Do not Mask Randomly: Effective Domain-adaptive Pre-training by Masking In-domain Keywords

Shahriar Golchin[†], **Mihai Surdeanu**[†], **Nazgol Tavabi**[◇], **Ata Kiapour**[◇]
[†]University of Arizona, Tucson, AZ, USA
[◇]Harvard Medical School, Boston, MA, USA
golchin@arizona.edu

Abstract

We propose a novel task-agnostic in-domain pre-training method that sits between generic pre-training and fine-tuning. Our approach selectively masks *in-domain keywords*, i.e., words that provide a compact representation of the target domain. We identify such keywords using KeyBERT (Grootendorst, 2020). We evaluate our approach using six different settings: three datasets combined with two distinct pre-trained language models (PLMs). Our results reveal that the fine-tuned PLMs adapted using our in-domain pre-training strategy outperform PLMs that used in-domain pre-training with random masking as well as those that followed the common pre-train-then-fine-tune paradigm. Further, the overhead of identifying in-domain keywords is reasonable, e.g., 7–15% of the pre-training time (for two epochs) for BERT Large (Devlin et al., 2019).[1]

1 Introduction

Employing large pre-trained language models (PLMs) is currently a common practice for most natural language processing (NLP) tasks (Tunstall et al., 2022). A two-stage pre-train-then-fine-tune framework is usually used to adapt/fine-tune PLMs to downstream tasks (Devlin et al., 2019). However, motivated by ULMFiT (Howard and Ruder, 2018) and ELMo (Peters et al., 2018), Gururangan et al. (2020) showed that incorporating in-domain pre-training (also known as domain-adaptive pre-training) between generic pre-training and fine-tuning stages can lead to further performance improvements in downstream tasks because it "pulls" the PLM towards the target domain. At this intermediate stage, the domain adaptation for PLMs is typically handled by continuing pre-training in the same way, i.e., using randomly-masked tokens on unstructured in-domain data (Devlin et al., 2019). Here, we argue that this intermediate pre-training should be performed differently, i.e., masking should focus on *words that are representative of target domain* to streamline the adaptation process.

We propose a novel task-independent in-domain pre-training approach for adapting PLMs that increases domain fit by focusing on *keywords* in the target domain, where keywords are defined as "a sequence of one or more words that offers a compact representation of a document's content" (Rose et al., 2010). By applying token masking only to in-domain keywords, the meaningful information in the target domain is more directly captured by the PLM. This is in contrast to the classic pre-training strategy that randomly masks tokens (Devlin et al., 2019), which may overlook domain-meaningful information, or the in-domain pre-training methods that selectively mask tokens deemed important given the downstream task (Gu et al., 2020, inter alia), which require incorporating information from the downstream task into the pre-training stage. We empirically show that our method offers a better transmission of high-quality information from the target domain into PLMs, yielding better generalizability for the downstream tasks.

The key contributions of this paper are:

(1) We propose the first task-agnostic selective masking technique for domain adaptation of PLMs that relies solely on in-domain keywords. In particular, we first extract contextually-relevant keywords from each available document in the target domain using KeyBERT (Grootendorst, 2020) and keep the most frequently occurring keywords to be masked during the adaptation phase.

(2) We evaluate our proposed strategy by measuring the performance of fine-tuned PLMs in six different settings. We leverage three different datasets for text classification from multiple domains: IMDB movie reviews (Maas et al., 2011),

[1]The code for all of our experiments is available at https://github.com/shahriargolchin/do-not-mask-randomly.

Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023), pages 13–21

Amazon pet product reviews from Kaggle,[2] and PUBHEALTH (Kotonya and Toni, 2020). Our experiments show that the classifiers trained on top of two PLMs—in our case, Bidirectional Encoder Representations from Transformers (BERT) Base and Large (Vaswani et al., 2017; Devlin et al., 2019)—that are adapted based on our suggested approach outperform all baselines, including the fine-tuned BERT with no in-domain adaptation, and fine-tuned BERT adapted by random masking. Further, the overhead of identifying in-domain keywords is reasonable, e.g., 7–15% of the pre-training time (for two epochs of data) for BERT Large.

2 Related Work

Bidirectional Encoder Representations from Transformers (BERT) brought pre-training to transformer networks (Vaswani et al., 2017) through masked language modeling (MLM) (Devlin et al., 2019). They showed that a simple two-step paradigm of generic pre-training followed by fine-tuning to the target domain can significantly improve performance on a variety of tasks.

However, after showing that infusing an intermediate pre-training stage (commonly known as in-domain pre-training) can help pre-trained Long Short-Term Memory models learn domain-specific patterns better (Howard and Ruder, 2018; Peters et al., 2018), Gururangan et al. (2020) found that the same advantage applies to PLMs as well. Since then, several efforts proposed different domain-adaptive pre-training strategies.

Unsurprisingly, one of the most extensively utilized in-domain pre-training methodologies has been to employ classic random masking to adapt PLMs into several domains (Lee et al., 2020; Beltagy et al., 2019; Alsentzer et al., 2019; Tavabi et al., 2022b,a; Araci, 2019). Following this, Zheng et al. (2020) introduced the fully-explored MLM in which random masking is applied to specific non-overlapping segments of the input sequence. The limitation of random masking that we aim to address is that it may put unnecessary focus on tokens that are not representative of the target domain.

In contrast, task-specific selective masking methods mask tokens that are important to the downstream task. For each task, "importance" is defined differently: Gu et al. (2020) let an additional neural model learns important tokens given the task at hand; Ziyadi et al. (2020) defined importance by masking entities for the named entity recognition task, and Feng et al. (2018) found important tokens by input reduction—maintaining model's confidence in the original prediction by reducing input—and they were left with a few (potentially nonsensical) tokens that were treated as important to model. Similarly, Li et al. (2020) designed a task-dependent objective for dialogue adaptation, and Ke et al. (2019) proposed label-aware MLM for a sentiment analysis task. In the same vein, token selection in certain domains, e.g., biomedical and clinical domains, was performed based on the entities relevant to the domain (Lin et al., 2021; Zhang et al., 2020b; Pergola et al., 2021).

Note that other MLM-based pre-training strategies focused on training a language model from scratch (Zhang et al., 2020a; Joshi et al., 2020; Sun et al., 2019, inter alia). However, since our work focuses on in-domain pre-training, we skip this part for brevity.

In this study, we propose an information-based domain-adaptive pre-training that, without being aware of the downstream task, selectively masks words that are information-dense with respect to the target domain. As a result, PLMs adapted using our mechanism outperform baselines adapted with random masking or fine-tuned directly. In the following sections, we refer to our approach as "keyword masking pre-training."

3 Approach

3.1 Extracting In-domain Keywords

In order to extract keywords relevant to the domain of interest, we use KeyBERT (Grootendorst, 2020). In a nutshell, KeyBERT uses BERT's (Devlin et al., 2019) contextualized embeddings to find the n-grams–in our scenario, unigrams–that concisely describe a given document. In particular, word embeddings with the highest cosine similarity to the overall document-level representation are identified as keywords that best represent the entire document. We configure KeyBERT to extract up to 10 keywords from each input document. Note that we do not pre-train or fine-tune BERT as the underlying model for KeyBERT.

3.2 Removing Noisy Keywords

After extracting domain-specific keywords, we compute the frequency of each specific word that has been recognized as a keyword in all in-domain

[2] https://www.kaggle.com/datasets/kashnitsky/exploring-transfer-learning-for-nlp

documents. Subsequently, we sort them in descending order of their frequency and keep only the most frequent ones. This simple strategy allows us to remove keywords that are likely to be noisy or irrelevant to the target domain.

Figure 1 summarizes the noisy keyword removal process for PUBHEALTH dataset (see Appendix B for other domains). Note that the actual figure has a very long tail on the right, indicating that the actual in-domain keywords (or parts where information is condensed in the target domain) are frequently repeated. The graph displays the frequency of terms along with the number of times they are identified as keywords. In the PUBHEALTH dataset, for example, more than 10,000 words were detected as keywords only once. Thus, we select the cut-off point where the curve is intended to leap up, signaling that keywords with repetition below the threshold were excluded from the list of domain-relevant keywords.[3] Namely, in the PUBHEALTH dataset, all words detected fewer than eight times as a keyword were removed from the list of in-domain keywords, and consequently, for performing keyword masking pre-training. The provided examples on the graph in Figure 1 indicate a qualitative indication that KeyBERT, coupled with our frequency-based heuristic, selects meaningful domain-specific keywords. For example, our approach identifies relevant keywords (e.g., *health*, *coronavirus*), while skipping other less relevant ones (e.g., *gym*, *gift*).

3.3 Keyword Masking Pre-training

We pair the list of retrieved candidate keywords with all target domain documents to perform keyword masking pre-training. If any of the keywords from the list appear in the input documents, the tokens corresponding to those keywords get masked given the masking probability. In our pre-training strategy, we use a constant learning rate scheduler with a high masking probability rather than a linear one to force the majority of tokens associated with keywords to be masked while continuously learning from surrounding tokens. As our approach inherits from MLM (Devlin et al., 2019), the tokens related to keywords are masked 80% of the time, replaced 10% of the time with other tokens, and left unchanged 10% of the time. Note that during pre-training masking only applies to the tokens that match the candidate keywords. Therefore, there is no pre-training for unmasked tokens.

3.4 Fine-tuning and Baselines

We compare the performance of all fine-tuned PLMs adapted using our technique with two other baselines: fine-tuned PLMs adapted using random masking, and fine-tuned PLMs with no in-domain adaptation. For all these settings we employ both BERT Base and BERT Large (Devlin et al., 2019).

4 Experimental Setup

Data: In our experiments, we chose tasks and datasets with sufficient amounts of unlabeled data for the domain adaptation stage in order to observe the effects of keyword selection.[4] In particular, we evaluate our method on three text classification datasets: PUBHEALTH (Kotonya and Toni, 2020), which contains public health claims associated with veracity labels, IMDB movie reviews dataset (Maas et al., 2011), and Amazon pet product reviews dataset (from a Kaggle competition).[5]

Based on the thresholds we studied for filtering out the noisy keywords (see Section 3.2), we gathered 2,116, 7,274, and 6,881 domain-specific keywords from the PUBHEALTH dataset, IMDB dataset, and Amazon dataset, respectively.

Settings: We use KeyBERT (Grootendorst, 2020) to extract up to 10 unigram keywords per input document utilizing contextualized word embeddings of BERT Base (Devlin et al., 2019), stratified by the Maximal Marginal Relevance (MMR) (Carbonell and Goldstein, 1998) with a threshold of 0.8.

To perform keyword masking pre-training, we set the masking probability to 0.75 with a constant learning scheduler. The other hyperparameters are left at their default values from the Hugging Face data collator for whole word masking (Wolf et al., 2020). For random masking pre-training, we set the masking probability to 0.15, which is a standard value for continual MLM pre-training, and left the remaining hyperparameters at the values provided by the Hugging Face data collator for language modeling (Wolf et al., 2020). Note that the default learning rate scheduler is linear. Further, in all settings, pre-training is limited to two epochs, and

[3]The threshold is adjusted via three points: an empirically chosen point from the graph, a point before, and a point after it. Following keyword masking based on each of these three thresholds, we choose the one that resulted in the highest F1 score on the validation split as the final threshold.

[4]For example, we did not use the GLUE dataset (Wang et al., 2018) because the included texts are short.

[5]https://www.kaggle.com/datasets/kashnitsky/exploring-transfer-learning-for-nlp

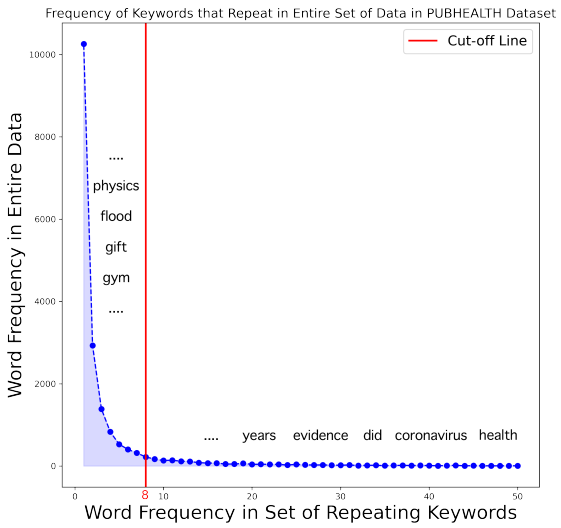

Figure 1: The graph shows the frequency of the last 50 most frequent keywords in the PUBHEALTH domain and the cut-off line for removing noisy keywords.

PUBHEALTH Dataset		
Adaptation Method	**Accuracy (%)**	**F1 Score (%)**
No Adaptation	64.80	63.23
Random Masking	65.77	64.94
Our Keyword Masking	***66.09**	***65.40**
IMDB Movie Reviews Dataset		
Adaptation Method	**Accuracy (%)**	**F1 Score (%)**
No Adaptation	94.44	94.43
Random Masking	94.96	94.95
Our Keyword Masking	***95.36**	***95.35**
Amazon Pet Product Reviews Dataset		
Adaptation Method	**Accuracy (%)**	**F1 Score (%)**
No Adaptation	85.89	85.73
Random Masking	86.33	86.31
Our Keyword Masking	***87.14**	***86.98**

Table 1: A comparison between the performance of fine-tuning adapted PLMs using our keyword masking and other baselines when *BERT Base* is used as the PLM. The best results are shown **bold** and the obtained statistically significant results compared to random masking are indicated by an asterisk (*) (see Appendix E).

PUBHEALTH Dataset		
Adaptation Method	**Accuracy (%)**	**F1 Score (%)**
No Adaptation	66.42	**65.08**
Random Masking	63.90	64.74
Our Keyword Masking	***66.66**	64.74
IMDB Movie Reviews Dataset		
Adaptation Method	**Accuracy (%)**	**F1 Score (%)**
No Adaptation	95.38	95.37
Random Masking	95.50	95.49
Our Keyword Masking	**95.52**	**95.51**
Amazon Pet Product Reviews Dataset		
Adaptation Method	**Accuracy (%)**	**F1 Score (%)**
No Adaptation	85.69	85.71
Random Masking	86.84	86.72
Our Keyword Masking	***87.58**	***87.51**

Table 2: A comparison between the performance of fine-tuning adapted PLMs using our keyword masking and other baselines when *BERT Large* is used as the PLM. The best results are shown **bold** and the obtained statistically significant results compared to random masking are indicated by an asterisk (*) (see Appendix E).

the batch size of 16 is adopted during both the adaptation and fine-tuning stages.

With the learning rate set to 2e-5 and the weight decay set to 0.01 (Devlin et al., 2019), we fine-tune the whole network for all of our adapted models and baselines for up to four epochs in all datasets, while keeping the other hyperparameters at the default value of Hugging Face (Wolf et al., 2020). The models that obtained the highest F1 score in the validation partition are then chosen and evaluated on the test split of the datasets.

5 Results and Discussion

Table 1 and 2 report the performance of fine-tuned models that used multiple domain-adaptive pre-training methods for each of our settings: three different datasets and two distinct PLMs. Table 1 contains the results for *BERT Base* as underlying PLM; Table 2 uses *BERT Large*.

In particular, each table contrasts the performance of two fine-tuned baselines—one without adaptation/in-domain pre-training and one with random masking in-domain pre-training—to a fine-

tuned model adapted using our keyword masking.

Both tables show that our approach outperforms all other baselines in all six settings. The improvements are statistically significant in four out of six settings (Appendix E). This highlights the importance of selecting information-carrying keywords for masking during the in-domain pre-training.

The results reveal that our suggested in-domain pre-training technique outperforms alternative settings with or without standard in-domain pre-training on target domain unlabeled data. Although the benefits of continual pre-training vary depending on the domain and the task at hand (Gururangan et al., 2020), our adaptation strategy always has a greater impact on PLMs in capturing domain-specific patterns compared to typical random masking when in-domain adaptation has a positive impact on downstream tasks. This indicates that our pre-training method indeed exposes the PLMs to relevant in-domain representations.

Given the superior outcomes seen in our six different experiments, we can argue that our selective masking strategy, which is task-agnostic as random masking yet more effective, could potentially widely replace random masking in the intermediate pre-training stage for a variety of NLP tasks. Other than performance, our method is simple and has no "pathological behavior" (Feng et al., 2018) (see Appendix C). Additionally, our method takes 2 to 10 minutes of computational overhead to extract keywords. This accounts for 7% to 39% of pre-training time of only two epochs (Appendix A).

6 Conclusion

We proposed the first task-agnostic selective masking pre-training approach, dubbed "keyword masking," to adapt PLMs to the target domains. For keyword masking, we first extract in-domain keywords from the target domain using KeyBERT (Grootendorst, 2020), and after excluding the noisy ones, we only mask the selected keywords during adaptation.

We evaluated our methodology using six different settings. The results revealed that when in-domain pre-training is conducted using our approach, all fine-tuned PLMs outperform those with no adaptation or adapted using random masking. Further, we observed that our pre-training approach was superior for difficult tasks, i.e., datasets with many labels and more complexity. Lastly, keyword masking pre-training can be widely substituted with random masking during shift domain in NLP tasks since it is task-independent, as simple to use as random masking, and more effective.

7 Limitations

Although all pre-training approaches require a sufficient amount of data, given how we defined keywords, longer sequences suit our approach better than short ones for studying the effects of keyword selection. Further, as shown in this study, our findings strongly imply that the strategy we suggested for adapting PLMs can effectively enhance their performance on text classification as the downstream task. To determine whether these findings can translate to other NLP applications, however, further experiments are required.

8 Ethics Statement

Although keyword extraction may amplify bias depending on the input documents and the way it extracts keywords, KeyBERT (Grootendorst, 2020) has not been reported to exhibit this behavior. Further work may be necessary to thoroughly explore the potential of introducing undesired bias.

References

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*.

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 335–336, New York, NY, USA. Association for Computing Machinery.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37 – 46.

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.

B. Efron. 1979. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1):1 – 26.

Bradley Efron. 2003. Second Thoughts on the Bootstrap. *Statistical Science*, 18(2):135 – 140.

Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Number 57 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, Boca Raton, Florida, USA.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. *arXiv preprint arXiv:1804.07781*.

Francis Galton. 1892. *Finger prints*. 57490-57492. Macmillan and Company.

Maarten Grootendorst. 2020. Keybert: Minimal keyword extraction with bert.

Yuxian Gu, Zhengyan Zhang, Xiaozhi Wang, Zhiyuan Liu, and Maosong Sun. 2020. Train no evil: Selective masking for task-guided pre-training. In *Conference on Empirical Methods in Natural Language Processing*.

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. *ArXiv*, abs/2004.10964.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Pei Ke, Haozhe Ji, Siyang Liu, Xiaoyan Zhu, and Minlie Huang. 2019. Sentilare: Sentiment-aware language representation learning with linguistic knowledge. *arXiv preprint arXiv:1911.02493*.

Neema Kotonya and Francesca Toni. 2020. Explainable automated fact-checking: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5430–5443, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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.

Junlong Li, Zhuosheng Zhang, Hai Zhao, Xi Zhou, and Xiang Zhou. 2020. Task-specific objectives of pre-trained language models for dialogue adaptation. *ArXiv*, abs/2009.04984.

Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana K. Savova. 2021. Entitybert: Entity-centric masking strategy for model pretraining for the clinical domain. In *Workshop on Biomedical Natural Language Processing*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Mary L. McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia Medica*, 22:276 – 282.

Gabriele Pergola, Elena Kochkina, Lin Gui, Maria Liakata, and Yulan He. 2021. Boosting low-resource biomedical qa via entity-aware masking strategies. *arXiv preprint arXiv:2102.08366*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1(1-20):10–1002.

David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. 2015. Hidden technical debt in machine learning systems. *Advances in neural information processing systems*, 28.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Nigel C. Smeeton. 1985. Early history of the kappa statistic. *Biometrics*, 41(3):795–795.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Nazgol Tavabi, James Pruneski, Shahriar Golchin, Mallika Singh, Ryan Sanborn, Benton Heyworth, Amir Kimia, and Ata Kiapour. 2022a. Building large-scale registries from unstructured clinical notes using a low-resource natural language processing pipeline. *medRxiv*.

Nazgol Tavabi, Marium Raza, Mallika Singh, Shahriar Golchin, Harsev Singh, Grant D Hogue, and Ata M Kiapour. 2022b. A natural language processing pipeline to study disparities in cannabis use and documentation among children and young adults a survey of 21 years of electronic health records. *medRxiv*.

L. Tunstall, L. von Werra, and T. Wolf. 2022. *Natural Language Processing with Transformers*. O'Reilly Media.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Ningyu Zhang, Qianghuai Jia, Kangping Yin, Liang Dong, Feng Gao, and Nengwei Hua. 2020b. Conceptualized representation learning for chinese biomedical text mining. *arXiv preprint arXiv:2008.10813*.

Mingzhi Zheng, Dinghan Shen, Yelong Shen, Weizhu Chen, and Lin Xiao. 2020. Improving self-supervised pre-training via a fully-explored masked language model. *arXiv preprint arXiv:2010.06040*.

Morteza Ziyadi, Yuting Sun, Abhishek Goswami, Jade Huang, and Weizhu Chen. 2020. Example-based named entity recognition. *arXiv preprint arXiv:2008.10570*.

A Cost of In-domain Keyword Extraction

For all of our settings, we leverage a single NVIDIA RTX A6000 GPU. Depending on the size of the datasets and the length of input documents, running KeyBERT (Grootendorst, 2020) for extracting in-domain keywords adds additional computation that can be different between 2 and 10 minutes in our settings. The overhead time for keyword extraction and the in-domain pre-training time for each of the settings are compared in Table 3. As can be noticed, the time ratio for keyword extraction to pre-training time ranges from 7% to 15% in settings using BERT Large, and 19% to 39% for settings with BERT Base, which is reasonable. Note that when pre-training is performed for more epochs, this ratio noticeably decreases. The reported ratios are based on only two epochs of in-domain pre-training in our settings.

B Removing Noisy Keywords (Graphs)

Similar to Figure 1, which illustrates the removal of noisy keywords for the PUBHEALTH dataset, Figure 2 displays this procedure for the IMDB dataset and the Amazon dataset.

C Pathological-free Behavior

It is possible that tokens to be selected for masking are not associated with the domain according to human experts, but they nevertheless yield better downstream classifiers. For instance, Feng et al. (2018) demonstrated that even when the model is left with a small number of tokens after input reduction, it can still be confident in its predictions even though the left tokens are meaningless. A similar phenomenon was reported for prompting (Shin et al., 2020). To show our masking method's non-pathological behavior, we asked two human annotators to annotate the domain relevance of 50 randomly-chosen words that were selected for masking by the respective method.[6] The annotations were performed using a three-point Likert scale: irrelevant, moderately relevant, and relevant.

Table 4 reports the results of this experiment as well as the Kappa inter-annotator agreement score (Galton, 1892; Cohen, 1960; Smeeton, 1985; McHugh, 2012). We draw two observations from this table. First, the agreement between the two

[6]The annotators were two of the authors. The annotations were independent, i.e., no annotator saw the decisions of the other. The names of the methods used to generate the 50 words to annotate were hidden during annotation.

Dataset Name	BERT Base Pre-train Time	BERT Large Pre-train Time	Keyword Extraction Time	Time Ratio to BERT Base (%)	Time Ratio to BERT Large (%)
PUBHEALTH	4.35	11.22	1.71	39	15
IMDB	29.98	79.97	9.14	30	11
Amazon	38.21	100.01	7.47	19	7

Table 3: The pre-training time for two epochs, and inference time for KeyBERT in minutes.

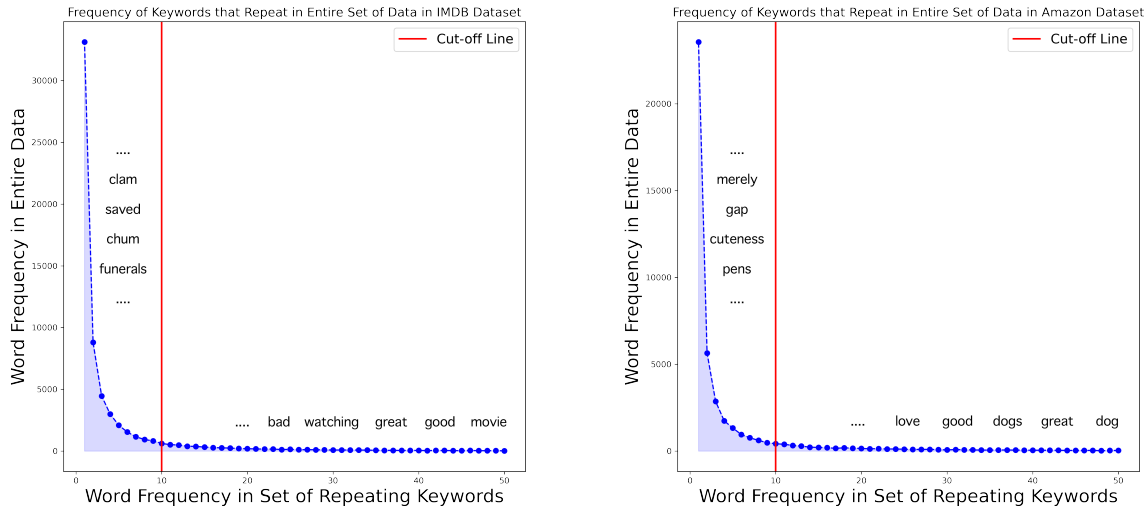

Figure 2: The graphs show the frequency of the last 50 most frequent keywords in IMDB and Amazon datasets along with the cut-off line for removing noisy keywords. For keyword masking, keywords are selected from the subset to the right of the cut-off line (due to space constraints, we do not show the actual lengthy right tail of the charts). A few examples of words that were and were not selected as in-domain keywords given this heuristic are shown on the graphs as well.

annotators is high—substantial or near perfect—which indicates that this task is well-defined. Second, the annotators agreed that the number of moderately- or fully-relevant words is much higher in the keyword-based strategy than in the random masking method. This result further highlights that our masking strategy is indeed relying on the identification of domain-relevant keywords to mask rather than picking up artifacts of the entanglement present in neural architectures (Sculley et al., 2015).

D Implementation of Keyword Masking

To implement our keyword masking strategy, we develop a new data collator by subclassing the Hugging Face data collator for whole word masking (Wolf et al., 2020). Our data collator masks only the tokens according to a certain list of keywords given a probability of masking. Note that our data collator inherits from MLM (Devlin et al., 2019), and no other words or tokens are masked during pre-training except keywords provided by the list.

E Statistical Analysis

We analyze the statistical significance of the obtained improvements using a bootstrap resampling technique with 1,000 samples in the resampling process (Efron, 1979; Efron and Tibshirani, 1993; Efron, 2003). The hypothesis that we investigate is if the results achieved by keyword masking are better than the random masking pre-training strategy. We implement two variants of this hypothesis: one compares F1 scores, and the other compares accuracies.

Table 5 lists the results of this analysis. Overall, the table exhibits that in situations when PLM benefits well from in-domain pre-training, the difference between keyword masking and random masking is statistically significant for both F1 and accuracy scores with p-values ≤ 0.05. The differences are not statistically significant in the two scenarios: the IMDB dataset with BERT Large and PUBHEALTH dataset with BERT Large (only for F1 score). This validates our findings since when BERT Large was employed, the results from

Dataset Name / Masking Method	No. Irrelevant Words	No. Moderately Related Words	No. Related Words	Kappa Value	Level of Agreement
PUBHEALTH / Random Masking	32	6	8	0.84	Near Perfect
PUBHEALTH / Keyword Masking	14	10	16	0.70	Substantial
Amazon / Random Masking	40	5	3	0.87	Near Perfect
Amazon / Keyword Masking	11	17	14	0.72	Substantial
IMDB / Random Masking	42	0	4	0.65	Substantial
IMDB / Keyword Masking	11	7	24	0.73	Substantial

Table 4: The results of measuring inter-rater reliability using Cohen's kappa coefficient for 50 randomly selected words/tokens for masking during in-domain pre-training.

Dataset Name / PLM Name	F1 Score p-value	Accuracy p-value
PUBHEALTH / BERT Base	0.015	0.018
PUBHEALTH / BERT Large	0.505	0.010
IMDB / BERT Base	0.046	0.050
IMDB / BERT Large	0.468	0.454
Amazon / BERT Base	0.000	0.000
Amazon / BERT Large	0.002	0.002

Table 5: The computed p-values for F1 score and accuracy for each of our settings using bootstrap resampling with 1,000 samples.

keyword masking and random masking for the IMDB dataset are quite similar and close to the fine-tuned vanilla PLM. Similarly, in the PUBHEALTH dataset with BERT Large, keyword masking and random masking tie in the F1 score, making the difference in the F1 score not statistically significant. These results further confirm that the benefits of in-domain pre-training vary depending on the domain and the task at hand (Gururangan et al., 2020); however, when in-domain pre-training has a positive impact on performance and causes significant improvement compared to non-adapted setting, our approach outperforms random masking and yields statistically significant gains.

F Detailed Description of Datasets

PUBHEALTH Dataset The PUBHEALTH dataset is divided into three sections: train, test, and validation. Samples in each partition are public health claims with one of four veracity labels including false, unproven, true, or mixture. The labels were assigned by domain experts based on an explanation that they provided for every claim, available in a separate column. These explanations serve as in-domain unstructured data for our use. 9,832 samples in the train split, 1,225 samples in the validation split, and 1,235 samples in the test split form our dataset after a few unlabeled samples were removed.[7]

IMDB Movie Reviews Dataset The two portions of the IMDB dataset are labeled and unlabeled reviews, each having 50,000 reviews. The train, validation, and test splits are generated by dividing the labeled portion by 80%, 10%, and 10%, respectively. That is, 40,000 reviews are allotted to the train split and 5,000 each to the validation and test splits. The unlabeled 50,000 reviews are used for pre-training.

Amazon Pet Product Reviews Dataset There are six different labels for reviews in the Amazon pet product dataset used in the Kaggle competition: dogs, fish aquatic pets, cats, birds, bunny rabbit central, and small animals. The dataset contains four splits: train, test, validation, and unlabeled. However, since the test split does not include labels, we create our own test split by randomly choosing a portion of the train split that is equal in size to the validation split. As a result, in our setting the validation and test splits each includes 17,353 samples; the train split contains 34,704 samples. In addition, there are 100,000 reviews without labels in the dataset's unlabeled portion that serve as pre-training data.

[7] This dataset contains a small number of claims that did not fall under any of the four aforementioned veracity labels.