Domain adaptation of Large Language Models for Sparse Information Retrieval

Katharina Sommer
katharina.sommer@ru.nl

Evgeniia Egorova
evgeniia.egorova@ru.nl

Eugene Shalugin
evgenii.shalugin@ru.nl

KEYWORDS

LoRA, Information Retrieval, MS MARCO, BERT, Domain Adaptation, Transfer Learning

1 INTRODUCTION

The evolution of information retrieval systems has undergone significant advancements, particularly with the advent of Large Language Model-based methods (LLMs). These methods have notably enhanced retrieval capabilities, offering more sophisticated ways to handle and process vast amounts of information. However, the quest for domain-specific relevance in these systems remains a notable challenge. As highlighted in the BEIR benchmark paper [16], achieving high relevance in specific domains is complex due to the unique and specialized nature of domain-specific information.

This challenge is further intensified by the scarcity of annotated ranking datasets, which are often absent in many domains. The lack of these datasets makes the construction of a robust model difficult, rendering simple fine-tuning approaches impractical across various scenarios. Consequently, there exists a pressing need to adapt these models to work efficiently with sparse data, especially in niche or highly specialized fields.

Addressing this gap, our project delves into the domain adaptation of LLMs, particularly focusing on employing transfer learning techniques and Parameter Efficient Fine Tuning (PEFT) [8] to enhance information retrieval. By leveraging transfer learning, we aim to mitigate the need for extensive domain-specific annotated data, thus streamlining the adaptation process. This approach enables the models to learn from a broad base of general data and then apply that knowledge to more specific, domain-related tasks.

The primary goal of our study is to search for a way to elevate the ranking performance of LLM-based retrievers within specific domains. Our framework involves adapting the Large Language Model used in the retriever to grasp the nuances and intricacies of a particular domain through targeted training. This process entails training the LLMs with domain-specific data, followed by the application of transfer learning methodologies to endow the model with specialized ranking capabilities. Our second objective aims to enhance ranking via domain adaptation in the most effective way without losing overall quality for out-of-domain data. For this purpose, we experiment with the PEFT techniques which are declared to significantly decrease the number of stored parameters and accelerate the training process while keeping the performance roughly on the same level. [8]

2 RELATED WORK

The field of domain adaptation in information retrieval, particularly with the use of Large Language Models (LLMs), has seen a massive development in research and innovation. The BEIR benchmark paper [16] stands as a cornerstone in this domain, allowing to systematically evaluate the performance of various retrieval models across diverse domains and highlighting the need for domain-specific adaptation techniques.

Building on this foundation, innovative methods emerged like GenQ and GPL [17], which focus on generating domain-specific ranking data. This approach is particularly crucial given the scarcity of annotated datasets in many domains, a challenge that impedes the effective fine-tuning of LLMs.

In the book Pretrained Transformers for text ranking: Bert and Beyond [13] the authors discuss the utilization and adaptation of BERT and other transformer-based models for information retrieval in specialized domains, offering a comprehensive overview of current practices and future directions.

The paper "Dense Retrieval Adaptation using Target Domain Description" [9] introduces a new category of domain adaptation in information retrieval (IR). It focuses on a scenario where the retrieval model doesn't have access to target documents but can use a brief textual description of the target domain. The paper explores methods like generative pseudo-labeling and answer-aware strategies for domain data selection. It also discusses the use of prompting in language models, distinguishing between instruction-tuned and description-based approaches, and proposes a methodology for adapting dense retrieval models to a target domain using synthetic training sets generated from domain descriptions.

The paper "Domain Adaptation for Dense Retrieval through Self-Supervision by Pseudo-Relevance Labeling" [12] addresses the limited generalization ability of dense retrieval models in target domains with different distributions. It proposes a self-supervision approach where pseudo-relevance labels are automatically generated in the target domain. This involves using the BM25 model for initial document ranking and then re-ranking top documents with the interaction-based model T53B. The approach is further enhanced by combining it with knowledge distillation, using an interaction-based teacher model trained on the source domain. This method demonstrates improved performance over other approaches and helps refine the state-of-the-art query generation approach GPL when fine-tuned on pseudo-relevance labeled data.

3 RESEARCH QUESTIONS

To summarize, in this project, we pose three main research questions to examine and evaluate:

(1) Does Domain Adaptation pre-training enhance the retriever's performance within the domain?
(2) Does pre-training adversely affect scores for out-of-domain data?
(3) How does Parameter-Efficient Fine-Tuning impact pre-training (classification) and transfer learning (ranking) results?

4 METHODS

We decided to treat the evaluation scores presented in the BEIR paper [16] as the baseline results. To make our outcomes comparable, we replicated the exact training setup from the paper. We chose the DeepCT framework [5] as the target approach since the GitHub repository of this project [14] provided all the notebooks and files required for training which allowed us to recreate the same configuration as the authors of the BEIR.

Following the original approach and the BEIR paper, we used BERT *(bert-base-uncased)* [7] as the base model. As a part of domain adaptation, the original model was then fine-tuned on the domain-oriented data for the sequence classification task. At this point, we conducted the experiments both with and without Parameter-Efficient Fine-Tuning (PEFT). The resulting network was then passed to the DeepCT algorithm. Finally, the outcome model was evaluated against in- and out-of-domain datasets and compared to a non-finetuned BERT-based DeepCT retriever as well as to other approaches outlined in the BEIR paper.

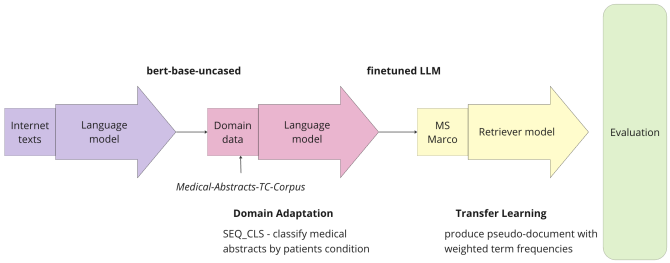

Figure 1: General Pipeline

The sections below provide detailed descriptions of each step in our methodology, outlining the specific procedures, and configurations employed in our experimental setup.

4.1 Domain Adaptation

The first step of the pipeline was the Domain Adaptation. At this stage, we mainly focused on utilizing Parameter-Efficient Fine-Tuning, in particular, the LoRA technique [10]. With LoRA the amount of trainable parameters in the network is drastically reduced: the weights are updated using an update matrix which contains much fewer parameters than the initial network. This approach is supposed to accelerate the training process and allow storing just the adapters with less weights.

For this purpose, we utilized HuggingFace's recently published library that implements multiple PEFT methods including LoRA [2]. However, this decision brought up several limitations regarding the task types available for training – the library supports only a few of those, namely "SEQ_CLS", "SEQ_2_SEQ_LM", "CAUSAL_LM", "TOKEN_CLS", "QUESTION_ANS" and "FEATURE_EXTRACTION".

4.1.1 Unsupervised Learning. The initial idea was to perform the domain adaptation on an unsupervised task with an unlabeled corpus. This would be beneficial as it won't require any domain-specific labeled data for narrow fields and would make the proposed approach versatile and adaptable. Among the task types available for LoRA only two out of six are suitable for unsupervised learning: "CAUSAL_LM" and "SEQ_2_SEQ_LM". However, significant technical challenges arose while attempting to set up training for these tasks.

Causal Language Modelling (LM) is the classic approach for text generation training: the primary goal is to predict the next token in a sequence given the previous tokens. This method would work perfectly with decoder-based architectures like GPT or T5, but not with BERT which naturally is an encoder. The results of the experiments with training BERT for Causal LM were unsatisfactory both with and without PEFT, and since our experimental design hardly relied on using BERT in particular we decided to refute this approach.

As for the SEQ_2_SEQ_LM task, BERT was utilized as both encoder and decoder and the resulting model was trained to retrieve initial sequences from the ones with corrupted tokens. After that, only the encoder part would have been utilized for the next steps. However, the overall performance and predictions were unsatisfying for an already-trained model. This result was rather anticipated as training of the decoder part essentially mirrors causal language modeling, and, as already established, BERT is not suitable for that.

Consequently, after unsuccessful attempts to set up BERT fine-tuning with LoRA for an unsupervised task, it was decided to switch to supervised learning albeit with the drawback of relying on the availability of annotated data.

4.1.2 Supervised Learning. Due to the constraints presented in the previous section, the decision was made to pursue domain adaptation utilizing the "SEQ_CLS" methodology. However, uncertainties arose regarding this approach, given that sequence classification is not inherently focused on language comprehension. Intuitively, this adaptation might not foster the understanding of language nuances or concealed dependencies which is the primary goal of the domain adaptation, but rather emphasize identifying categorical distinctions.

Despite these reservations, we proceeded with this method due to the absence of alternative viable options.

4.2 Transfer Learning with DeepCT

During the domain adaptation, the network was trained for sequence classification with LoRA. During that, the model's weights were not frozen. As the outcome of the training we obtained the LoRA adapter weights which were then merged with the initial vanilla BERT parameters to form a new model. Last, the classification layer was dropped, and the resulting network was provided to the DeepCT framework [5].

The main idea of the DeepCT approach is to use the contextualized text representation from BERT to produce advanced context-aware term weight frequencies (TF) for a passage. Then, based on those TFs each passage can be represented as a pseudo-document that contains "keywords multiplied with the learnt term-frequencies" [16].

The resulting records can be stored in the inverted index and efficiently used for retrieval tasks with BM-25 or ranking functions.

5 DATASET

This section describes the datasets used for the different steps.

For domain adaptation the Medical Abstract TC Corpus [15] dataset was used, which classifies medical abstracts on five different classes of conditions. It consists of a total of 14438 samples, 11550 of them for training and 2888 for testing. The dataset has the following columns: "condition_label" which takes a value between 1-5 depending on the category (condition) of the abstract and "medical_abstract" which holds the abstract. The conditions (categories) in this dataset are as follows: "neoplasms", "digestive system diseases", "nervous system diseases", "cardiovascular diseases" and "general pathological conditions".

For the training with DeepCT the MS-Marco passages [6] dataset was used. It contains 351023 data points and for each entry has a query, term recall, and document.

For the in-domain evaluation the NFCorpus [15] dataset is used. The dataset is available through the BEIR framework and is focused on the medical domain. It contains 323 queries and approximately 3600 documents. For the out-domain evaluation, the FiQA [1] dataset was used. This dataset is also available through the BEIR framework and focuses on the finance domain. It contains 648 queries and approximately 57000 documents.

6 EXPERIMENTAL SETUP

The following section describes the setup used for training BERT with LoRA as well as training the domain-adapted model with DeepCT and evaluating the resulting model. The setup of all steps can be found on GitHub [11].

6.1 Domain Adaptation with LoRA

The domain adaption was done on medical data because of the availability of annotated data for supervised learning as well as three medical datasets in the BEIR framework that can be used for the evaluation. Here, domain adaptation was done using the Medical Abstract TC Corpus [15] dataset.

The fine-tuning of the model was done using the HuggingFace library [2]. Since the model in this step is only fine-tuned using LoRA, a pre-trained BERT model is used [7]. The base BERT model is then fine-tuned on the medical domain using LoRA. The fine-tuning is done using the "SEQ_CLS" task with rank 32, a scaling factor of also 32, a dropout of 0.1, and on the *query* and *value* modules. With these configurations, about 1.1 million out of the 110.6 million (around 1,1%) parameters were trained.

The training was done for 14 epochs with an evaluation after every epoch. We found that the training is most stable around epochs 10-12 so it was decided to use the checkpoint from epoch 11 as the network. This LoRA adapter merged with the base BERT network achieves an evaluation loss of 0.9 and an evaluation accuracy of 0.63 on the Medical Abstract dataset.

To evaluate the effect of PEFT with LoRA, the pre-trained BERT model was also fine-tuned in full, so without using PEFT, with the same dataset for 14 epochs, with 500 warmup_steps and 0.01 weight_decay.

6.2 Transfer Learning with DeepCT

For DeepCT the following implementation was used [5].

The domain adaptation with LoRA outputs a PyTorch safetensor file with the trained network. To use this network with DeepCT it has to be converted to TensorFlow.

The network is then trained on the MS Marco dataset for one epoch with a maximum sequence length of 128, a batch size of 16, and a learning rate of 0.00002.

6.3 Evaluation

The evaluation was done using the evaluation script for DeepCT from BEIR [16].

For the in-domain evaluation the NFCorpus [4] dataset provided by BEIR is used because it is a publicly available dataset for the medical domain. For the out-of-domain evaluation, the FiQA [1] dataset, a dataset about finances, is used. The reason for using the FiQA dataset was to have a dataset of a completely unrelated domain in comparison to the medical one to analyze if fine-tuning on one specific domain decreases the performance of other domains.

6.4 Technical Difficulties

This section describes some of the technical difficulties that occurred during the setup of our experiments and how these problems were solved.

6.4.1 Tensorflow Version. The first problem that we encountered was that in the original project, the DeepCT algorithm was implemented using TensorFlow version 1. Since our setup was on Google Colab which only supports Python 3 and therefore only TensorFlow version 2 it was necessary to upgrade the code to the TF v2 to be able to run it. To achieve that, we used a script from TensorFlow to perform all the necessary conversions automatically [3]. Nonetheless, after applying the script some parts of the code (for instance, the layer normalization) remained still written in TF v1 so it had to be converted manually. An overview of all the statements that needed to be changed can be found in the DeepCT notebook on our GitHub [11].

6.4.2 Network Conversion. After the domain adaptation with LoRA, we obtained the adapter which then was merged with the weights of the initial base BERT. This whole part was done using the PyTorch framework and HuggingFace library, so the resulting network was saved in the native Hugging Face format of safetensors. However, the next step of our pipeline, the DeepCT, was only accepting a TensorFlow checkpoint file. None of the frameworks provided a direct conversion between the two formats, so we had to manually iterate over all the layers of the PyTorch model and map the weights to TF checkpoint. The implementation of this algorithm is listed in the DeepCT.ipynb on the GitHub page of the project [11].

6.4.3 Evaluation Setup. To be able to compare our results to the one of the BEIR paper the decision was made to use the evaluation script provided by BEIR. Because part of this evaluation script uses Docker, which does not work with Google Colab, it was necessary to set up a virtual Linux machine to be able to run the Docker part of the evaluation. Another problem that occurred was that while receiving the results from the Docker container it would give Java heap overflow errors which would lead to different results every time it was run. Therefore, it was necessary to start the Docker container with more memory for Java operations.

7 RESULTS AND DISCUSSION

This section discusses our results and findings of the experiments that were conducted and compares them with each other.

(1) Figure 2 shows the training and validation loss for the domain adaptation with and without LoRA. The red lines indicate which checkpoints were used for DeepCT. It can be seen that the checkpoint at 17000 steps has the lowest validation loss before it starts going up again. Therefore this checkpoint as well as one before that, one after that, and the last one were taken to compare the results at different iterations.
It can also be seen that vanilla fine-tuning leads to fast and severe overfitting. This does not happen in the network with LoRA which is due to less trainable parameters. By reducing the amount of trainable parameters, the capacity of the model is artificially reduced. Therefore the network with LoRA is not adapting exactly to the dataset which makes it more robust to overfitting.

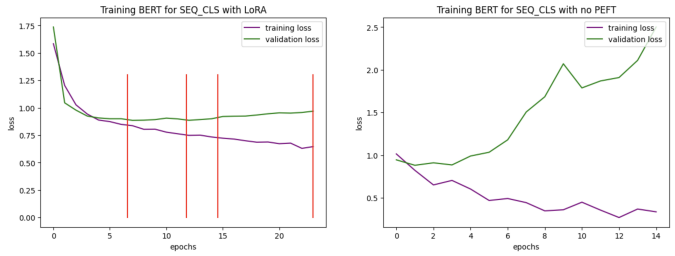

Figure 2: Training for Sequence Classification with and without LoRA

(2) Figure 3 shows the training loss for DeepCT for the different networks. The evaluated networks are the one without any domain adaption (no DA), four with domain adaptation and LoRA performed for N steps (DA for N steps), and the one with domain adaptation but without using PEFT.

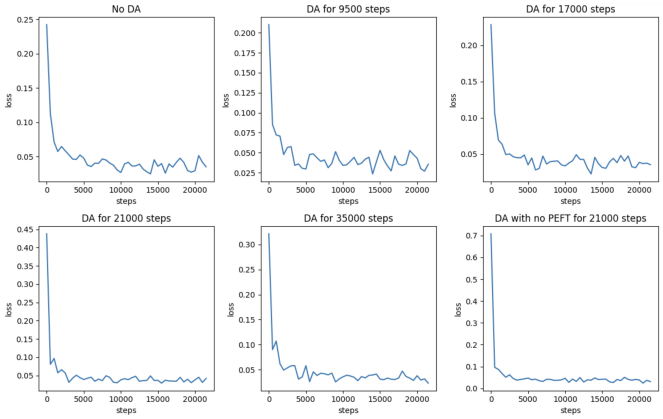

Figure 3: DeepCT training loss for ranking

It can be seen that the train loss for ranking is about the same for every network. In terms of the speed and stability, the training curves also look similar for all of the experiments.

(3) Table 1 shows the evaluation results for the medical and finance datasets for the different DeepCT networks. The bold values represent the highest NDCG@10 scores per dataset, while the underlined values mark the networks that outperformed the vanilla BERT without any domain adaptation. We also added the scores outlined in the BEIR benchmark [16] as the baseline.

	no DA	DA 9500	DA 17000	DA 21000	DA 35000	no PEFT DA 21000	baseline
MED	0.3273	0.3260	0.3282	0.3294	**0.3314**	0.3286	0.283
FIN	0.2690	**0.2710**	0.2602	0.2597	0.2693	0.2730	0.188

Table 1: NDCG@10 for Different Configurations

The following two figures represent the same results for in-domain evaluation (Figure 4) and out-domain evaluation (Figure 5).

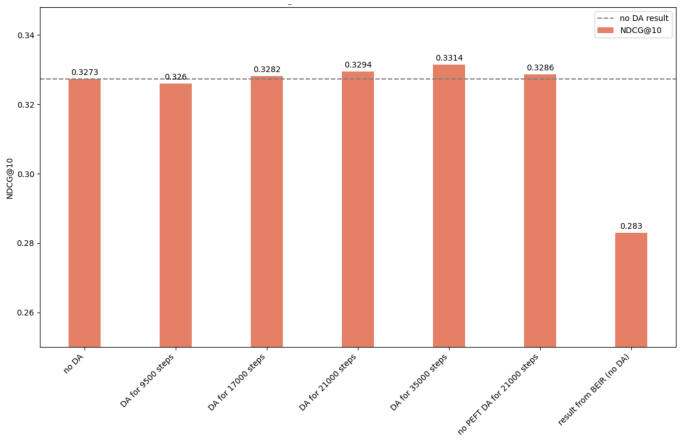

Figure 4: NDCG@10 for Medical (in-domain) dataset

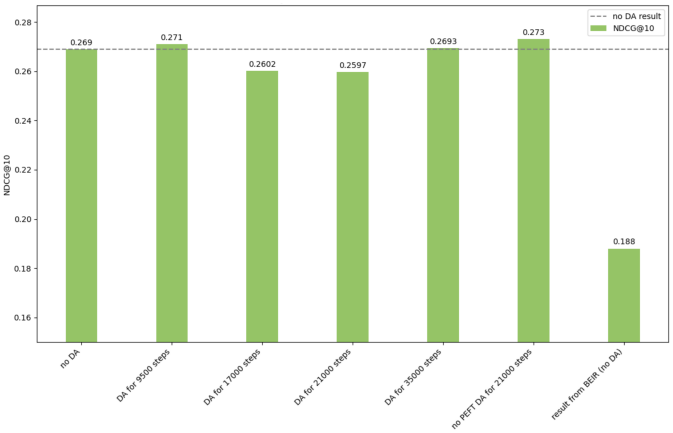

Figure 5: NDCG@10 for Finance (out-domain) dataset

It can be seen that for the medical dataset, on which domain adaptation was done, the performance generally improves

with an increase of training iterations for domain adaptation with PEFT. In some perspective, these results confirm our initial hypothesis: in the conducted experiments the larger the extent of domain adaptation was, the more the performance increased within the domain. However, the margin of the improvement appeared to not be that significant: NDCG@10 of 0.3273 with no DA in comparison to 0.3314 with 35000 steps of DA, so the difference is only 0.4%.

As for the finance out-of-domain dataset, domain adaptation leads to an insignificant drop (not more than 1%) in quality for models with a small amount of DA, but levels it up for the models with a larger extent of DA. Such an outcome does not align with our expectations, nonetheless, the differences between the scores are still not large enough to make any claims.

Compared to the baseline results of the BEIR paper, our networks perform significantly better, for both in and out-of-domain evaluation. We expected that our vanilla BERT model would roughly produce the same results as the baseline of the paper but in our results, these two networks differ significantly. This fact does not allow us to conclude that domain adaptation is useful in this scenario because the improvements from the domain adaptation are not significant compared to our vanilla BERT. We attempted to replicate the setup of the BEIR paper as closely as possible but there is the chance that we performed the training with other hyperparameters which can explain the difference between our results and the one from the paper.

(4) Finally, the fine-tuning without PEFT does not give a significant gain in comparison to fine-tuning it with LoRA. The model trained with PEFT scored 0.3294 while the network based on BERT that was domain-adapted for the same number of steps but without LoRA scored 0.3286. Therefore, we can conclude that using LoRA does not affect the evaluation results. It is important to mention that using PEFT, more specifically LoRA, for domain adaptation is less time-consuming as during domain adaptation less amount of parameters are fine-tuned. Namely, vanilla fine-tuning took 5 hours for 14 epochs with batch_size = 16, while PEFT domain adaptation took 4 hours for 14 epochs with batch_size = 8.

In conclusion, our results depict an upward trend for the in-domain evaluation but the differences are not significant enough to make any statements about the efficiency of domain-adaption of LLMs for ranking, so some future explorations might be required.

Also, another interesting approach to examine would be to use a more language comprehension-oriented task for the domain adaptation, for instance, Masked Language Modelling. As already mentioned, during the sequence classification a network might not focus on learning to understand the language but rather emphasize identifying categorical distinctions. Therefore, it doesn't bring much difference to the resulting retrieval task. It might prove beneficial for the retrieval task performance to experiment with the domain adaptation task as the network would learn to model the language rather than classify sequences of texts.

REFERENCES

[1] 2018. FiQA. https://sites.google.com/view/fiqa/. Accessed: 2023-12-10.

[2] 2023. HuggingFace. https://huggingface.co/. Accessed: 2023-12-06.

[3] 2023. Migrate Tensorflow. https://www.tensorflow.org/guide/migrate/upgrade. Accessed: 2023-12-11.

[4] Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A Full-Text Learning to Rank Dataset for Medical Information Retrieval. *Proceedings of the 38th European Conference on Information Retrieval.* http://www.cl.uni-heidelberg.de/~riezler/publications/papers/ECIR2016.pdf

[5] Zhuyun Dai and Jamie Callan. 2019. Context-Aware Sentence/Passage Term Importance Estimation For First Stage Retrieval. arXiv:1910.10687 [cs.IR]

[6] Zhuyun Dai and Jamie Callan. 2019. DeepCT Train Data File. http://boston.lti.cs.cmu.edu/appendices/arXiv2019-DeepCT-Zhuyun-Dai/data/.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[8] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence* 5, 3 (2023), 220–235.

[9] Helia Hashemi, Yong Zhuang, Sachith Sri Ram Kothur, Srivas Prasad, Edgar Meij, and W Bruce Croft. 2023. Dense Retrieval Adaptation using Target Domain Description. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval.* 95–104.

[10] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL]

[11] Evgenii Shalugin Katharina Sommer, Evgeniia Egorova. 2023. InformationRetrival. https://github.com/SommerKatharina/InformationRetrival.

[12] Minghan Li and Eric Gaussier. 2022. Domain Adaptation for Dense Retrieval through Self-Supervision by Pseudo-Relevance Labeling. *arXiv preprint arXiv:2212.06552* (2022).

[13] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2022. *Pretrained transformers for text ranking: Bert and beyond.* Springer Nature.

[14] Andreas Rücklé Abhishek Srivastava Iryna Gurevych Nandan Thakur, Nils Reimers. 2021. BEIR. https://github.com/beir-cellar/beir.

[15] Tim Schopf, Daniel Braun, and Florian Matthes. 2023. Evaluating Unsupervised Text Classification: Zero-Shot and Similarity-Based Approaches. In *Proceedings of the 2022 6th International Conference on Natural Language Processing and Information Retrieval* (Bangkok, Thailand) *(NLPIR '22).* Association for Computing Machinery, New York, NY, USA, 6–15. https://doi.org/10.1145/3582768.3582795

[16] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2).* https://openreview.net/forum?id=wCu6T5xFjeJ

[17] Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2021. Gpl: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. *arXiv preprint arXiv:2112.07577* (2021).