Unsupervised Dense Retrieval for Scientific Articles

Dan Li and **Vikrant Yadav** and **Zubair Afzal** and **Georgios Tsatsaronis**
Elsevier
{d.li, v.yadav, zubair.afzal, g.tsatsaronis}@elsevier.com

Abstract

In this work, we build a dense retrieval based semantic search engine on scientific articles from Elsevier. The major challenge is that there is no labeled data for training and testing. We apply a state-of-the-art unsupervised dense retrieval model called Generative Pseudo Labeling that generates high-quality pseudo training labels. Furthermore, since the articles are unbalanced across different domains, we select passages from multiple domains to form balanced training data. For the evaluation, we create two test sets: one manually annotated and one automatically created from the meta-information of our data. We compare the semantic search engine with the currently deployed lexical search engine on the two test sets. The results of the experiment show that the semantic search engine trained with pseudo training labels can significantly improve search performance.

1 Introduction

Search engines are deeply integrated into Elsevier's information services of its scientific literature data. An example is the one provided by ScienceDirect[1], providing researchers with search services on more than 19M full text articles. These search engines are currently based on lexical search models such as BM25. The deployment of such models is effortlessly simplified by using popular industry-standard libraries such as Elasticsearch[2]. However, lexical search suffers from the lexical gap problem such as misspellings, synonyms, abbreviations, ambiguous words, and ignoring of word order (Formal et al., 2021).

Recently, dense retrieval (DR) models have proven to be highly effective in solving the lexical gap problem while still remain fast search speed (Karpukhin et al., 2020; Xiong et al., 2020). DR models map queries and passages to a common vector space and retrieve relevant passages by searching for (approximate) nearest neighbors. DR has been well studied on laboratory data but still in the early stage for industry-level applications (Hofstätter et al., 2022; Kim, 2022). DR is mainly applied in multi-modal search in industry where traditional lexical search is not possible, like text-image search (Radford et al., 2021) or music search (Castellon et al., 2021).

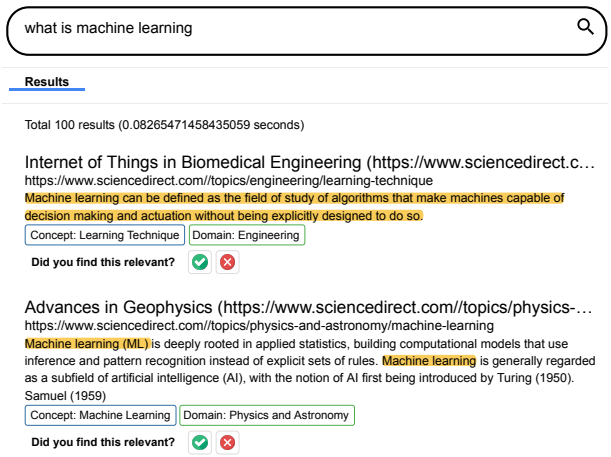


Figure 1: Interface of our semantic search engine.

It is of great interest to use state-of-the-art DR models to build semantic search engines for industry. Such search engines can enable efficient access and search to scientific literature of Elsevier and help researchers in their journey (Elsevier, 2022). Our goal in this work is to develop a semantic search engine that needs no relevance-labeled data to train the DR model, thus allowing easy adaptation to new domains and easy deployment in industry.

There are several challenges to be tackled. First, training a DR model requires sufficient labeled data such as MS-MARCO (Nguyen et al., 2016), whereas there is often no such data for specific domains or startups. In our case, we have a large collection of passages from scientific articles but no relevance label. Furthermore, it is shown that DR

[1]https://www.sciencedirect.com
[2]https://www.elastic.co

Proceedings of EMNLP 2022 Industry Track, pages 323–331
December 9–11, 2020.

models trained on one domain do not generalize to another (Thakur et al., 2021). The passages in our corpus have a different word distribution compared to that in MS-MARCO. Besides, the passages are also unbalanced regarding their domains (see Section 4.3). Therefore, using the models trained on MS-MARCO will not yield high retrieval performance. It is interesting to tackle the domain difference problem. Finally, there is no test set to evaluate search performance and creating a good-quality test set is time-consuming and expensive. All these challenges hinder the application of DR models in industry setting.

In this work, we trained a DR model using a state-of-the-art unsupervised dense retrieval model called GPL (Wang et al., 2021). It uses a pre-trained query generator to generate queries from passages. The passage is considered as positive for the generated queries. Negative passages for generated queries are retrieved using existing dense retrieval models trained on MS-MARCO. An existing cross-encoder model trained on MS-MARCO produces relevance scores of query-passage pairs as supervision signals to train the DR model.

Finally, we constructed two test sets by either manual annotation or automatic extraction of existing relevance information from the meta field of the corpus. The experimental results show that our best model can significantly improve the retrieval performance compared to lexical and semantic search baselines.

The semantic search engine we have created for our product is shown in Figure 1. It is currently deployed and running in a beta test mode.

2 Related Work

2.1 Dense retrieval

The very first work on dense retrieval (DR) was proposed by Karpukhin et al. (2020). DR uses text encoders to represent queries and documents as dense vectors and retrieve documents by similarity scores between query vectors and document vectors. It has shown to achieve competitive performance in first-stage retrieval compared with traditional lexical retrieval method.

Researchers have been working towards improving the effectiveness of DR models through negative sampling (Xiong et al., 2020; Zhan et al., 2021; Lin et al., 2021), pre-trained language models (Gao and Callan, 2021), and pseudo relevance labels (Prakash et al., 2021; Yu et al., 2021), as well as improving the efficiency of DR models with sparse representation (Zhan et al., 2022; Thakur et al., 2022).

2.2 Unsupervised dense retrieval

Unsupervised dense retrieval (UDR) aims to train dense retrieval models without manually labeled data. It generates high-quality pseudo labeled data and designs proper loss functions to train DR models.

The first step is to generate positive examples, which is done by extraction or generation. For example, Izacard et al. (2021) extracted a pair of relevant texts form the same document using the inverse cloze task and independent cropping. Wang et al. (2021) generated queries from documents using existing encoder-decoder as positive examples.

The second step is to generate negative examples. Izacard et al. (2021) used contrastive loss to create negative batches within a batch and across batches. Wang et al. (2021) used existing weak retrievers to retrieve top-k documents as negatives.

The third step is to design training loss. Due to the noisy fact of pseudo examples, traditional pairwise ranking loss (Burges, 2010) is not a good choice because the training are easily affected by noisy labels. Instead, contrastive loss is widely used (Izacard et al., 2021; Xu et al., 2022). On the other hand, relevance scores from existing generalizable cross-encoder have been used as supervision signal (Wang et al., 2021).

3 Methodology

3.1 GPL Model Training

Since there are no relevance labels for the passages in our corpus, we apply a recent unsupervised dense retrieval model GPL (Wang et al., 2021) to train our dense retrieval model. We generate 3 queries from each passage using a pre-trained query generator (Nogueira et al., 2019). The passage-query pairs will be the pseudo positive examples. For each generated query, we retrieve similar passages using two existing DR models trained on the MS-MARCO dataset (Reimers and Gurevych, 2019), and take the first 50 of each model as pseudo negatives. Finally, we use a student-teacher training method. The teacher model is a cross-encoder trained on MS-MARCO which shows good performance in zero-shot retrieval tasks (Hofstätter et al., 2020). The student model is the bi-encoder DR model to be learned.

The student-teacher training is used because the pseudo labels are noisy and can not be directly used in the traditional pairwise ranking loss (Burges, 2010) or contrastive loss (Wu et al., 2018). Instead, using a cross-encoder has been demonstrated to generalize well on different datasets (Hofstätter et al., 2020) and thus can be used as a teacher model through knowledge distillation.

For the knowledge distillation we have used MarginMSE loss (Hofstätter et al., 2020). It is defined as:

$$L_{MarginMSE} = -\frac{1}{M}\sum_{i=0}^{M-1} |\hat{\delta}_i - \delta_i|^2 \quad (1)$$
$$\hat{\delta}_i = f_{be}(q_i)^T f_{be}(p_i^+) - f_{be}(q_i)^T f_{be}(p_i^-)$$
$$\delta_i = f_{ce}(q_i, p_i^+) - f_{ce}(q_i, p_i^-)\,,$$

where f_{be} is the bi-encoder, which maps the text of query or passage to a vector, f_{ce} is the cross-encoder, which maps the text of query and passage to a score, q_i is the query, p_i^+ is the positive passage, and p_i^- is the negative passage.

By minimizing $L_{MarginMSE}$, the MarginMSE loss avoids the hard treatment of the positives and negatives as in pairwise ranking loss (Burges, 2010) and contrastive loss (Wu et al., 2018). For example, for $(false\ positive, negative)$ pairs or $(positive, false\ negative)$ pair, we do not expect the bi-encoder to put them far away in the embedding space or have small similarity scores. The cross-encoder will assign a small δ value and the MarginMSE loss will teach the bi-encoder to produce small $\hat{\delta}$ value as well.

Implementation. We use *all-t5-base-v1* as the query generator because it is designed to generate key-word queries, which is similar with the terms or topics people search in our product. We use *msmarco-distilbert-base-v3* and *msmarco-MiniLM-L-6-v3* as the negative retrievers, and *ms-marco-MiniLM-L-6-v2* as the teacher cross-encoder as suggested in GPL. We use *sebastian-hofstaetter/distilbert-dot-tas_b-b256-msmarco* as the starting checkpoint of the student bi-encoder because this is the best bi-encoder on MS-MARCO. The teacher model and the student model contain 22M and 66M parameters, respectively. All the models aforementioned can be downloaded from Huggingface[3]. We set batch size 16. We set maximum sequence length 512. Note that the passages are snippet from the articles and have on average 474 English words or 723 WordPiece (Wu et al., 2016) tokens. Cutting off of the passages loses information. It is worthy split the passages into shorter ones and we leave the work for future study.

3.2 Test Set Construction

Corpus The corpus we are working on supports a web service providing concept definitions and subject overviews for researchers who want to expand their knowledge about scholarly and technical *terms*.[4] For example, for the term "water purification", a web page is created that contains its definition, several scientific article snippets containing other definitions of the term, and several relevant terms as well. The corpus contains about 2M passages extracted from scientific articles. The articles are from 20 different domains and not evenly distributed across domains. Figure 3 shows the domain distribution.

Manual test set. We aim to develop a semantic search engine on top of this corpus, so that when a user searches a term, the semantic search engine returns passages containing the definition of the term. Therefore, the ideal queries are questions about scientific terms, and the ideal relevant passages are those talks about (part of) the definitions the terms.

As the data contains scientific terms from 20 domains, we sample one term from each. We only sample those having Wikipedia pages to increase the chance that there exists relevant passages for a query.

We use the widely-used pooling method in information retrieval (Ferro and Peters, 2019) to select passages for annotation. We include 3 different retrieval systems in the pool including the BM25 model (Pérez-Iglesias et al., 2009), the TAS model (Hofstätter et al., 2021), and the GPL model trained by us, in order to ensure the passages in the test set are diverse and not biased towards either lexical retrieval or semantic retrieval methods. We randomly sample from the top-10 passages in the ranking lists.

We had 3 workers annotating the selected query-passage pairs. Conflicts of annotation were discussed until an agreement was reached. Finally, 20 queries and 539 query-passage pairs are selected and annotated.

Auto test set. Although the manual test set has high quality, it is too small and thus sensitive for

[3]https://huggingface.co/models

[4]https://www.sciencedirect.com/topics/index

evaluation. We use the noisy meta information in our corpus and construct a larger test set. The passages in the corpus is organised by terms. Each term has several passages associated with it that are considered relevant and containing the definition of the term. The extraction of the definition and relevant passages are done by production system based on lexical methods. Thus the passages can be roughly taken as relevant to that term. To balance terms from different domains, we sample 10 terms from each domain and take all the passages associated with it as relevant. Finally, we have 200 queries and 3,562 relevant labels.

3.3 System Architecture

Figure 2 shows the architecture of the semantic search engine. The system is divided into two parts, offline and online. In the offline part, we download the corpus from Amazon S3 buckets, then on Amazon Sagemaker we preprocess the corpus, train the the bi-encoder model and convert the passages into 768-dimensional vectors using the trained model. The HNSW[5] algorithm is used to index the passages.

The online part is divided into two parts. One of the parts is an API-based service running on Amazon Sagemaker. The task of this service is to convert the user query into a vector and find the passages closest to the query vector using the index we created in offline mode. The other part is a UI based interface running on an Amazon EC2 instance. This part processes the user query and displays the passage associated with the query through a UI interface.

The EC2 instance and API run on an Intel Xeon-based processor and the cost of running them is 1 dollar per hour. For training the model, we use the AWS p3.8x.large EC2 instance type. This instance is installed with NVIDIA Tesla V100 GPUs. The cost of training the model was approximately 200 dollars. During inference time, the system is running on a CPU instance and it is able to process up to 70 requests/second. The average time needed to get the search result for a query is 0.03 second.

4 Experimental Setup

4.1 Research Questions

(RQ1) How does the model perform compared with the current production model and other baselines?

[5]https://github.com/nmslib/hnswlib

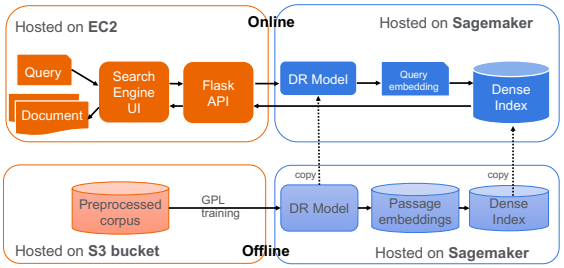


Figure 2: Architecture of the semantic search engine.

(RQ2) Is it necessary to use the whole corpus to train the model?

(RQ3) Whether balancing passages from different domains in training batches improves model performance?

4.2 Baselines

BM25. This baseline is the current search engine in production. It uses lexical retrieval model BM25 implemented in Elasticsearch.

TAS (Hofstätter et al., 2021). The *sebastian-hofstaetter/distilbert-dot-tas_b-b256-msmarco* model is a zero-shot baseline. It was the best bi-encoder on MSMARCO when this paper was submitted. We also use this model as the starting checkpoint to train the GPL model.

BM25+CE. This is a two-stage baseline implemented by us. It consists of lexical retrieval and re-ranking. We first use BM25 to produce a ranked list of passages, then use a cross-encoder *ms-marco-MiniLM-L-6-v2* trained on MSMARCO to rerank the top-1000 passages. We use this model as the teacher model when training GPL.

4.3 Dataset

We use two test sets including the Manual and the Auto. Table 2 shows the statistics. The Manual has 20 queries and the Auto has 200 queries. Auto also has more labels for query-passage pairs. Note there is 0 non-relevant labels for Auto, however this does not affect the evaluation as all the rest passages without a relevant label will be counted as non-relevant. To speed up evaluation, we randomly sample a subset of passages for the models to retrieve from, combined with the passages in each of the two test sets. This results in two test corpora consisting of 100,513 and 102,506 passages for the Manual and the Auto. The test corpora have the same domain distribution with the full corpus.

Model	Manual test set				Auto test set			
	NDCG@10	MAP@10	MRR@10	R@100	NDCG@10	MAP@10	MRR@10	R@100
BM25[1]	61.86	42.59	77.38	87.05	**53.08**	**24.17**	68.81	**70.81**
TAS	47.78	25.98	74.17	67.64	28.20	9.97	46.81	40.16
GPL	71.29	42.42	85.70	91.71	49.78	22.44	74.21	59.41
GPL_BLC	**74.42**	**44.96**	**87.62**	**91.77**	50.16	22.47	**75.49**	59.69
BM25+CE[2]	**84.90**	**54.96**	**95.00**	**90.99**	**68.33**	**36.87**	**86.68**	**78.56**

[1] Production model of our product.
[2] Upper bound of our model.

Table 1: Retrieval performance (%). The best values for each metric and the upper bound method is in bold.

	Manual	Auto
Test set		
# query	20	200
# passage	539	2614
# relevant	289	3562
# non-relevant	251	0
Test corpus		
# passage	100513	102506

Table 2: Statistics of test sets.

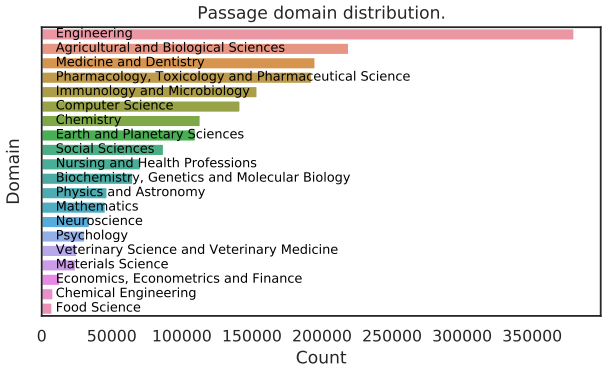


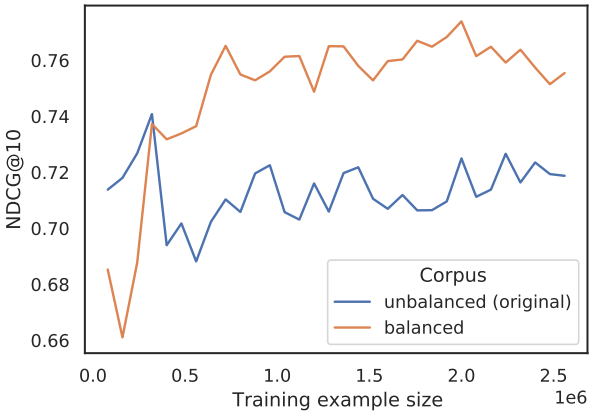
Figure 3: Passage domain distribution. The top 5 domains cover about 58.1% of the passages and the bottom 5 domains only contains about 3.97% of the passages.

5 Results

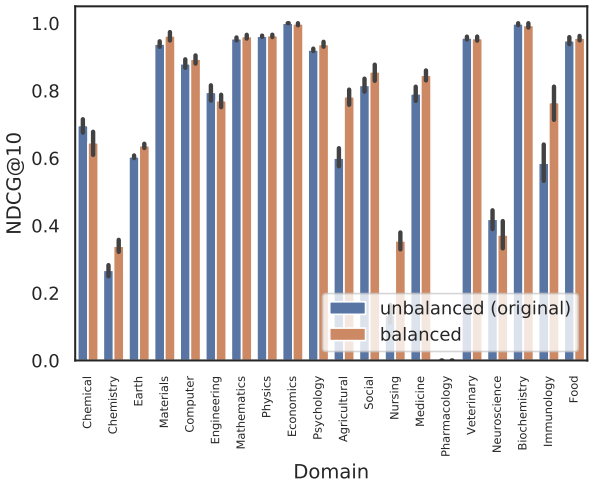
5.1 Retrieval performance

In this section, we aim to answer RQ 1. We use a subset of 83K passages from our corpus and generate 3 queries for each passages and generate 100 negative passages for each query. Finally, we sample 4M training examples in the format of $(q_i, p_i^+, p_i^-, \delta_i)$. It is suggested that such a volume is enough to train a GPL model for a new domain (Wang et al., 2021). We also empirically demonstrate the impact of training example size in Section 5.2. We train two GPL models: GPL is trained on 83K passages randomly sampled. GPL_BLC is trained on 83K passages which are balanced sampled from the 20 domains. Since we aim to build a one-stage retrieval model, we compare our model with a lexical retrieval model – BM25 and a zero-shot dense retrieval model – TAS. We also compare with a two-stage method – BM25+CE.

Table 1 shows the retrieval performance. First, BM25 performs robustly well on the two test sets, while zero-shot TAS performs poorly. It indicates that dense retrieval models do not generalize well on new domain. This finding is consistent with the work of Thakur et al. (2021). The difference of metrics between BM25 and TAS is larger on Auto, because we have annotated both lexical and semantic relevant passages in the Manual test set while most relevant passages in the Auto test set are obtained by lexical methods only. The dense retrieval model TAS is thus down-estimated on Auto. Second, BM25+CE performs the best. It improves NDCG@10, MAP@10, and MRR@10 to a large margin compared to BM25. The cross-encoder model (*ms-marco-MiniLM-L-6-v2*) is trained on MS-MARCO. Thus, the result indicates the good generalization capability of cross-encoder ranking models. Third, GPL or GPL_BLC perform better than BM25 on most the metrics and better than TAS on all the metrics. For example, an MRR@10 of 87.62 means that GPL_BLC can rank relevant passages on the first or second position on averaged queries, an R@100 of 91.77 means that GPL_BLC can retrieve 91.77% of the relevance passages in top 100. Note that the performance difference between GPL and GPL_BLC is big on Manual but small on Auto. The possible reason is that on Auto most semantically relevant passages are not labeled in the test set.



(a) NDCG@10 averaged over queries against training example size.



(b) Query-wise NDCG@10 of model trained with all the 4M training examples.

Figure 4: NDCG@10 of the Unbalanced and Balanced corpus.

Figure 5: NDCG@10 of GPL model trained with different number of examples. The x-axis is from 0.1×10^7 to 3.2×10^7.

5.2 The impact of training example size

In this section, we aim to answer RQ 2. We use all the 2M passages in the corpus and generate 32M training examples to train the model. We save the checkpoint every 160K examples. We evaluate model performance on the Manual test set. Figure 5 shows the NDCG@10 score against the training example size. We observe that more training examples do help to improve the performance of the model. The performance increases fast at the beginning and achieves an NDCG@10 of 0.74 with about 1M training examples, it then increases slowly towards an NDCG@10 of 0.80.

To sum up, it is not necessary to train the GPL model with all passages in our corpus; a volume of 1M training examples should be sufficient for the model.

5.3 The impact of domain distribution

In this section, we aim to answer RQ 3. Since there is meta information about what domain the passages belong to in our corpus, we compare the model trained on the random 83K passages (*Unbalanced*) and the model trained on the 83K evenly distributed in the 20 domains (*Balanced*). Figure 4 shows the NDCG@10 of corpus 83K and 83K-balance. We use the Manual set as the test set. We observe that (1) there is a large improvement on 83K-balanced compared to 83K-unbalanced; (2) the NDCG@10 increase for most queries, and the improvement is especially large for those with low NDCG@10.

5.4 Case study

In this section, we show one query and the top 3 ranked passages selected from the Manual test set to analyze the retrieval effectiveness. We showcase three models including BM25, TAS, and GPL. The case study helps us to know how the retrieved passages are different for the DR model trained on the target domain, the zero-shot DR model and the lexical retrieval model. BM25, as expected, retrieves passages containing exact match of words in the query. As it is a bag-of-word model, we observe that the word "water" and "purification" do not always appear together in the passages. TAS can retrieve semantically similar passages, but they are sometimes off the topic. For example, the 1st passage retrieved by TAS is about "fuel purification", it even contains the definition. However, it is not about "water purification". TAS_GPL can retrieve relevant passages which even contain the definition.

Model	BM25	TAS	GPL
Query	What is Water Purification		
1st passage	...Importance of purification. Physicochemical properties of our model system. Adsorption layer of a nonionic surfactant. Ionic surfactant at the air water interface...	...The purification process is shown schematically in Figure 7-38. Fuel purification is a one-stage extraction procedure which employs centrifuges to treat distillates...	...Water purification for human consumption purposes consists in the removal of different contaminants as chemicals (i.e., pollutants, toxic metals), biological contaminants...
Relevance	0	0	2
2nd passage	... Such basic issues have to be addressed ahead of any assessment of water purification technologies, since such purification may not even be necessary...	...Purification is practical with distillate fuel and light crude oils having a minimum 0.5% water in the fuel, with a...	...Fuel purification is a one-stage extraction procedure which employs centrifuges to treat distillates and light crude oils without adding water...
Relevance	1	0	0
3rd passage	...Preparation of clarified growth media from an overnight culture of bacterial cells is the first and perhaps most important step in purifying OMVs. Before proceeding to purification...	...Disinfection, the desired result of field water treatment, means the removal or destruction of harmful microorganisms. Technically, it refers to chemical means such as...	The terms "water treatment" and "water purification" are extensively used for any unit operations and processes that involve methods and processing steps ...
Relevance	0	2	2

Table 3: Case study.

For example, the 1st passage is a good definition of "water purification". The case clearly shows that lexical retrieval and dense retrieval find very different passages. This is because their ways of representing texts are completely different. Furthermore, training DR models on the target domain can improve retrieval performance to a large margin even though the training labels are noisy.

6 Conclusions & Future Work

In this work, we build a semantic search engine on scientific articles. To tackle the challenge of no labeled data for both training and test, we apply a state-of-the-art unsupervised dense retrieval model named GPL. As the articles are unbalanced across different domains, we sample passages from multiple domains to form balanced training batches. We also created two test sets for the evaluation: one manually annotated and one automatically constructed from the meta information of our corpus.

We compare the semantic search engine with the currently deployed lexical search engine on the test sets. Both the qualitative and quantitative experiment results show that the semantic search engine can significantly improve the search performance. This results suggest that GPL is a robust and effective model for unsupervised dense retrieval.

For the future work, we will train the query generator and the negative retriever on our data to generate a better quality of both positive and negative training example to improve the retrieval performance.

7 Limitations

Currently, we see 3 limitations in our work. First, the query generator is trained on a different domain, which results in skipping important keywords or phrases around which the query should be generated. Second, the negative retrievers are not adapted to the domain. The results obtained by these retrievers are negative but not "hard negative". This leads to limitations in learning of the student model. Third, the semantic search engine we build has not been evaluated on production population. We plan to conduct online evaluation in the future.

References

Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11(23-581):81.

Rodrigo Castellon, Chris Donahue, and Percy Liang. 2021. Codified audio language modeling learns useful representations for music information retrieval. *arXiv preprint arXiv:2107.05677*.

Elsevier. 2022. All elsevier digital solutions.

Nicola Ferro and Carol Peters. 2019. *Information Retrieval Evaluation in a Changing World: Lessons Learned from 20 Years of CLEF*, volume 41. Springer.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2288–2292.

Luyu Gao and Jamie Callan. 2021. Condenser: a pre-training architecture for dense retrieval. *arXiv preprint arXiv:2104.08253*.

Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving efficient neural ranking models with cross-architecture knowledge distillation. *arXiv preprint arXiv:2010.02666*.

Sebastian Hofstätter, Nick Craswell, Bhaskar Mitra, Hamed Zamani, and Allan Hanbury. 2022. Are we there yet? a decision framework for replacing term based retrieval with dense retrieval systems. *arXiv preprint arXiv:2206.12993*.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. In *Proc. of SIGIR*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Towards unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Yubin Kim. 2022. Applications and future of dense retrieval in industry. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3373–3374.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021. In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 163–173.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPs*.

Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019. From doc2query to doctttttquery. *Online preprint*, 6.

Joaquín Pérez-Iglesias, José R Pérez-Agüera, Víctor Fresno, and Yuval Z Feinstein. 2009. Integrating the probabilistic models bm25/bm25f into lucene. *arXiv preprint arXiv:0911.5046*.

Prafull Prakash, Julian Killingback, and Hamed Zamani. 2021. Learning robust dense retrieval models from incomplete relevance labels. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1728–1732.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, and Jimmy Lin. 2022. Domain adaptation for memory-efficient dense retrieval. *arXiv preprint arXiv:2205.11498*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

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*.

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*.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings*

of the IEEE conference on computer vision and pattern recognition, pages 3733–3742.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2022. Laprador: Unsupervised pretrained dense retriever for zero-shot text retrieval. *arXiv preprint arXiv:2203.06169*.

HongChien Yu, Chenyan Xiong, and Jamie Callan. 2021. Improving query representations for dense retrieval with pseudo relevance feedback. *arXiv preprint arXiv:2108.13454*.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1503–1512.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2022. Learning discrete representations via constrained clustering for effective and efficient dense retrieval. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1328–1336.