying factors contributing to this underfitting in VE-KD.

E Distillation on Models with Having Different Architecture

To investigate the effect of this method on models having different architecture, we also conducted comparative experiments involving GPT2[7] and T5-small[8]. We used the same tasks as in the pre-training phase:

- GPT2: predict-the-next-word task
- T5: fill-in-the-blank task

E.1 Separate Token Masking

As we did for BERT, we separated the knowledge by using separate token masking. The absence of the [MASK] token in the vocabulary of GPT2 and T5 necessitates replacing it with other tokens.

- GPT2: using the [UNK] token as the [MASK] token:

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

$$x_{\text{label}} = [x_1, x_2, x_3 ..., x_n]$$

- T5: using sentinel tokens [extra_id_0] ~[extra_id_99] as the [MASK] token:

[7] https://huggingface.co/openai-community/gpt2

[8] https://huggingface.co/google-t5/t5-small

	BERT-base	DistilBERT	VE-KD	VE-KD_DA[5]	Adapt-and-Distill
Hidden dimension	768	384	384	384	384
NER					
BC5CDR-chem	**89.25**	86.95	87.39	88.05	89.00
BC5CDR-disease	**81.44**	77.42	78.53	79.36	80.49
NCBI-disease	**85.67**	82.75	83.19	82.09	83.58
BC2GM	**80.90**	76.47	77.12	77.28	77.93
JNLPBA	**77.69**	73.88	72.92	74.16	75.16
PICO extraction					
ebmnlp	**72.34**	69.02	69.72	70.62	70.75
Relation extraction					
chemprot	**71.86**	62.04	64.03	65.29	66.67
DDI	**80.04**	67.80	67.10	68.95	68.24
GAD	**80.41**	76.60	76.14	78.99	77.60
Document classification					
HoC	**80.20**	75.90	79.40	78.69	78.56
Question answering					
Pubmedqa	51.62	50.40	**53.60**	52.60	49.00
BioASQ	70.36	66.43	67.86	67.86	70.71
Average of all tasks	**76.82**	72.14	73.08	73.66	73.97
Macro-average	**74.79**	70.33	71.75	72.16	72.25

Table 7: Comparison of models with smaller hidden dimensions. Bold indicates the best performance, and underline indicate the second best.

$$x_{\text{input}} = [[\text{extra\_id\_0}], x_2, [\text{extra\_id\_1}], ..., x_n].$$

$$x_{\text{label\_s}} = [[\text{extra\_id\_0}], x_1, [\text{extra\_id\_1}], x_3].$$

$$x_{\text{label\_t}} = [[\text{extra\_id\_0}], x_1[0], x_1[1], [\text{extra\_id\_1}], x3].$$

Because of the differences between the tokenizers used by the teacher and the student, the length of the labels varies between them. For instance, a domain-specific token such as x1, which is recognized by the teacher's tokenizer, may be divided into multiple tokens, such as $x_1[0], x_1[1]$.

Masked tokens for GPT2 and T5 constructed with the 50% to 50% proportion from the general and domain-specific terms.

E.2 Experimental Setting

For domain-specific terms (domain knowledge), we used pre-training tasks to extract knowledge from the corpus. For general terms (and general knowledge), we used the same similarity loss function with Distil-BERT. As with VE-KD for BERT, similarity calculations were made only within the scope of general terms (without the new token position for GPT2, and without the new token or sentinel token for T5).

E.3 Result and Analysis

Regrettably, the defined usage of the VE-KD method for the generation of language models did not enhance the performance of our student models, as Table 8 and Table 9 show.

Compared with GPT2, GPT2$_{\text{Distillation}}$ achieved nearly identical performance. However, GPT2$_{\text{VE-KD}}$ experienced a 4% degradation in performance, suggesting that the [UNK] token may not be adequately trained to handle masked token problems. Hence, a more optimal model design or the use of distinct alternate tokens is needed. This would allow GPT-2 to improve its learning capacity, both from the teacher's knowledge and corpus, concurrently.

Similarly, compared with T5, T5$_{\text{Distillation}}$ achieved nearly the same performance levels. Nevertheless, T5$_{\text{VE-KD}}$ saw only a minor reduction in performance of about 1%. This result indicates that T5's fill-in-the-blank task might be more suitable for VE-KD's concept than for GPT2's predict-the-next-word task. A potential explanation for this decrease in performance is the dissimilarity in label length, which leads to different position embeddings, thereby causing confusion for the model. However, to surpass the performance of the teacher model, more in-depth exploration may be necessary.

	GPT2	GPT2 distillation	GPT2 VE-KD
NER			
BC5CDR-chem	**75.69**	75.05	68.55
BC5CDR-disease	64.34	**65.59**	60.24
NCBI-disease	**67.94**	67.44	61.00
BC2GM	**57.68**	56.62	53.67
JNLPBA	59.60	**61.01**	57.84
PICO extraction			
ebmnlp	66.87	**67.24**	64.28
Relation extraction			
chemprot	**68.71**	66.31	63.03
DDI	**68.55**	68.07	62.47
GAD	**80.13**	75.59	74.52
Document classification			
HoC	80.48	**81.03**	76.76
Question answering			
pubmedqa	**53.80**	53.00	50.60
BioASQ	64.29	68.57	**70.00**
Average of all tasks	**67.34**	67.13	63.58
Macro-average	68.78	**68.84**	65.66

Table 8: Comparison of GPT2 models using different methods. Bold indicates the best performance.

	T5	T5 distillation	T5 VE-KD
NER			
BC5CDR-chem	**84.68**	83.22	81.03
BC5CDR-disease	67.21	**71.45**	67.83
NCBI-disease	**81.16**	77.69	77.25
BC2GM	**75.48**	71.20	70.29
JNLPBA	62.34	**66.67**	63.98
PICO extraction			
ebmnlp	59.86	**62.16**	59.08
Relation extraction			
chemprot	**57.47**	54.84	54.43
DDI	58.46	**58.60**	55.51
GAD	77.54	76.42	**78.49**
Document classification			
HoC	75.48	**75.58**	74.66
Question answering			
pubmedqa	55.20	**56.40**	56.20
BioASQ	67.14	70.71	**77.14**
Average of all tasks	68.50	**68.74**	67.99
Macro-average	67.04	**67.73**	67.06

Table 9: Comparison of T5 models using different methods Bold indicates the best performance.