Findings of the 2020 Conf on Empirical Methods in Natural Language Processing (EMNLP 2020), November 2020

exBERT: Extending Pre-trained Models with Domain-specific Vocabulary Under Constrained Training Resources

Wen Tai
Center for Artificial Intelligence in Medicine,
Chang Gung Memorial Hospital
iroup9797@cgmh.org.tw

H. T. Kung
Harvard University
kung@harvard.edu

Xin Dong
Harvard University
xindong@g.harvard.edu

Marcus Comiter
Harvard University
marcuscomiter@g.harvard.edu

Chang-Fu Kuo
Center for Artificial Intelligence in Medicine,
Chang Gung Memorial Hospital
zandis@adm.cgmh.org.tw

Abstract

We introduce exBERT, a training method to **ex**tend **BERT** pre-trained models from a *general* domain to a new pre-trained model for a *specific* domain with a new additive vocabulary under constrained training resources (i.e., constrained computation and data). exBERT uses a **small** *extension module* to learn to adapt an augmenting embedding for the new domain in the context of the original BERT's embedding of a general vocabulary. The exBERT training method is novel in learning the new vocabulary and the extension module while keeping the weights of the original BERT model fixed, resulting in a substantial reduction in required training resources. We pre-train exBERT with biomedical articles from ClinicalKey and PubMed Central, and study its performance on biomedical downstream benchmark tasks using the MTL-Bioinformatics-2016 dataset. We demonstrate that exBERT consistently outperforms prior approaches when using limited corpus and pre-training computation resources.

1 Introduction

Pre-trained language representation models have led to breakthrough performance improvements in downstream natural language processing (NLP) tasks including named entity recognition (Sang and De Meulder, 2003), question answering (Rajpurkar et al., 2016), and sentence classification (Socher et al., 2013). However, pre-trained language models face two challenges as their applications expand: **1) Large Training Resources:** Training requires substantial computation and data, see, e.g., BERT-large (Devlin et al., 2018), RoBERTa (Liu et al., 2019). **2) Embedding of Domain-specific Vocabulary:** A specialized domain, such as the biomedical domain on which this work focuses, has its own vocabulary, and sentences in the domain may have words from both the original language model's vocabulary and new domain-specific vocabulary. Being able to operate on this mixture of vocabulary is essential in achieving high performance on downstream tasks in the new domain (Garneau et al., 2019).

These challenges are particularly pronounced in biomedical domains, where there are many domain-specific words. Prior approaches have addressed these issues by either constructing the pre-trained model from scratch with a new vocabulary (e.g., SciBERT (Beltagy et al., 2019)) or adapting the existing pre-trained model by using it as the initial model in learning vocabulary embeddings for the new domain (e.g., BioBERT (Lee et al., 2019)). However, constructing the model with a new vocabulary from scratch requires substantial computational resources and training data. Adapting the existing pre-trained model leads to sub-optimal performance on downstream tasks because the original vocabulary may not be proper for biomedical domains (Garneau et al., 2019; Hu et al., 2019).

We propose exBERT, a novel approach that addresses these challenges by explicitly incorporating the new domain's vocabulary, while being able to reuse the original pre-trained model's weights *as is* to reduce required computation and training data. Specifically, exBERT **ex**tends **BERT** by augmenting its embeddings for the original vocabulary with new embeddings for the domain-specific vocabulary via a learned small "extension" module. The output of the original and extension modules are combined via a trainable weighted sum operation. exBERT after pre-training achieves higher performance than the BioBERT adaption method under constrained training resources when evaluated on two biomedical downstream benchmark NLP tasks: name entity recognition (NER) (Doğan et al., 2014; Li et al., 2016) and relation extraction (RE) (Bhasuran and Natarajan, 2018).

The primary contribution of this paper is a pre-training method allowing low-cost embedding of domain-specific vocabulary in the context of an existing large pre-trained model such as BERT. The source code is available at https://github.com/cgmhaicenter/exBERT.

2 Related Work

Learning representations of natural languages is useful for a variety of NLP tasks (McCann et al., 2017; Liu et al., 2019). It has been demonstrated that larger model size and corpus size benefit performance (Radford et al., 2019). A widely used pre-training model, BERT (Devlin et al., 2018), is a transformer-based model (Vaswani et al., 2017) pre-trained with a masked language model and next sentence prediction task. The vocabulary used by BERT contains words and subwords extracted from a general language corpus (English Wikipedia and BooksCorpus) by WordPiece (Wu et al., 2016).

To get a biomedical domain-specific pre-training language model, BioBERT (Lee et al., 2019) continues training the original BERT model with a biomedical corpus without changing the BERT's architecture or the vocabulary, and achieves improved performance in several biomedical downstream tasks. However, the use of original BERT's general vocabulary often splits a domain-specific word into several sub-words, making the training much more challenging.

SciBERT (Beltagy et al., 2019) compares the vocabulary extracted from general and scientific articles, and finds 58% of the scientific vocabulary is not included in the original BERT's vocabulary. To address this problem, SciBERT uses a new vocabulary, including high-frequency words and subwords in scientific articles. Results show that the new vocabulary helps the performance of downstream tasks. However, the new vocabulary is not recognized by the pre-trained model; therefore, the model needs to be trained from scratch, requiring substantial computing resources and training data.

In a recent study, PubMedBERT (Gu et al., 2020) pre-trained the model from scratch with PubMed articles and a customized vocabulary (constructed from the PubMed articles). This study indicates that a proper vocabulary helps the performance of downstream tasks in specific domains. However, training the model from scratch is extremely expensive in terms of data and computation.

In multilingual language modeling, the out of vocabulary (OOV) problem harms the performance due to the limited vocabulary that cannot cover all the words in each language. The mixture mapping method of (Wang et al., 2019) represents each OOV word as a mixture of English subwords where the English subwords are already in the original vocabulary. However, our preliminary experiments have shown that directly initializing the embedding of the domain-specific words with the mixture of the subword embeddings dose not benefit the performance.

Transfer learning with extra adaptors (Houlsby et al., 2019) applied to the pre-trained model shows competitive performance compared with fine-tuning the pre-trained model. Training only a relatively small adaptor module is parameter efficient and the origin model is kept the same. Similar to this concept but not in a fine-tuning paradigm, we pre-train only the size-free extension module and the embedding layer of the extension vocabulary.

3 exBERT Approach

For exBERT, we augment the original BERT's embedding layer with an extension embedding layer and corresponding domain-specific extension vocabulary, and add an extension module to each transformer layer.

3.1 Extension Vocabulary and Embedding Layer

First, we derive an extension vocabulary from the target domain (biomedical for this paper) corpus via WordPiece (Wu et al., 2016), while keeping the original general vocabulary used by BERT unchanged. Any token in the extension vocabulary already present in the original general vocabulary is deleted to ensure the extension vocabulary is an absolute complement to the original vocabulary. We then add a corresponding embedding layer for the extension vocabulary, which is randomly initialized at the beginning and can be optimized during pre-training. The overall vocabulary, containing 30,522 (original) and 17,748 (extension) tokens, is used for tokenizing input text. This approach contrasts from SciBERT (Beltagy et al., 2019), which replaces the entire vocabulary and then pre-trains the model from scratch. We tried different extension vocabulary sizes and found that increasing the vocabulary size has a small impact on performance (e.g., increasing the extension vocabulary size by an additional 12K words only improve performance by 0.0041 F1 score). This is due to the fact that there is no clear drop off in vocabulary frequency of occurrence. Further, increasing vocabulary size increases time-to-convergence, so in order to bound the convergence time we choose a relatively small extension vocabulary size.

As illustrated in Figure 1(a), the output embedding of a given sentence consists of embedding vectors from both the original and extension embedding layer. Taking the sentence 'Thalamus is a part of brain' as an example, our overall vocabulary will tokenize it into eight tokens ('tha', '##lam', '##us', 'is', 'a', 'part', 'of', 'brain'), with the embedding vector of 'thalamus' coming from the extension embedding layer and all other tokens' embedding vectors from the original pre-trained embedding layer. Without the extension vocabulary, the original BERT might have tokenized 'thalamus' into three tokens, ('tha', '##lam', '##us'), compared to 'thalamus' tokenized as a single word under our method. Therefore by adding the extension vocabulary and corresponding embedding layer, exBERT enables more meaningful tokenization of input text.

However, there are still two issues: (1) Embedding vectors of the extension vocabulary are unknown to the pre-trained BERT model, (2) Distribution of token representation in the original vocabulary may experience a shift from the general domain to the target domain due to the use of different sentence styles, formality, intent, and so on. For example, the same word in the context of different domains may have different representations.

We address these issues by applying a weighted combination mechanism that allows the original BERT model and extension module to cooperate.

3.2 Extension module

exBERT augments each layer of the original BERT (referred to as the "off-the-shelf" module) by adding an extension module to its side as depicted in Figure 1(b).

To combine the output from the off-the-shelf module $T_{\text{ofs}}(\cdot)$ and the extension module $T_{\text{ext}}(\cdot)$, we use a weighted sum mechanism as below:

$$H^{l+1} = T_{\text{ofs}}(H^l) \cdot \sigma(w(H^l)) + T_{\text{ext}}(H^l) \cdot (1 - \sigma(w(H^l))$$

where H^l is the output of l-th layer and w is the weighting block, a fully-connected layer with size 768×1 that outputs the weight used to do a weighted summation of embedding vectors from the two modules. To make the output magnitude of the weighting block consistent, a sigmoid function $\sigma(\cdot)$ is used to constrain the output. The size of the extension module is flexible as long as its output shape matches that of the off-the-shelf module.

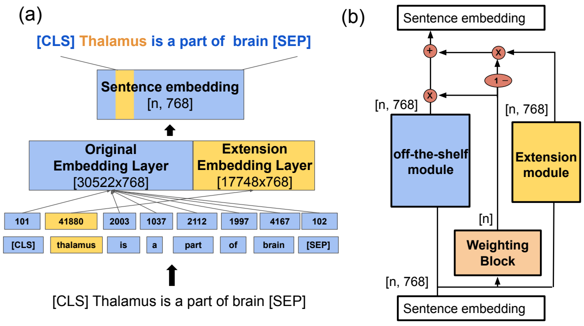

Figure 1: Sentence embedding and the exBERT architecture. (a) Derivation of the sentence embedding based on both the original and extension vocabulary. (b) Each input sentence consists of n 768-dimensional embedding vectors where n is 128 in our experiments. The output embedding is a component-wise weighted (computed by the weighting block) sum of outputs from the two modules.

4 Performance Evaluation of ExBERT

4.1 Experiment setup

exBERT Adaptive Pre-training All instances of BERT in this section refer to Bert-base-uncased (BERT). For exBERT, the 'extension module' uses the same transformer-based architecture as BERT (Devlin et al., 2018) with smaller sizes. The 'off-the-shelf' part of exBERT is a copy of the BERT model. During pre-training, this part remains completely fixed, and only the extension module and the weighting block are updated (except for a special experiment related to Figure 3(b)). Training uses the Adam optimizer (learning rate = $1e-04$, β_1 = 0.9, and β_2 = 0.999) on 4 V100 NVIDIA GPUs. The batch size and input length are set to 256 and 128, respectively.

We construct a biomedical corpus (which we call 17G-Bio in this paper) consisting of 17 GB articles from ClinicalKey (Clinicalkey) (2GB) and PubMed Central (PMC) (15GB). All or part of this corpus is used for the adaptive pre-training discussed in the next two sections.

Fine-tuning We compare different pre-trained models' performance after fine-tuning them on two downstream tasks: named entity recognition (NER) and relation extraction (RE)[1]. In other words, all

[1] Due to space limitation, results of RE are put in the Appendix. Basically, they show the same trends as NER.

scores in this paper are models' performance on the downstream tasks. Specifically, all pre-trained models are fine-tuned with the same setting: only the top three layers are fine-tuned with a learning rate of 10^{-5} and batch size of 20 for 3 epochs on the MTL-Bioinformatics-2016 dataset (MTL).

We first examine exBERT pre-trained under a limited corpus (sample randomly 5% data from the 17G-Bio) and computation resources (update model on the sampled corpus for three epochs), as a function of the extension module size. We test five different extension module sizes, 16.3%, 23.4%, 33.2%, 66.3% and 100%, of the off-the-shelf module size (with hidden sizes of 120, 180, 252, 504, 768 and feed-forward layer sizes of 512, 720, 1024, 2048, 3072, respectively). The performance of exBERT is compared against the original BERT and an our own trained version of BioBERT, rrBioBERT (*reduced-resource BioBERT*) pre-trained with the aforementioned limited resources but in the same way of BioBERT (Lee et al., 2019).

4.2 Impact of the Extension Module Size

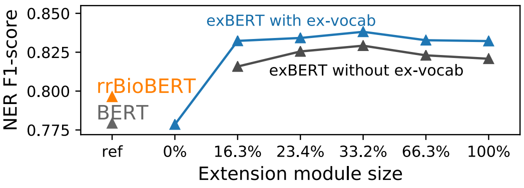

Figure 2: Performance (macro F1-score on the NER task) of exBERT model pre-trained on 5% of the 17G-Bio corpus as a function of extension module sizes, compared against BERT and rrBioBERT. The blue and black curves represent the exBERT model with and without vocabulary extension respectively.

exBERT	Extension module size				
	16.3%	23.4%	33.2%	66.3%	100%
w/ ex-vocab	0.8323	0.8342	0.8381	0.8327	0.8322
w/o ex-vocab	0.8158	0.8255	0.8292	0.8230	0.8208
Baselines	rrBioBERT	BERT	ex-vocab only		
	0.7963	0.7793	0.7785		

Table 1: Numerical data of Figure 2

As shown in Figure 2, exBERT outperforms the rrBioBERT model, even with a quite small extension module size (16.3%). This demonstrates that exBERT's pre-training using the extension module is effective and efficient, and the performance is stable when there is a sufficient number of parameters in the extension model. In the rest of this paper, we set the size of extension modules at 33.2%.

Further, under a separate experiment, we have studied a scenario where we include the extension vocabulary and the corresponding embedding layer but do not include the extension module (0% in Figure 2). We then update the whole model with the aforementioned limited resources. We find that this setting yields poor performance, showing that the extension module is crucial to make the original and extension vocabularies work together.

Furthermore, we have experimented with the paradigm that pre-trains only the extension module without the extension vocabulary (black curve in Figure 2). The result shows the exBERT's improvement in performance comes not only from the extension module, but also from the additional domain-specific vocabulary.

4.3 Impact of Training Time on Performance

We next examine exBERT's performance as a function of training time. We conduct adaptive pre-training of exBERT for 24 hours on the whole 17G-Bio corpus. For comparison, we also pre-train oiBioBERT (*our-implemented* BioBERT) with the same hardware and corpus but in the same manner as the way of BioBERT (Lee et al., 2019).

For every 4 hours of pre-training, we compare the performance of exBERT and oiBioBERT. Because the addition of the extension module incurs additional computation, given this 24-hour pre-training, the oiBioBERT model proceeds through a larger portion (34%) of the corpus than exBERT (24%). Nevertheless, as Figure 3(a) shows, for all amounts of pre-training time, exBERT outperforms oiBioBERT. This may be surprising given that exBERT takes less data due to increased computation (1.4x). We believe that the superior performance of exBERT reflects a significant benefit accrued by having the new domain's vocabulary explicitly represented in the exBERT model.

	Pre-training time					
	4 hrs	8 hrs	12 hrs	16 hrs	20 hrs	24 hrs
exBERT	.8283	.8366	.8382	.8390	.8396	.8413
oiBioBERT	.8109	.8085	.8106	.8166	.8207	.8104

	F1 score	Pre-training time	Model size
exBERT	0.8587	64 hrs	153M
	0.8611	128 hrs	153M
oiBioBERT	0.8117	64 hrs	110M
	0.8188	128 hrs	110M
SciBERT	0.8737	672 hrs	110M
BioBERT	0.8421	480 hrs	110M
BERT	0.7793	768 hrs	110M

Table 2: Numerical data of Figure 3

We also pre-train the models for a longer time on

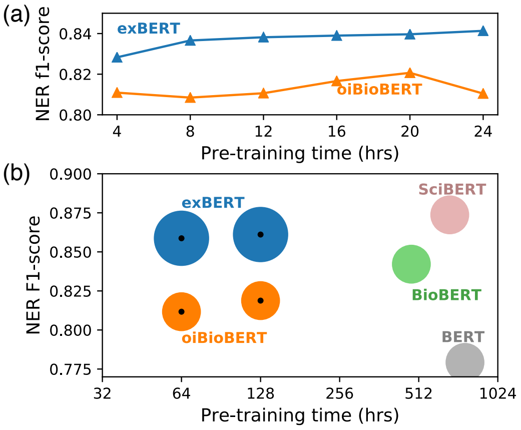

Figure 3: The NER performance for exBERT and oiBioBERT pre-trained on the whole 17G-Bio corpus. (a) Models pre-trained with varying amounts of training time. (b) Performance comparison against additional models, where for exBERT both the off-the-shelf and extension modules are updated. The size of a disc corresponds to the model size, and the axis is in a log scale. The discs with a black dot inside indicate models pre-trained by the authors of this paper.

the whole 17G-Bio corpus. After pre-training the exBERT model for 24 hours (only updating the extension embedding layer and modules), we continually pre-train the whole exBERT model, consisting of both the off-the-shelf and extension modules, recognizing that the larger corpus may be able to support the training of the whole model. We compare the results with three baselines, BERT (gray), BioBERT (green), and SciBERT (pink) (all of them are directly downloaded from their open-source implementations) as shown in Figure 3(b). For comparison, we convert the training time of these models to the time it may take with the same computing platform of this work (4 V100 GPUs), and assume that a TPU core has the same computing power as 2 V100 GPUs.

As shown in Figure 3(b), for a given training time, exBERT always outperforms oiBioBERT. We also find the exBERT pre-trained with lower resources (4 V100 GPUs, 64 hrs) outperforms the original BioBERT (8 V100 GPUs, 240 hrs, or 4 V100 GPUs 480 hrs in Figure 3(b)).

We additionally compare the size of the different models, represented as the disc size in Figure 3(b). The size of exBERT model (138 million parameters, with the extension modules' size being 33.2% of the off-the-shelf modules' size) is generally larger than the original BERT (110 million parameters) due to the added extension embedding layer and modules. Although we provide model sizing information here, this paper focuses on maximizing performance under constrained computation and data rather than minimizing model size. As future work, the model size could be reduced by, e.g., model compression methods (Gordon et al., 2020) or using a smaller distilled version of BERT (Sanh et al., 2019) as the off-the-shelf module.

5 Conclusion

exBERT is proposed to maximize the use of an elaborately pre-trained model for a general domain by empowering the model's continual learning ability to adapt and shift the learned representation for a new domain with a low training cost. exBERT adds a new domain-specific vocabulary and the corresponding embedding layer, as well as a small extension module to the original unmodified model. The exBERT approach greatly improves the efficiency of adapting a pre-training model for a new target domain.

With exBERT we can reuse pre-trained language models for new domains under limited training resources. The approach could be particularly attractive to ad-hoc and special-purpose domains with unique vocabularies, such as some fields in law, medicine, and engineering, which have very limited training data for model pre-training and demand fast turnaround training.

Acknowledgements

This work is supported in part by the Air Force Research Laboratory under agreement number FA8750-18-1-0112 and a gift from MediaTek USA.

References

Iz Beltagy, Arman Cohan, and Kyle Lo. 2019. Scibert: Pretrained contextualized embeddings for scientific text. *arXiv preprint arXiv:1903.10676*.

BERT. `https://github.com/google-research/bert`.

Balu Bhasuran and Jeyakumar Natarajan. 2018. Automatic extraction of gene-disease associations from literature using joint ensemble learning. *PloS one*, 13(7):e0200699.

Clinicalkey. `https://www.clinicalkey.com`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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.

Nicolas Garneau, Jean-Samuel Leboeuf, and Luc Lamontagne. 2019. Predicting and interpreting embeddings for out of vocabulary words in downstream tasks. *arXiv preprint arXiv:1903.00724*.

Mitchell A Gordon, Kevin Duh, and Nicholas Andrews. 2020. Compressing bert: Studying the effects of weight pruning on transfer learning. *arXiv preprint arXiv:2002.08307*.

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

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. *arXiv preprint arXiv:1902.00751*.

Ziniu Hu, Ting Chen, Kai-Wei Chang, and Yizhou Sun. 2019. Few-shot representation learning for out-of-vocabulary words. *arXiv preprint arXiv:1907.00505*.

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

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*.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.

MTL. https://github.com/cambridgeltl/MTL-Bioinformatics-2016.

PMC. https://www.ncbi.nlm.nih.gov/pmc/tools/ftp/.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

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*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Hai Wang, Dian Yu, Kai Sun, Janshu Chen, and Dong Yu. 2019. Improving pre-trained multilingual models with vocabulary expansion. *arXiv preprint arXiv:1909.12440*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

A Appendix

We provide our results on the RE task mentioned in Section 4 in the same formats as Figure 2 and 3. We find the performance of the models on the RE task follows a similar trend to the NER task. In particular, exBERT outperforms the rrBioBERT and oiBioBERT under the same pre-training conditions. Note that following previous work (Beltagy et al., 2019; Lee et al., 2019), we use the micro F1 score here.

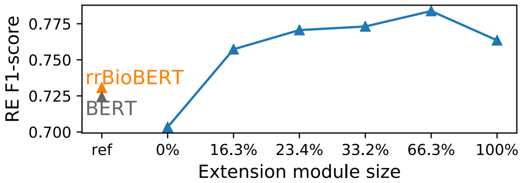

Figure 4: Performance (micro F1-score on the RE task) of exBERT model pre-trained on 5% of the 17G-Bio corpus as a function of extension module sizes, compared against BERT and rrBioBERT.

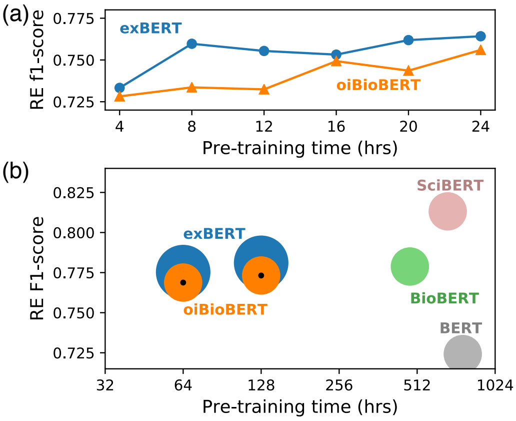

Figure 5: The RE performance (micro F1 score) for exBERT and oiBioBERT pre-trained on the whole 17G-Bio corpus. (a) Models pre-trained with varying amounts of training time. (b) Performance comparison against additional models, where for exBERT both the off-the-shelf and extension modules are updated. The size of a disc corresponds to the model size, and the axis is in a log scale. The discs with a black dot inside indicate models pre-trained by the authors of this paper.