AVocaDo: Strategy for Adapting Vocabulary to Downstream Domain

Jimin Hong[*], **Taehee Kim**[*], **Hyesu Lim**[*] and **Jaegul Choo**
Korea Advanced Institute of Science and Technology (KAIST)
`{jimmyh, taeheekim, hyesulim, jchoo}@kaist.ac.kr`

Abstract

During the fine-tuning phase of transfer learning, the pretrained vocabulary remains unchanged, while model parameters are updated. The vocabulary generated based on the pretrained data is suboptimal for downstream data when domain discrepancy exists. We propose to consider the vocabulary as an optimizable parameter, allowing us to update the vocabulary by expanding it with domain-specific vocabulary based on a tokenization statistic. Furthermore, we preserve the embeddings of the added words from overfitting to downstream data by utilizing knowledge learned from a pretrained language model with a regularization term. Our method achieved consistent performance improvements on diverse domains (i.e., biomedical, computer science, news, and reviews).

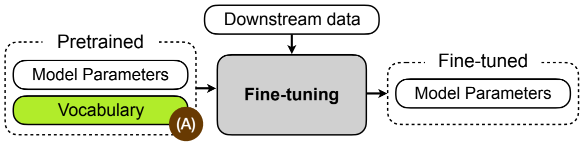
1 Introduction

A language model (LM) is pretrained with a large corpus in a general domain and then is fine-tuned to perform various downstream tasks, such as text classification, named entity recognition, and question answering. However, fine-tuning the LM is challenging when the downstream domain is significantly different from the pretrained domain, requiring domain adaptation to improve the downstream performance [Gururangan et al., 2020, Lee et al., 2020, Beltagy et al., 2019].

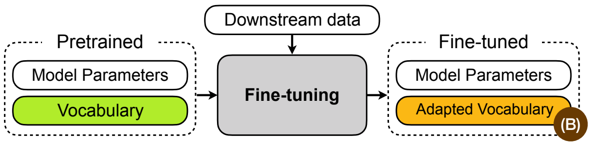
Prior approaches conducted additional training with a large domain-specific corpus in between pretraining and fine-tuning. In these approaches, the pretrained vocabulary remains unchanged, although the model is being adapted to a downstream domain, such as biomedicine or politics.

We argue that the vocabulary should also be adapted during the fine-tuning process towards downstream data. Recent studies (e.g., SciBERT [Beltagy et al., 2019]) showed that using an

[*]Equal Contribution



(a) Fine-tuning only model parameters



(b) Fine-tuning both model parameters & vocabulary (**AVocaDo**)

Domain Word	Tokenized with (A)	Tokenized with (B)
bluetooth	blue ##tooth	bluetooth
corticosterone	co ##rti ##cos ##ter ##one	cor ##tic ##osterone
disrespectful	di ##sr ##es ##pe ##ct ##ful	disrespect ##ful

(c) Examples of tokenization

Figure 1: **Overview of AVocaDo.** AVocaDo updates the vocabulary (b) not only fine-tuning the model parameters as done by previous approaches (a). fine-tuning the vocabulary has benefit on tokenizing domain-specific words (c).

optimized vocabulary for a particular downstream domain is more effective than using the vocabulary generated in pretraining stage. However, these approaches required a large domain-specific corpus additional to the downstream data in order to construct optimized vocabulary for the downstream domain.

We propose to **A**dapt the **Voca**bulary to downstream **Do**main (**AVocaDo**), which updates the pretrained vocabulary by expanding it with words from the downstream data without requiring additional domain-specific corpus. The relative importance of words is considered in determining the size of the added vocabulary. As shown in Figure 1-(c), domain-specific words are tokenized in unwilling manner in the corresponding domain. For example, in reviews domain, the "bluetooth" represents a short-range wireless technology standard, but when the word is tokenized into "blue" and "tooth", the

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4692–4700
November 7–11, 2021. ©2021 Association for Computational Linguistics

combined meaning of each subword is totally different from the intended meaning of "bluetooth". Furthermore, we propose a regularization term that prevents the embeddings of added words from overfitting to downstream data, since downstream data is relatively small compared to the pretraining data.

The experimental results show that our proposed method improves the overall performance in a wide variety of domains, including biomedicine, computer science, news, and reviews. Moreover, the advantage of the domain adapted vocabulary over the original pretrained vocabulary is shown in qualitative results.

2 Related Work

As transfer learning has shown promising results in natural language processing (NLP), recent work leveraged the knowledge learned from the pretrained model, such as BERT [Devlin et al., 2018] in various domains.

SciBERT [Beltagy et al., 2019] trains a language model with the large domain-specific corpus from scratch, showing that the vocabulary constructed from the domain-specific corpus contributes to improving performance. Lee et al. [2020] and Gururangan et al. [2020] conducted additional training on a pretrained LM with a large domain-specific corpus before fine-tuning. On the other hand, exBERT [Tai et al., 2020] extended the pretrained model with new vocabulary to adapt to biomedical domain. Similarly, Poerner et al. [2020] and Sato et al. [2020] proposed to expand vocabulary and leverage external domain-specific corpus to train new embedding layers.

On the contrary, AVocaDo requires only downstream dataset in domain adaptation. Furthermore, our method selects a subset of domain-specific vocabulary considering the relative importance of words.

3 Methods

In **AVocaDo**, we generate domain-specific vocabulary based on the downstream corpus. The subset of the generated vocabulary is merged with the original pretrained vocabulary. The size of subset is controlled by the fragment score. Afterwards, we apply a regularization term during fine-tuning to prevent the embeddings of added words from overfitting to the downstream data.

Algorithm 1 Adapting Vocab in AVocaDo

```
Input: pretrained vocab V_P; corpus C;
        domain-specific vocab V_D.
Output: adapted vocab V_A.
Initialize hyperparamenters α, β, γ.
V_A ← V_P ∪ {V_D_i}_{i=0}^{α}
while f_C(V_A) > γ do
    V_A ← V_A ∪ {V_D_i}_{i=α}^{α+β}
    α ← α + β
end while
return V_A
```

3.1 Adapting Vocabulary

In this section, we describe the procedure of adapting the vocabulary to the downstream domain through Algorithm 1. First, the domain-specific vocabulary set $V_{\mathcal{D}}$ is constructed from the downstream corpus $\mathbf{C}$ given a vocabulary size $N_{\mathcal{D}}$ and a tokenizing algorithm. The adapted vocabulary set $V_{\mathcal{A}}$ is constructed by merging the subset of $V_{\mathcal{D}}$, size of $n_{\mathcal{D}}$, with the original pretrained vocabulary set $V_{\mathcal{P}}$, size of $N_{\mathcal{P}}$. In other words, $N_{\mathcal{A}}$, the size of $V_{\mathcal{A}}$, is equal to the sum of the merged vocabulary sets, i.e, $N_{\mathcal{A}} = n_{\mathcal{D}} + N_{\mathcal{P}}$. Note that $n_{\mathcal{D}} < N_{\mathcal{D}}$, because when too many words are added, the added infrequent subwords might cause the rare word problem [Luong et al., 2015, Schick and Schütze, 2020].

The subset of $V_{\mathcal{D}}$, that is added to $V_{\mathcal{P}}$, is determined by the *fragment score* $f_{\mathbf{C}}(V) \in \mathbb{R}$, which we introduce as a new metric that measures the relative number of subwords tokenized by a vocabulary V from a single word in corpus $\mathbf{C}$, i.e.,

$$f_{\mathbf{C}}(V) = \frac{\text{the number of subwords tokenized by } V}{\text{the number of words in } \mathbf{C}}. \quad (1)$$

Motivated by Rust et al. [2020], we keep $f_{\mathbf{C}}(V_{\mathcal{A}})$ from exceeding a certain threshold γ. γ is a hyperparameter determining the lower bound of the $f_{\mathbf{C}}(V_{\mathcal{A}})$. Decreasing the lower bound leads $V_{\mathcal{A}}$ to less finely tokenize $\mathbf{C}$. In contrast, increasing the lower bound leads $V_{\mathcal{A}}$ to finely tokenize $\mathbf{C}$.

We sought to consider the importance of subwords when adding $V_{\mathcal{D}}$. We simply selected a subset of $V_{\mathcal{D}}$ following the order of merging subwords used in byte pair encoding algorithm [Sennrich et al., 2015]. The number of added vocabulary in each iteration is indicated by the hyperparameters α and β.

In summary, as frequent subword pairs are added from $V_{\mathcal{D}}$ to $V_{\mathcal{A}}$ as subwords, the $f_{\mathbf{C}}(V_{\mathcal{A}})$ decreases.

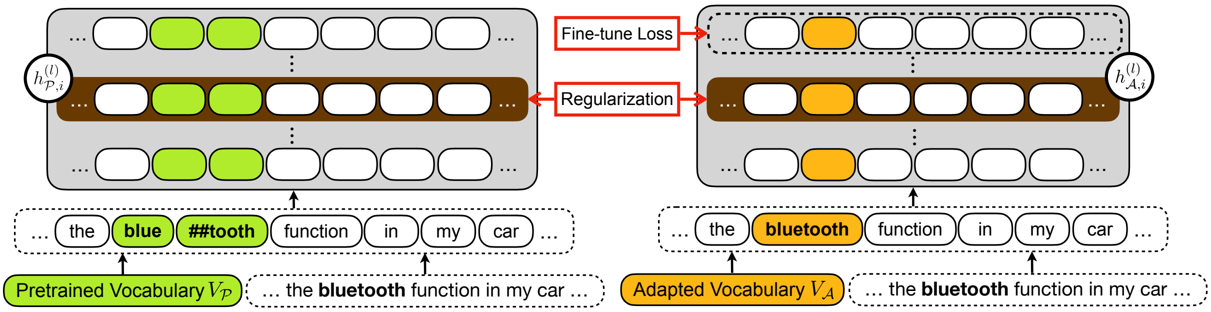


Figure 2: **Fine-tuning with regularization.** Identical sentence "... the bluetooth function in my car ...", sampled from AMAZON, is tokenized with pretrained vocabulary (left) and with adapted vocabulary (right). The domain-specific word "bluetooth" is tokenized in two ways, which are highlighted as green and yellow respectively. The model is fine-tuned with regularization on l-th layer, highlighted as brown box, to preserve the embeddings of added words (e.g., bluetooth) from overfitting to downstream dataset.

The objective of adding $V_\mathcal{D}$ to $V_\mathcal{A}$ is to decrease the $f_\mathbf{C}(V_\mathcal{A})$, but we make sure that $f_\mathbf{C}(V_\mathcal{A})$ does not become too small, i.e., lower than the threshold γ. Therefore, we continue to add $V_\mathcal{D}$ to $V_\mathcal{A}$ if $f_\mathbf{C}(V_\mathcal{A})$ is higher than γ, and terminate the merging step otherwise.

3.2 Fine-tuning with Regularization

The embeddings of words in the subset of $V_\mathcal{D}$ which is merged with $V_\mathcal{P}$ to construct the adapted vocabulary $V_\mathcal{A}$ are trained only with downstream data during fine-tuning. Since the size of downstream data is much smaller than that of the pretraining corpus, the embeddings trained only with the downstream data possibly suffer from overfitting. To prevent the potential overfitting, we leverage the pretrained contextual representation learned from a large corpus.

In contrastive learning [Chen et al., 2020], a pair of instances is encouraged to learn representations in relation to the similarity of the instances. We apply this contrastive learning framework as a regularization in fine-tuning. As described in Figure 2, an identical sentence is tokenized in two ways: one with the pretrained vocabulary $V_\mathcal{P}$ and the other with the adapted vocabulary $V_\mathcal{A}$. A minibatch consists of B input sentences $\mathbf{x} = \{x_1, \ldots x_B\}$. Each input x_i is tokenized with two types of vocabularies, and their l-th layer encoder outputs are denoted as $h^{(l)}_{\mathcal{P},i}$ and $h^{(l)}_{\mathcal{A},i}$. Note that they are encoded with a single encoder given the identical input sentence, but with different tokenizations. $h^{(l)}_{\mathcal{P},i}$ and $h^{(l)}_{\mathcal{A},j}$ are considered as a positive pair when $i = j$, and as a negative pair when $i \neq j$. The positive pair $h^{(l)}_{\mathcal{P},i}$ and $h^{(l)}_{\mathcal{A},i}$ are trained to maximize the agreement by the regularization term $\mathcal{L}_{\text{reg}}$ i.e.,

$$\mathcal{L}_{\text{reg}}(\mathbf{h}^{(l)}_\mathcal{A}, \mathbf{h}^{(l)}_\mathcal{P}) = -\frac{1}{B}\log\sum_{i=1}^{B}\frac{e^{(\text{sim}(h^{(l)}_{\mathcal{A},i}, h^{(l)}_{\mathcal{P},i})/\tau)}}{\sum_{j=1}^{B} e^{(\text{sim}(h^{(l)}_{\mathcal{A},i}, h^{(l)}_{\mathcal{P},j})/\tau)}}, \tag{2}$$

where τ is a softmax temperature, B is a batch size, $\mathbf{h}^{(l)}_\mathcal{A} = \{h^{(l)}_{\mathcal{A},1}, \ldots, h^{(l)}_{\mathcal{A},B}\}$ and $\mathbf{h}^{(l)}_\mathcal{P} = \{h^{(l)}_{\mathcal{P},1}, \ldots, h^{(l)}_{\mathcal{P},B}\}$. The cosine similarity function is used for $\text{sim}(\cdot)$. $\mathbf{h}^{(l)}_\mathcal{A}$ is prevented from overfitting by making it closer to its positive sample.

The model is trained to perform the target task with the regularization term $\mathcal{L}_{reg}$. The output of the encoder with $V_\mathcal{A}$ is supervised by the label of downstream data with cross entropy loss $\mathcal{L}_{CE}$. The total loss $\mathcal{L}$ for domain adaptive fine-tuning is formalized as

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda\mathcal{L}_{reg}, \qquad \mathcal{L}_{CE} = -\frac{1}{B}\sum^{B}\sum_{i=1}^{C} t_i \log(f(s_i)), \tag{3}$$

where f is a softmax function, C is the total number of classes, s_i is the logit for i-th class, B is the batch size, and t_i is the target label. In our implementation, we set λ as 1.0 for all experiments.

4 Experimental Settings

Datasets We conducted experiments on four domains that are significantly different from the pretraining domain; biomedical (BIOMED) papers, computer science (CS) papers, NEWS, and amazon reviews (REVIEWS). CHEMPROT [Kringelum et al.], ACL-ARC [Jurgens et al.], HYPERPARTISAN [Kiesel et al., 2019], and AMAZON [McAuley et al., 2015] datasets are used in respective domains. Target task for each dataset is text classification. Appendix C describes more details.

Domain	Dataset	$\text{BERT}_{\text{base}}$	$\text{BERT}_{\text{AVocaDo}}$	SciBERT	$\text{SciBERT}_{\text{AVocaDo}}$	BioBERT	$\text{BioBERT}_{\text{AVocaDo}}$
BioMed	ChemProt	79.38	**81.07**(+1.69)	82.16	**82.71**(+0.55)	83.58	**84.42**(+0.84)
CS	ACL-ARC	56.82	**67.28**(+10.46)	66.89	**75.02**(+8.13)	-	
News	HyperPartisan	84.51	**89.31**(+4.80)	-	-	-	-
Reviews	Amazon	55.50	**68.51**(+13.01)	-	-	-	-

Table 1: **Comparisons with baselines in four different domains.** Pretrained LMs (i.e., $\text{BERT}_{\text{base}}$, SciBERT, and BioBERT) are fine-tuned in two ways: one with pretrained vocabulary (represented without a subscription) and the other with adapted vocabulary (represented with subscription AVocaDo). The performance improvement is represented inside the parentheses with $+$. The reported value is averaged F_1 score (micro-F_1 for ChemProt and macro-F_1 for the others) over five random seeds. Invalid comparisons are represented as -.

Evaluation Protocol We report the macro-F_1 score for ACL-ARC, HyperPartisan, and Amazon and micro-F_1 score for ChemProt as done by previous work [Lee et al., 2020, Beltagy et al., 2019]. The score is averaged over five random seeds.

5 Results

5.1 Quantitative Results

$\text{BERT}_{\text{base}}$ [Devlin et al., 2018], SciBERT [Beltagy et al., 2019], and BioBERT [Lee et al., 2020] are chosen as the pretrained LMs for our experiments. Each model is fine-tuned in two ways: one with pretrained vocabulary and the other with adapted vocabulary.

SciBERT is pretrained with scientific corpus while BERT is pretrained with general domain corpus (e.g., Wikipedia), and thus SciBERT can be fine-tuned only with BioMed and CS. BioBERT conducted additional training with biomedical corpus, so that BioBERT can be fine-tuned only with BioMed.

As described in Table 1, fine-tuning with AVocaDo significantly improved the performance of the downstream task in all domains. Note that the performance is improved despite the low-resource environment, where the size of dataset is smaller than 5,000 as described in Appendix C (ChemProt, ACL-ARC, and HyperPartisan). In BioMed domain, applying AvocaDo improved the overall performance in various pretrained language models. This improvement shows that utilizing the domain-specific vocabulary has additional benefits on the downstream domain. In CS, AVocaDo outperforms $\text{BERT}_{\text{base}}$ and SciBERT, showing the performance improvements of 10.46 in $\text{BERT}_{\text{base}}$ and 8.13 in SciBERT. In News and Reviews, our strategy significantly improved the performance; 4.80 in News and 13.01 in Reviews.

Domain	Domain Word	Pretrained Vocab $V_{\mathcal{P}}$	Adapted Vocab $V_{\mathcal{A}}$
BioMed	glucuronidation sulfhydration	g, lu, cu, ron, ida, tion sul, f, hy, dra, tion	glucuron, ida, tion sulf, hydr, ation
CS	nlp* syntactic	nl, p syn, ta, ctic	nlp syntactic
News	tweet disrespectful	t, wee, t di, sr, es, pe, ct, ful	tweet disrespect, ful
Reviews	otterbox thunderbolt	otter, box thunder, bolt	otterbox thunderbolt

Table 2: **Qualitative results.** Carefully selected tokenization examples from $V_{\mathcal{P}}$ and $V_{\mathcal{A}}$. $*$ represents capitalized in the original sentence.

5.2 Qualitative Results

To analyze the effectiveness of the adapted vocabulary $V_{\mathcal{A}}$, we show the sampled words from each domain that are tokenized with two types of vocabulary in Table 2.

The adapted vocabulary $V_{\mathcal{A}}$ tokenizes the domain-specific word into subwords that are informative in the target domain. For example, in the case of "sulfhydration", the word is tokenized as "sul, f, hy, dra, tion" with $V_{\mathcal{P}}$ and "sulf, hydr, ation" with $V_{\mathcal{A}}$. "sulf" and "hydr" imply "sulfur" and "water" respectively, which are frequently used in BioMed domain.

Furthermore, $V_{\mathcal{A}}$ preserves the semantic of a domain-specific word by keeping it as a whole word, where the subwords tokenized with $V_{\mathcal{P}}$ have completely different semantics from its original meaning. For instance, "otterbox" is an electronics accessory company in the Reviews domain. However, with $V_{\mathcal{P}}$, it is split into "otter" and "box", where the "otter" is a carnivorous mammal and "box" is a type of container. Randomly sampled tokenization examples from $V_{\mathcal{P}}$ and $V_{\mathcal{A}}$ are presented in Appendix Table 8.

5.3 Ablation Studies

The effectiveness of each component in AVocaDo, i.e., vocabulary adaptation and contrastive regularization, is shown in this section. As described in

Model	CHEMPROT	ACL-ARC	HYPERPARTISAN	AMAZON
AVocaDo	81.07	67.28	89.31	68.51
w/o $\mathcal{L}_{reg}$	78.45(-2.62)	64.00(-3.28)	87.84(-1.47)	61.23(-7.28)
$\text{BERT}_{\text{base}}$	79.35(-1.72)	56.82(-10.46)	84.51(-4.80)	55.50(-13.01)

Table 3: **Ablation study.** w/o $\mathcal{L}_{reg}$ denotes that the model is fine-tuned with the adapted vocabulary but not applying regularization loss. $\text{BERT}_{\text{base}}$ denotes that the model is fine-tuned without applying AVocaDo. The performance difference is represented inside the parentheses.

Dataset	Size of added vocabulary $n_{\mathcal{D}}$ 500	1000	2000	3000	AVocaDo
CHEMPROT	80.36	80.43	80.24	79.89	**81.06**
ACL-ARC	65.24	64.43	66.08	65.37	**67.28**
HYPERPARTISAN	84.70	85.03	80.49	84.85	**89.31**
AMAZON	68.57	68.31	**68.85**	67.89	68.51

Table 4: **Analysis on the size of the added vocabulary.** $n_{\mathcal{D}}$ is manually set (500, 1000, 2000, and 3000) or automatically determined (AVocaDo).

Table 3, vocabulary adaptation improves the performance in three domains (i.e., ACL-ARC, HYPERPARTISAN, and AMAZON) even in the absence of the regularization term.

5.4 Size of Added Vocabulary

The size of the added vocabulary $n_{\mathcal{D}}$ is automatically determined by the fragment score of the adapted vocabulary $V_{\mathcal{A}}$, as described in Algorithm 1. In order to analyze how $n_{\mathcal{D}}$ affects the performance, we compare the performance of downstream tasks by manually setting the $n_{\mathcal{D}}$ as 500, 1000, 2000, and 3000 without using the fragment score, as shown in Table 4. Automatically determined $n_{\mathcal{D}}$ is 1600, 700, 2850 and 1300 for each dataset. Except for AMAZON dataset, we demonstrate that determining $n_{\mathcal{D}}$ by the fragment score shows the optimal performance.

6 Conclusion

In this paper, we demonstrate that a pretrained vocabulary should be updated towards a downstream domain when fine-tuning. We propose a fine-tuning strategy called **AVocaDo** that adapts the vocabulary to the downstream domain by expanding the vocabulary based on a tokenization statistic, and by regularizing the newly added words. Our approach shows consistent performance improvements in diverse domains on various pretrained language models. AVocaDo is applicable to a wide range of NLP tasks in diverse domains without any restrictions, such as massive computing resources or a large domain-specific corpus.

Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program(KAIST)), and Samsung Advanced Institute of Technology, Samsung Electronics Co., Ltd. We thank the anonymous reviewers for their helpful feedback and discussions. We also thank Seong-Su Bae for his insight and helpful opinion.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In *Proc. the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proc. the International Conference on Machine Learning (ICML)*, 2020.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. In *Proc. the International Conference on Learning Representations (ICLR)*, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proc. the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. Measuring the evolution of a scientific field through citation frames. *Transactions of the Association for Computational Linguistics(TACL).*

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. SemEval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, 2019.

Jens Kringelum, Sonny Kim Kjaerulff, Søren Brunak, Ole Lund, Tudor I. Oprea, and Olivier Taboureau. ChemProt-3.0: a global chemical biology diseases mapping. *Database*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 2020.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. Addressing the rare word problem in neural machine translation. 2015.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proc. of Special Interest Group on Information Retrieval (SIGIR)*, 2015.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. Inexpensive domain adaptation of pretrained language models: case studies on biomedical ner and covid-19 qa. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP): Findings*, 2020.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proc. the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Shoetsu Sato, Jin Sakuma, Naoki Yoshinaga, Masashi Toyoda, and Masaru Kitsuregawa. Vocabulary adaptation for domain adaptation in neural machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP): Findings*, 2020.

Timo Schick and Hinrich Schütze. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proc. the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015.

Merity Stephen, Xiong Caiming, Bradbury James, and Richard Socher. Pointer sentinel mixture models. In *Proc. the International Conference on Learning Representations (ICLR)*, 2016.

Wen Tai, HT Kung, Xin Luna Dong, Marcus Comiter, and Chang-Fu Kuo. exBERT: Extending pre-trained models with domain-specific vocabulary under constrained training resources. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP): Findings*, 2020.

Appendix

A Details on Fragment Score

Fragment score is a measure of the *fineness* of tokenization. We observed that the pretrained vocabulary set $V_{\mathcal{P}}$ tokenizes domain-specific words (i.e., words that are frequently appeared in a downstream corpus but not in a pretrained corpus) into larger number of subwords than the number of subwords that non-domain-specific words are tokenized into (Figure 3). These finely tokenized subwords are not semantically informative enough.

Inspired by the observations, we construct a new vocabulary $V_{\mathcal{A}}$ that *less finely* tokenizes the domain-specific words than $V_{\mathcal{P}}$, i.e., $V_{\mathcal{A}}$ such that $f_{\mathbf{C}}(V_{\mathcal{A}}) < f_{\mathbf{C}}(V_{\mathcal{A}})$. This is why we chose the fragment score of the newly constructed vocabulary set $V_{\mathcal{A}}$ as a metric for selecting a subset of domain-specific vocabulary $V_{\mathcal{D}}$.

B Different Aspects of the Vocabularies

Figure 3 shows the relative number of tokenized subwords from a single word in four domains where the publicly available vocabulary in BERT [Devlin et al., 2018] is denoted as $V_{\mathcal{P}}$ and domain adapted vocabularies are denoted as $V_{\mathcal{A}}$. WikiText [Stephen et al., 2016] represents the general domain that is similar to the corpus that is used for pretraining BERT, while others are chosen as the downtream domain. The red and orange bar indicate the average number of subwords tokenized with pretrain vocabulary and adapted vocabulary. We observe that AVocaDo mitigates the domain gap.

C Implementation Details

C.1 Downstream Datasets

Domain	Dataset	Task (# of Classes)	Train	Dev.	Test
BIOMED	CHEMPROT	relation (13)	4169	2427	3469
CS	ACL-ARC	citation intent (6)	1688	114	139
NEWS	HYPERPARTISAN	partisanship (2)	515	65	65
REVIEWS	AMAZON	helpfulness (2)	115251	5000	25000

Table 5: Datasets used in experiments. Sources: CHEMPROT [Kringelum et al.], ACL-ARC [Jurgens et al.], HYPERPARTISAN [Kiesel et al., 2019], and AMAZON [McAuley et al., 2015].

We used four datasets in various domains for classification. As shown in Table 5, the size of the training data varies from 500 to about 110,000. The number of classes for each dataset varies from 2 to 13.

C.2 Experimental Settings

In all experiments, we trained the networks on a single 3090 RTX GPU with 24GB of memory. We implemented all models with PyTorch using Transformers library from Huggingface. All baselines are reproduced as described in previous works [Gururangan et al., 2020, Tai et al., 2020, Lee et al., 2020, Beltagy et al., 2019]. In our experiment, the performance in HYPERPARTISAN dataset tends to have high variance depending on random seeds since the size of the dataset is extremely small. To produce reliable results on this dataset, we discard and resample seeds.

The embeddings of newly added words in AVocaDo are initialized as a mean value of BERT embeddings of subword components. For instance, if the word "bluetooth" is tokenized into ["blue","##tooth"] with $V_{\mathcal{P}}$ and "bluetooth" with $V_{\mathcal{A}}$, we initialize the embedding of "bluetooth" with the average value of the two subword embeddings.

C.3 Hyperparameters

Hyperparameter	Value
lower bound of fragment score γ	3
number of added vocabulary (initial) α	500
number of added vocabulary β	50
batch size B	16
learning rate	1e-5, 2e-5, 5e-5
number of epochs	10
temperature τ	from 1.5 to 3.5
domain vocabulary size $N_{\mathcal{D}}$	10,000

Table 6: Hyperparameters used in experiments. We conduct grid search for finding the best hyperparameter settings.

As shown in Table 6, we followed the hyperparameter setting in the previous work [Lee et al., 2020, Beltagy et al., 2019, Gururangan et al., 2020]. To search the value for learning and temperature τ, we use grid search.

D Qualitative Results

For each downstream dataset, we randomly sampled ten words that are differently tokenized by pretrained vocabulary $V_{\mathcal{P}}$ and by adapted vocabulary $V_{\mathcal{A}}$. As shown in Table 8, subwords tokenized by $V_{\mathcal{A}}$ are more informative in the target domain

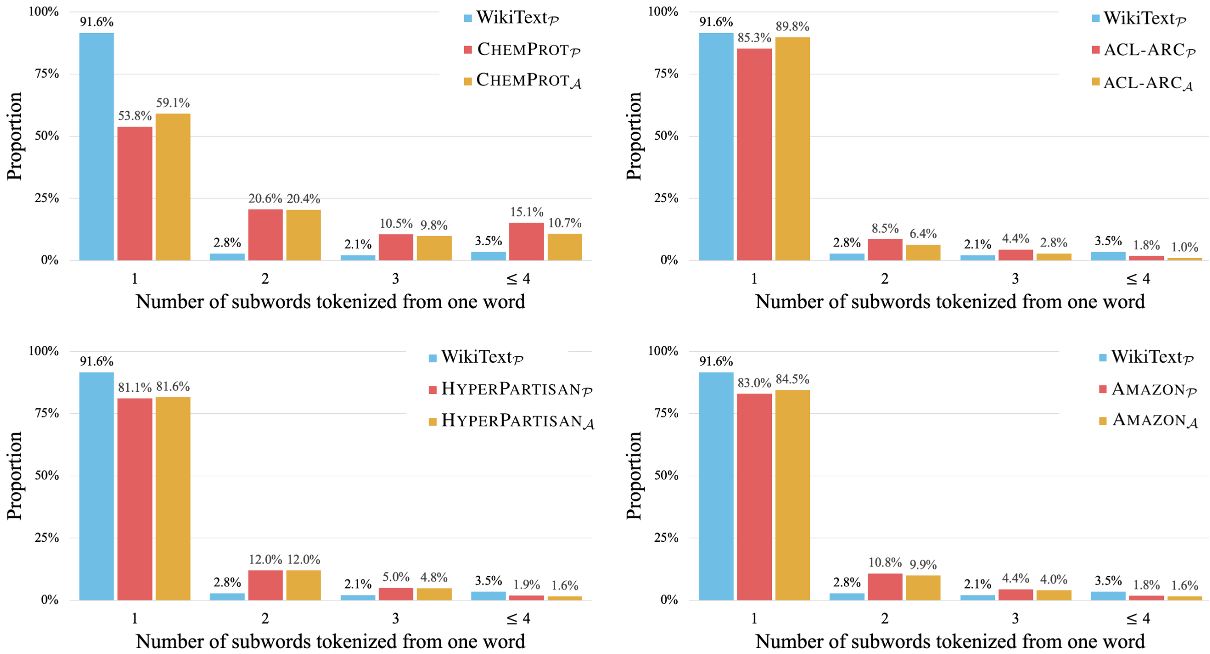


Figure 3: The analysis of the pretrained and adapted vocabularies on WikiText and downstream domains. $\mathcal{P}$ and $\mathcal{A}$ denote the pretrained vocabulary and the adapted vocabulary respectively. **AVocaDo** mitigates the domain gap in terms of the average number of subwords tokenized from a single word.

because they preserve the semantic of a domain-specific word.

E Comparison with Previous Works

Model	Adaptive Pretraining	Domain-specific Corpus
AVocaDo	×	downstream corpus only
SciBERT	✓	3.17 billion words
BioBERT	✓	18.0 billion words
exBERT	✓	0.9 billion words
Gururangan et al. [2020]	✓	7.55 billion words

Table 7: **Comparison with previous works.** The adaptive pretraining phase and the size of biomedical domain corpus used for domain adaptation in previous works. No additional training resource is needed in AVocaDo.

AVocaDo does not require additional domain-specific corpus. As shown in Table 7, all other baseline models require an adaptive pretraining stage before fine-tuning using domain-specific corpus. In general, the corpus used for adaptive pretraining is relatively large compared to the size of downstream dataset. Therefore, most methodologies that require adaptive pretraining require large training resources.

F Other Baselines

We perform additional experiments with other baseline models. In this experiment, we set exBERT [Tai et al., 2020], which expands the pretrained vocabulary from original BERT$_{\text{base}}$ vocabulary, and SciBERT$_{\text{SCIVOCAB}}$ [Beltagy et al., 2019], which constructs the customized vocabulary based on science and biomedical large corpora as baselines. Table 9 shows the overall performance on BIOMED and CS domains. We outperform exBERT in BIOMED domain. In comparison with SciBERT$_{\text{SCIVOCAB}}$, AVocaDo shows the competitive performance.

G Other Pretrained Language Models

To demonstrate the performance of AVocaDo on the other pretrained language models, we additionally conducted experiments on RoBERTa [Liu et al., 2019] and ELECTRA [Clark et al., 2020]. Table 10 shows the overall performance on four downstream domains. RoBERTa and ELECTRA with AVocaDo shows the improvements on the various domains except for NEWS and BIOMED domain respectively.

Domain	Word	Pretrained Vocab $V_{\mathcal{P}}$	Adapted Vocab $V_{\mathcal{A}}$
BIOMED	epidermal	ep, ##ider, ##mal	epidermal
	cetuximab	ce, ##tu, ##xi, ##ma, ##b	ce, ##tu, ##xi, ##ma, ##b
	lumiracoxib	lu, ##mir, ##aco, ##xi, ##b	lum, ##irac, ##oxib
	peroxidation	per, ##ox, ##ida, ##tion	perox, ##ida, ##tion
	reductase	red, ##uc, ##tase	reductase
	dihydrotestosterone	di, ##hy, ##dro, ##test, ##ost, ##eron	dihydro, ##test, ##osterone
	pparalpha	pp, ##ara, ##pl, ##ha	ppar, ##alpha
	sulfhydration	sul, ##f, ##hy, ##dra, ##tion	sulf, ##hydr, ##ation
	glucuronidation	g, ##lu, ##cu, ##ron, ##ida, ##tion	glucuron, ##ida, ##tion
	proliferating	pro, ##life, ##rating	prolifer, ##ating
CS	annotation	ann, ##ota, ##tions	annotation
	unsupervised	un, ##su, ##per, ##vis, ##ed	unsupervised
	entails	en, ##tails	entail, ##s
	sgd	sg, ##d	sgd
	parser	par, ##ser	parser
	nlp	nl, ##p	nlp
	suumarization	sum, ##mar, ##ization	summarization
	syntactic	syn, ##ta, ##ctic	syntactic
	coreference	core, ##ference	coreference
	ner	ne, ##r	ner
NEWS	manafort	mana, ##fort	manafort
	disrespectful	di, ##sr, ##es, ##pe, ##ct, ##ful	disrespect ##ful
	tweet	t, ##wee, ##t	tweet
	divisive	di, ##vis, ##ive	div, ##isi, ##ve
	recaptcha	rec, ##ap, ##tch, ##a	recaptcha
	brexit	br, ##ex, ##it	brexit
	irreplaceable	ir, ##re, ##pl, ##ace, ##able	ir, ##re, ##place, ##able
	supermacists	su, ##pre, ##mac, ##ists	supermacists
	politicize	pol, ##itic, ##ize	politic, ##ize
	gop	go, ##p	gop
REVIEWS	telestial	tel, ##est, ##ial	tele, ##sti, ##al
	rechargeminutes	rec, ##har, ##ge, ##min, ##ute, ##s	recharge, ##min, ##utes
	verizon	ve, ##riz, ##on	verizon
	thunderbolt	thunder, ##bolt	thunderbolt
	bluetooth	blue, ##tooth	bluetooth
	otterbox	otter, ##box	otterbox
	headset	heads, ##et	headset
	kickstand	kicks, ##tan, ##d	kickstand
	detachable	det, ##ach, ##able	detach, ##able
	htc	h, ##tc	htc

Table 8: **Randomly sampled words that are differently tokenized by $V_{\mathcal{P}}$ and $V_{\mathcal{A}}$.**

Domain	Dataset	BERT$_{\text{base}}$	BERT$_{\text{AVocaDo}}$	exBERT	SciBERT$_{\text{BASEVOCAB}}$†	SciBERT$_{\text{AVocaDo}}$
BIOMED	CHEMPROT	79.38	**81.07**	74.63	**83.64**	82.71
CS	ACL-ARC	56.82	**67.28**	-	70.98	**75.02**

Table 9: **Comparisons with other baselines.** The symbol † indicates the performance reported by Beltagy et al. [2019].

Domain	Dataset	RoBERTa$_{\text{base}}$†	RoBERTa$_{\text{AVocaDo}}$	ELECTRA$_{\text{base}}$	ELECTRA$_{\text{AVocaDo}}$
BIOMED	CHEMPROT	81.9	**82.8**$(+0.9)$	**74.3**	73.4(-0.9)
CS	ACL-ARC	63.0	**67.3**$(+4.3)$	57.1	**59.3**$(+2.2)$
NEWS	HYPERPARTISAN	**86.6**	84.5(-2.1)	70.6	**77.8**$(+7.2)$
REVIEWS	AMAZON	65.1	**70.8**$(+5.7)$	66.2	**69.9**$(+3.7)$

Table 10: **Experiments on other pretrained language models.** The pretrained language models (i.e., RoBERTa$_{\text{base}}$ and ELECTRA$_{\text{base}}$) are fine-tuned with or without AVocaDo. The performance improvement is represented inside the parentheses with $+$. The symbol † indicates the performance reported by Gururangan et al. [2020].