Retrieval-Augmented Retrieval for Zero-Shot Dense Retrieval Adaptation

Shi Yu[♡], Xiangyang Li[♣], Zhao Cao[♣], Zhiyuan Liu[♡]
[♠] Dept. of Electron. Eng., Tsinghua University [♣] Huawei Technologies Co., Ltd.
[♡] Dept. of Comp. Sci. & Tech., Institute for AI, Tsinghua University
yus21@mails.tsinghua.edu.cn; liuzy@tsinghua.edu.cn
{lixiangyang21, lizhonghua3, caozhao1}@huawei.com

Abstract

In this paper, we propose Retrieval-Augmented Retrieval (RAR), which augments dense retrieval model with history query-document pairs. We design a two-stage retrieval pipeline, where we first retrieve similar query-document pairs from the training data, and then concatenate current query with them to enhance the query representation. Results on BEIR zero-shot ranking benchmark show that RAR is able to consistently improve over the vanilla DR models.

1 Introduction

Dense retrieval (DR) is an important component in many NLP tasks, e.g. question answering and fact verification, and serves as the backbone for web search. DR is supported by an architecture called *dual encoders*, where the query and document are encoded separately with two identical pre-trained language models (PLMs) into vector representations in the semantic space (Karpukhin et al., 2020; Xiong et al., 2021). The documents in the collection can be encoded offline and the retrieval is conducted approximate nearest neighbour (ANN) search.

Though achieving state-of-the-art results in in-domain supervised datasets like Natural Questions (NQ) (Kwiatkowski et al., 2019) and MS MARCO (Bajaj et al., 2016), research has shown that DR may perform worse than traditional lexical retrieval models like BM25 in some out-of-domain setups, where training labels are not available (Thakur et al., 2021). We argue that the failure mostly roots in the information scarcity in the user queries – compared to long, self-contained documents, user queries are often short and concise, sometimes mentioning rare concepts, challenging PLMs to accurately encoding in the semantic space, especially for the out-of-domain, tail queries.

Different from previous research which directly encodes the bare query, in this paper, we propose **R**etrieval-**A**ugmented **R**etrieval (**RAR**), to augment current query with information from the query-document pairs in the training data. Specifically, we first build all available query-document pairs in the training data into a index. The index serves as an additional knowledge source for the query encoder. We then employ a two-stage dense retrieval process: 1) query-query retrieval, where we retrieve a similar or relating query-document pair from the index; 2) query-document retrieval, where we concatenate current query with the retrieved query-document pair to form a new query which is then used to retrieve relevant documents. Finally, we ensemble the retrieval results of the new query and the original bare query to obtain the final retrieval results. In real-world scenario, the query-document pair index can be search logs and user click data, which are abundant in commercial search engines.

We conduct experiments on a recent information retrieval benchmark BEIR (Thakur et al., 2021), which contains a series of retrieval tasks in a wide range of domains. We evaluate RAR's zero-shot performance on BEIR, with the training data from MS MARCO as the knowledge source. Results show that, without any in-domain supervision data, RAR significantly improves over the baseline model on most BEIR datasets. This proves that incorporating previous search history is helpful for PLMs to better understand user intent in the current query.

2 Related Work

Dense Retrieval With the advent of pre-trained language models, dense retrieval (DR) models are developed in recent years to serve as a substitute for traditional bag-of-words information retrieval models, which naturally suffer from the vocabulary mismatch problem (Karpukhin et al., 2020; Xiong et al., 2021; Lee et al., 2019). They use transformer-based PLMs as the backbone, mapping the query and document into low-dimensional, dense vectors in the semantic space. DR models are trained via contrastive learning, where the query and the relevant (positive) document are pulled together while the query and the irrelevant (negative) document are pushed away. They benefit from large-scale training data and achieve superior performance with supervised training. However, as the recently-proposed BEIR benchmark indicates, the zero-shot performance of DR models on out-of-distribution data are still poor, often lagging behind traditional lexical retrieval models.

Prompt-based Learning With the emergence of large PLMs like GPT-3 (Brown et al., 2020), researchers explored prepending additional texts into PLMs as "prompt" to help assist learning (Liu et al., 2021b). The prompts can be task descriptions, automatically-searched tokens, and/or several labeled instances. The method of prepending labeled instances is often referred to as "in-context learning" (Liu et al., 2021a). Inspired by the new learning schemes, in this work, we explore the effectiveness of using available search data as the source for in-context examples to form better query representations.

3 Methodology

3.1 Preliminaries

Given a query q and a document collection D, a retrieval model retrieves a set of documents $d_1, d_2, ..., d_n$ from D and sorts them according to their relevance to the query. A DR model calculates the relevancy according to the query and document representations:

$$\text{rel}(q,d) = f(g(q), g(d)), \tag{1}$$

where $g()$ is the encoder and $f()$ is the scoring function. The encoder is often supported by a PLM with pooling head to produce a vector representation. The scoring function is often as simple as dot product. During inference, a approximate nearest neighbor (ANN) search is executed to find relevant documents using the document embeddings encoded in advance.

3.2 Retrieval-Augmented Retrieval

Typical DR models find relevant documents directly using the bare query, which may not include sufficient information. We complement the query with relating (q,d) at hand – in the training data. We achieve this by first building a search index containing all relevant (q,d) pairs in the training data (irrelevant pairs are discarded). The index can either be a traditional bag-of-word index (BM25 index) or a dense index.

During inference, RAR employs a two-stage retrieval process. In the first stage, we query the (q,d) index to get a similar query for user query q, denoted as q'. The relevant document for q' is denoted as d'. Then we concatenate q, q', and d' using the following template:

$$\begin{aligned}&T(q,q',d')\\ &=\text{This query: } q \text{ Sample query: } q\,'\text{Sample document: } d.\end{aligned} \tag{2}$$

The query encoder is fed in with the new input format to perform retrieval:

$$\text{rel}(q,d) = f(g(T(q,q',d')), g(d)). \tag{3}$$

Note that the input for the document encoder remains the same.

During training, following previous work, we first fine-tune the model with BM25 negatives, and then with self-retrieved negatives. In the first training stage, we build a BM25 index for train (q,d) pairs. In the second training stage, the related (q,d) pairs are retrieved from a dense index, produced from a plain dense retrieval model trained using the same BM25 negatives.

The final retrieval results of RAR is further fused with a plain DR counterpart using Reciprocal Rank Fusion (Cormack et al., 2009):

$$\text{rel}(q,d) = \frac{1}{k + r_1(q,d)} + \frac{1}{k + r_2(q,d)}, \tag{4}$$

where $r_1(q,d)$, $r_2(q,d)$ are the document ranks (start from 1) of a plain DR and RAR model, respectively, and k is a parameter.

4 Results

The results are presented in Table 1. On most BEIR datasets, RAR successfully improves over the T5-base baseline. with reciprocal rank fusion, RAR w/ Fusion further outperforms RAR, achieving the highest performance in all datasets excluding Quora, a duplicate-question retrieval dataset. We conjecture that the downgraded performance on Quora may be because the original query already contains enough information for duplicate-question retrieval.

Dataset	T5-base		RAR			RAR w/ Fusion		
	NDCG@10	R@100	NDCG@10	R@100	W/T/L(%)	NDCG@10	R@100	W/T/L(%)
Bio-Medical IR								
TREC-COVID	0.5551	0.0797	0.5598	0.0859	44/6/50	**0.6006**†	**0.0901**†	60/2/38
NFCorpus	0.2646	0.2313	0.2694	0.2415	29/43/28	**0.2780**†	**0.2437**†	32/50/18
Question Answering								
NQ	0.4152	0.8003	0.3929	0.7895	22/51/27	**0.4298**†	**0.8293**†	25/59/16
FiQA-2018	0.2476	0.5359	0.2522	0.5597†	24/53/22	**0.2613**†	**0.5730**†	24/61/14
Argument Retrieval								
ArguAna	0.3081	0.9296	0.3188†	0.9296	30/41/29	**0.3250**†	**0.9502**†	28/52/20
Touche-2020	0.2866	**0.4490**	0.2562	0.4075	31/22/47	**0.2905**	0.4424	43/22/35
Duplicate-Question Retrieval								
Quora	**0.8066**	**0.9850**	0.6618	0.9572	10/49/41	0.7609	0.9845	13/63/25
Entity Retrieval								
DBPedia	0.2735	0.3314	0.2652	0.3397	34/26/40	**0.2970**†	**0.3594**†	46/30/24
Citation Prediction								
SCIDOCS	0.1100	0.2607	0.1112	0.2721†	21/59/20	**0.1208**†	**0.2818**†	23/64/13
Fact Checking								
Scifact	0.5029	0.8087	0.5221	0.8113	19/63/17	**0.5353**†	**0.8403**†	19/73/8
Average	0.3770	0.5412	0.3610	0.5464	n.a.	**0.3899**	**0.5595**	n.a.

Table 1: BEIR results. † indicates statistically significant improvements over T5-base.

5 Conclusion

In this paper, we propose Retrieval-Augmented Retrieval (RAR), which introduces a query-query retrieval stage to enhance the representation of the query in query-document retrieval. Experiments on BEIR show that RAR significantly improves over vanilla dense retrieval methods.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Processing of NeurIPS*.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP*, pages 6769–6781.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: A benchmark for question answering research. *Transactions of the ACL*, 7:452–466.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of ACL*, pages 6086–6096.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021a. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021b. Pretrain, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

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*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwikj. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *Proceedings of ICLR*.

A Example Appendix

This is an appendix.