

Reinforced IR: A Self-Boosting Framework For Domain-Adapted Information Retrieval

Chaofan Li^{1*}, Jianlyu Chen^{2,3*}, Yingxia Shao^{1*}, Chaozhuo Li¹, Quanqing Xu[†],
Defu Lian^{3‡}, Zheng Liu^{2,4‡}

¹Beijing University of Posts and Telecommunications, ²Beijing Academy of Artificial Intelligence

³University of Science and Technology of China, ⁴The Hong Kong Polytechnic University
{cfl, shaoyx}@bupt.edu.cn, liandefu@ustc.edu.cn, {zhengliu1026, xuquanqing}@gmail.com

Abstract

While retrieval techniques are widely used in practice, they still face significant challenges in cross-domain scenarios. Recently, generation-augmented methods have emerged as a promising solution to this problem. These methods enhance raw queries by incorporating additional information from an LLM-based generator, facilitating more direct retrieval of relevant documents. However, existing methods struggle with highly specialized situations that require extensive domain expertise. To address this problem, we present **Reinforced-IR**, a novel approach that jointly adapts a pre-trained retriever and generator for precise cross-domain retrieval. A key innovation of Reinforced-IR is its **Self-Boosting** framework, which enables retriever and generator to learn from each other's feedback. Specifically, the generator is reinforced to generate query augmentations that enhance the retriever's performance, while the retriever is trained to better discriminate the relevant documents identified by the generator. This iterative process allows the end-to-end retrieval performance to be progressively optimized using an unlabeled corpus from the target domain. In our experiment, Reinforced-IR outperforms existing domain adaptation methods by a large margin, leading to substantial improvements in retrieval quality across a wide range of application scenarios. We have publicly released our code at this [repo](#).

1 Introduction

With the rapid advancement of large language models (LLMs), AI copilots have become deeply integrated into a wide variety of activities, such as addressing knowledge-intensive problems, analyzing professional documents, developing computer programs, and providing personal assistance (Achiam et al., 2023; Team et al., 2023; Anthropic, 2024;

Gao et al., 2023; Zhou et al., 2024; Zhang et al., 2024b). To produce reliable and trustworthy results in these tasks, it is essential to incorporate useful knowledge from external databases, a process known as retrieval-augmented generation of LLMs, i.e., RAG (Lewis et al., 2020). Because of the advantages in broad applicability and simplicity, dense retrieval emerges as a popular form of retriever in such applications. It employs an embedder to map the data into a vector space, enabling the retrieval of relevant information based on vector similarity (Zhao et al., 2024; Wang et al., 2023a). Recently, numerous open-source models and API services have been made publicly available (Izacard et al., 2021; Xiao et al., 2024; Neelakantan et al., 2022; Li et al., 2024), which significantly facilitate the utilization of corresponding techniques.

Given the diverse range of applications, it's important to adapt general retrievers to new working scenarios beyond their original training domains, known as cross-domain retrieval. To this end, a variety of domain adaptation methods have been proposed in recent years. A notable breakthrough was made by the development of *HyDE*-style methods (hypothetical document embedding) (Gao et al., 2022; Wang et al., 2023c), or more broadly, the *GAR* techniques (generation-augmented retrieval) (Mao et al., 2020). These methods leverage LLMs, like ChatGPT, to enrich the query with extra information, thus enabling relevant documents to be identified in a straightforward way. *However, the existing methods mainly rely on LLMs trained on general domains, which may lack necessary knowledge needed by a highly specialized domain, such as medical or legal retrieval.* Besides, the heavy reliance on proprietary LLMs often results in prohibitively high costs, which limits their applicability in many situations.

To address these challenges, we propose a novel domain adaptation framework called **Reinforced-IR**, which jointly adapts the retriever and LLM-

*These people contribute equally to this paper.

†Quanqing Xu is an independent researcher.

‡Zheng Liu and Defu Lian are the corresponding authors.

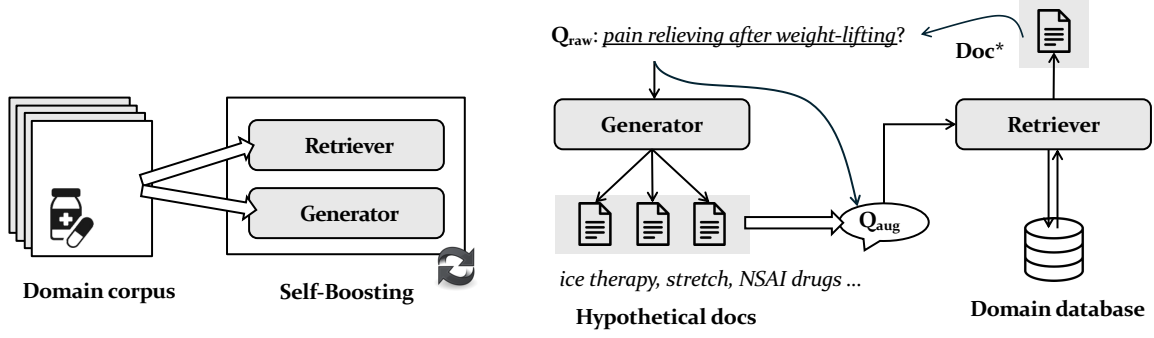


Figure 1: **Reinforced-IR** jointly adapts retriever and generator with an unlabeled domain corpus via self-boosting. The well-adapted generator augments raw query with hypothetical docs, which enables relevant docs to be retrieved.

based generator using a unlabeled corpus. Our method is distinguished for its design of **self-boosting** algorithm. It starts with a list of pseudo questions generated from the target domain’s unlabeled corpus. On one hand, the LLM-based generator is reinforced to perform high-quality query augmentation using the retriever’s feedback, such that relevant documents can be optimally retrieved for downstream tasks. This step is referred as the *Reinforcement Learning of generator with Retriever’s Feedback* (**RLRF**). On the other hand, the retriever is reinforced to discriminate the relevant documents preferred by the LLM-based generator. This step is called the *Reinforcement Learning of retriever with Generator’s Feedback* (**RLGF**). With the alternating execution of these two operations, the end-to-end retrieval performance can be progressively enhanced for the target domain.

We perform a comprehensive evaluation based on a variety of domain-specific datasets from BEIR (Thakur et al., 2021) and AIR-Bench (Chen et al., 2024b). We also include various retrievers and LLMs in our evaluation. According to the experiment result, Reinforced-IR substantially enhances the cross-domain performance of pre-trained retrievers and demonstrates notable advantages over the existing domain adaptation baselines. Additionally, the performance gains are especially pronounced on low-resource datasets that differ substantially from the original domains of the retrievers and LLMs, which further highlights the effectiveness of our approach for domain adaptation. Our model and source code will be shared with the public to advance the future research in this field.

In summary, the contributions of this paper are presented as follows:

- We introduce Reinforced-IR, a novel framework for cross-domain retrieval. To the best of our knowledge, this is the first work that jointly adapts retriever and generator to opti-

mize end-to-end retrieval performance.

- We design the **RLRF** and **RLGF** algorithms, enabling retriever and generator to mutually enhance each other’s performance based on an unlabeled corpus from the target domain.
- We conduct comprehensive experimental studies, which verify our significant advantage over existing cross-domain retrieval methods.

2 Method

In this section, we will first introduce the workflow of generation-augmented retrieval and formulate the problem. Then, we will elaborate the self-boosting algorithm, which optimizes the end-to-end retrieval performance using unlabeled data.

2.1 Generation-Augmented Retrieval

As a popular IR paradigm, dense retrieval identifies a query’s relevant documents based on embedding similarity. Given an embedding model $enc(\cdot)$, the query q and document d are transformed into latent vectors: $\mathbf{v}_q = enc(q)$, $\mathbf{v}_d = enc(d)$. On top of such results, the relevance score is calculated as the following inner product: $\sigma_{q,d} = \mathbf{v}_q^T \mathbf{v}_d$. It is expected that the most relevant document (d^*) can produce the highest relevant score compared to the rest of documents, i.e., $d^* : \max\{\mathbf{v}_q^T \mathbf{v}_d\}_{d \in D}$.

When applied to a new scenario, the model needs to handle different relevance patterns between query and document from its original domain. To bridge this gap, the query is augmented with extra information (Figure 1), like hypothetical docs in HyDE (Gao et al., 2022). Despite possible incomplete or inaccurate details, the generation-augmented retrieval (GAR) facilitates query and relevant docs to be matched in a more straightforward way. Nowadays, the query augmentation is often performed by a LLM-based generator $gen(\cdot)$,

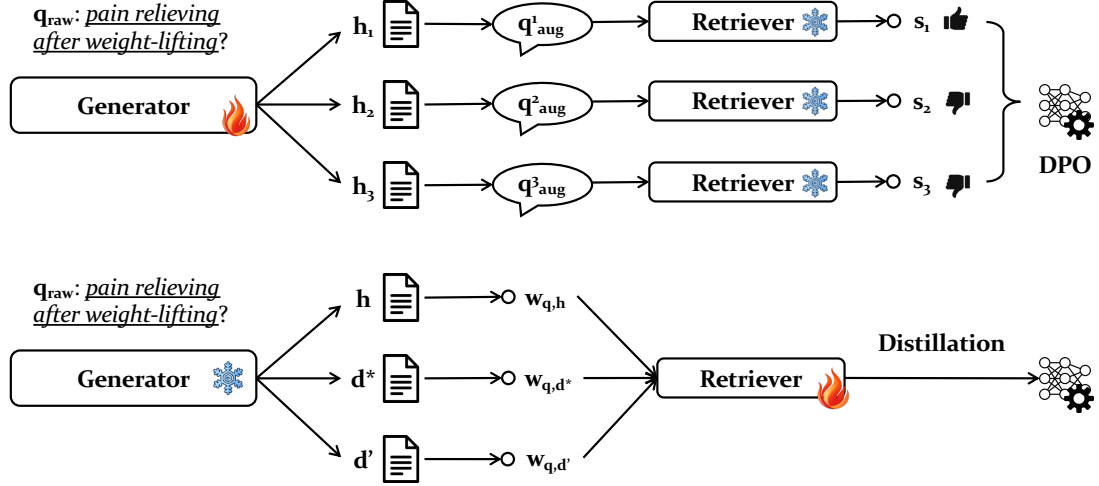


Figure 2: **Self-Boosting workflow.** 1) **RLRF**: the generator is reinforced to produce the retriever’s preferred query augmentation (marked by thumb-up) through DPO. 2) **RLGF**: the retriever is reinforced to discriminate the generator’s preferred documents (measured by preference score w_{\cdot}) in the form of knowledge distillation. Here, \mathbf{h} represents the hypothesis document, \mathbf{d}^* represents the positive sample, and \mathbf{d}' represents the negative sample.

which are directly prompted to generate a list of hypothetical docs (H_q) for the query:

$$H_q : \{h_i = \text{gen}(q, \text{prompt})\}_{i=1, \dots, L}. \quad (1)$$

Here, h_i is one of the sampled generation results. The system prompt is defined w.r.t. each concrete scenario, e.g., “{Query} (symptoms of some disease). Generate the treatment for the described disease” for a medical retrieval problem. Following the proposed workflow in HyDE, the augmented query embedding (\mathbf{v}'_q) is calculated as the linear combination of raw query embedding (\mathbf{v}_q) and each of the hypothetical document embeddings:

$$\mathbf{v}'_q = \alpha_0 * \mathbf{v}_q + \sum_{1 \dots L} \alpha_i * \mathbf{v}_{h_i}, \quad (2)$$

where $\mathbf{v}_{h_i} = \text{enc}(h_i)$, $\alpha_i > 0$, and $\sum_{0 \dots L} \alpha_i = 1$. Ultimately, the augmented query embedding \mathbf{v}'_q is used for the retrieval of relevant documents.

With the above definition, our problem is formulated as the joint optimization of embedding and generation model: $\text{enc}(\cdot)$, $\text{gen}(\cdot)$, such that the relevant documents in the target domain can be identified using the augmented query embedding, i.e., $d^* : \max\{\mathbf{v}'_q^T \mathbf{v}_d\}_{d \in D}$.

2.2 Self-Boosting

The optimization process begins with a unlabeled corpus (D) from the target domain. Following established practices (Ma et al., 2020; Thakur et al., 2021), we prompt the LLM to generate a set of synthetic queries: $Q \leftarrow \{q : Q\text{Gen}(d^*)\}_{D^*}$ for sampled documents D^* . The resulting pairs $\{(q, d^*)\}_Q$

serve the training source for domain adaptation. Building on this foundation, we introduce the **Self-Boosting** algorithm, which consists of two dual steps: generator optimization by *reinforcement learning from retriever’s feedback* (**RLRF**), and retriever optimization by *reinforcement learning from generator’s feedback* (**RLGF**). The two steps are iteratively performed, enabling progressive improvement in end-to-end retrieval performance.

2.2.1 Generator optimization by RLRF

The generator is prompted to produce a group of candidate hypothetical documents for each training query q : $H_q \leftarrow \{h_i = \text{gen}(q, \text{prompt})\}_{i=1, \dots, K}$, where K is the predefined sample size. Given this sampling result, the generator is reinforced to produce the best candidate which optimizes the retriever’s performance. Particularly, we simplify the calculation of augmented query embedding in Eq. 2 as the case with one single hypothetical document: $\mathbf{v}'_q = \alpha * \mathbf{v}_q + (1 - \alpha) * \mathbf{v}_{h_i}$, where we compute the retriever’s preference score as:

$$s_{q, h_i} = \mathbf{v}'_q^T \mathbf{v}_{d^*}. \quad (3)$$

By applying the above computation to every augmented query, we get $S_q \leftarrow \{s_{q, h_i}\}_{H_q}$ as the retriever’s feedback to the whole hypothetical documents. We further conduct direct preference optimization (DPO) to reinforce the generation of the retriever’s preferred query augmentation (Rafailov et al., 2024). For the simplicity of training, we only consider the hypothetical documents of the highest and lowest scores, and leverage them as the winning and losing candidates: h_w, h_l . To screen out

low-quality samples, we introduce the following filtering rules to the candidate documents:

$$1. s_{q,h_w} > s_q, \quad 2. s_{q,h_w} > \gamma * s_{q,h_l}, \quad (4)$$

where γ is a scaling factor: $\gamma > 1$, s_q indicates the preference score without using hypothetical document: $s_q = \mathbf{v}_q^T \mathbf{v}_{d^*}$. The first rule regularizes that the winning candidate must positively contribute to the retrieval result, while the second one guarantees the significance of winning candidate’s contribution. Finally, we apply the following loss for DPO:

$$\mathcal{L}^{dpo} = -\log \sigma \left(\beta \log \frac{\pi(h_w|q)}{\pi'(h_w|q)} - \beta \log \frac{\pi(h_l|q)}{\pi'(h_l|q)} \right), \quad (5)$$

where π and π' are the conditional likelihood from the adapted generator and the original generator respectively, and $\sigma(\cdot)$ is the sigmoid function.

2.2.2 Retriever Optimization by RLGF

The retriever needs to make effective use of the augmented query. To this end, we maximize the relevance score between \mathbf{v}'_q and \mathbf{v}_d : $\mathbf{v}'_q^T \mathbf{v}_d$. As \mathbf{v}'_q is a linear combination of multiple embeddings (Eq. 2), the following decomposition is made: $\alpha_0 * \mathbf{v}_q^T \mathbf{v}_d + \sum_L \alpha_i * \mathbf{v}_{h_i}^T \mathbf{v}_d$. However, optimizing this objective involves two different capabilities from the retriever: 1) *query-to-doc* matching as required by $\mathbf{v}_q^T \mathbf{v}_d$, 2) *doc-to-doc* matching as needed by $\mathbf{v}_{h_i}^T \mathbf{v}_d$. Thus, the direct optimization process is challenging, as it must realize two distinct goals simultaneously. To address this problem, we propose the *proximity objective* ($\rho_{q,d}$) as an alternative:

$$\rho_{q,d} = \alpha_0 * \mathbf{v}_q^T \mathbf{v}_d + \sum_L \alpha_i * \mathbf{v}_{h_i}^T \mathbf{v}_{h_i}. \quad (6)$$

The proximity objective maximizes the embedding similarity between the query and hypothetical documents, i.e., $\mathbf{v}_q^T \mathbf{v}_{h_i}$. Consequently, the retriever focuses solely on the *query-to-doc* matching capability, which makes it easier to optimize. In addition, the above objective leverages \mathbf{v}_q as an anchor where both \mathbf{v}_d and \mathbf{v}_{h_i} are moved close to it. Therefore, the similarity between \mathbf{v}_d and \mathbf{v}_{h_i} can also be improved from its optimization. Based on the above definition, we initially formulate the contrastive loss for retriever’s training:

$$\mathcal{L}^{ctr} = -\sum_{D_q^*} \log \frac{\exp(\mathbf{v}_q^T \mathbf{v}_d)}{\sum_{D'_q} \exp(\mathbf{v}_q^T \mathbf{v}_{d'})}, \quad (7)$$

where D_q^* is the entire collection of positive documents to q , including d^* and H_q , D'_q comprises d (d^* or H_q) and the negative documents to q .

Knowing that the LLM-based generator can provide precise assessment of relevance due to its inherent re-ranking capability (Sun et al., 2023), we leverage its feedback for fine-grained training of retriever. Particularly, we apply the following template \mathcal{T} : “*Query: {q}. Doc [1]: {d_1}, Doc [2]: {d_2}, ... Rank these documents based on their relevance to the query.*”, and obtain the generator’s ranking list “ $D_q = d_1, d_2, \dots, d_N$ ”. Based on this feedback, we define the following loss function:

$$\mathcal{L}^{dst} = -\frac{1}{|D_q|} \sum_{d_k=d_1}^{d_N} \log \frac{\exp(\mathbf{v}_q^T \mathbf{v}_{d_k})}{\sum_{D'_{q,k}} \exp(\mathbf{v}_q^T \mathbf{v}_{d'})}, \quad (8)$$

which follows a variational form of knowledge distillation. Here, $D'_{q,k} = d_k, \dots, d_N$ indicates d_k and the lower ranked documents of d_k . By minimizing the above loss, the retriever is reinforced to discriminate the documents preferred by the generator.

3 Experiment

The experiments are performed for the following research problems. **RQ 1.** Can Reinforced-IR effectively improve the cross-domain performance over the base retriever? **RQ 2.** Can Reinforced-IR outperform existing domain-adaptation methods? **RQ 3.** Whether Reinforced-IR is generally effective with different datasets and model options? **RQ 4.** Whether the proposed technical designs substantially contribute to the ultimate performance?

Following the settings in HyDE, We adopt Contriever (Izacard et al., 2021) as our default retriever. Because Contriever is a pre-trained model from unlabeled data, it provides an ideal option to analyze the domain-adaptation effect (Gao et al., 2022). We also consider Contriever-ft and RetroMAE (Xiao et al., 2022), which are fine-tuned from MSMARCO (Bajaj et al., 2016), as well as BGE M3 (Chen et al., 2024a), GTE (Li et al., 2023), Stella (Zhang et al., 2024a), which are fine-tuned from various labeled datasets. We leverage Llama-3-8B as our default generator, which is one of the strongest sub 10B LLMs at the time of this paper (Dubey et al., 2024). We perform extended analysis using both similarly sized LLMs, like Mistral-7B (Jiang et al., 2023a) and Qwen-2.5-7B (Hui et al., 2024), as well as larger and stronger models, including Qwen-2.5-72B, Llama-3-70B, and GPT-4o-mini.

Contriever and Llama-3-8B are set as the default combination of retriever and generator unless specific declaration.

	BEIR									AIR-Bench					
	FiQA	Scidocs	Fever	Arguana	Scifact	T-Covid	Touche	DBPedia	AVG	Law	News	Health	Finance	ArXiv	AVG
Contriever															
Contriever	24.5	14.9	68.2	37.9	64.9	27.3	16.7	29.2	35.4	13.2	36.2	34.3	36.2	23.0	28.6
HyDE	26.2	12.4	69.5	40.9	66.5	59.7	15.8	33.9	40.6	11.0	31.4	33.9	27.7	21.8	25.2
Doc2query	25.5	15.3	69.4	39.5	65.9	29.2	12.6	29.9	35.9	12.7	37.8	34.3	37.4	23.1	29.1
QGen	31.6	17.8	66.9	50.1	69.0	63.3	19.0	34.3	44.0	28.0	44.9	42.2	43.6	33.4	38.4
GPL	33.2	17.7	77.5	44.5	58.4	67.1	21.6	42.7	45.3	24.7	44.9	44.6	41.7	36.5	38.5
HyDE+QGen	34.5	17.0	70.9	49.9	68.2	69.3	21.2	38.1	46.1	21.9	40.9	31.0	36.0	29.0	31.7
HyDE+GPL	35.0	17.3	76.6	42.3	57.3	74.9	25.1	42.3	46.4	18.7	40.9	40.4	32.1	32.5	32.9
Reinforced-IR	36.8	19.2	81.3	52.6	70.9	78.6	31.1	47.5	52.3	28.4	47.6	45.3	46.1	38.4	41.2
BGE-M3															
BGE-M3	41.1	16.4	81.0	54.1	64.2	54.7	22.3	39.8	46.7	25.6	50.8	49.1	46.0	37.4	41.8
HyDE	39.2	16.9	75.8	53.2	67.2	71.7	19.6	42.5	48.3	20.7	45.6	44.1	42.7	31.6	36.9
Doc2query	37.7	16.8	74.8	56.0	64.3	39.4	14.0	39.5	42.8	23.8	48.4	44.7	45.7	37.8	40.1
QGen	41.8	18.3	80.3	65.6	67.8	70.1	22.1	41.4	50.9	32.2	50.7	45.3	48.3	38.2	42.9
GPL	43.2	18.7	79.1	65.5	65.3	74.6	24.1	41.5	51.5	27.5	50.6	51.0	45.6	37.7	42.5
HyDE+QGen	42.0	19.2	75.7	60.4	68.6	78.6	22.1	42.4	51.1	27.8	46.2	40.7	46.5	31.5	38.5
HyDE+GPL	40.8	19.4	75.6	58.0	67.4	78.3	23.4	42.4	50.7	20.5	44.9	44.7	41.3	30.5	36.4
Reinforced-IR	45.8	19.2	84.7	65.1	68.2	83.9	32.4	45.5	55.6	32.5	52.6	51.5	48.9	39.7	45.0

Table 1: Overall evaluation (nDCG@10 [%]) based on BEIR and AIR-Bench datasets.

We evaluate the experiment result with two dataset sources. The first one comprises eight low-resource datasets from BEIR (Thakur et al., 2021). These datasets have not been fine-tuned by any of the retrievers used in the experiments, making them suitable for assessing cross-domain retrieval performance (Gao et al., 2022). The second source includes five domain-specific datasets from Air-Bench development sets (Chen et al., 2024b). As these datasets were recently produced, they have not been used for training either the retrievers or the generators involved in the experiments.

3.1 Experiment Analysis

The experiment results are analyzed in comparison with two classes of baselines. The first class relies on generative augmentation, including HyDE (Gao et al., 2022), which augments the query with hypothetical documents, and Doc2query (Nogueira et al., 2019), which augments the document with pseudo queries. The second class leverages continual fine-tuning, including QGen (Ma et al., 2020; Thakur et al., 2021), which fine-tunes the retriever based on synthetic queries obtained from the target domain by contrastive learning, and GPL, which performs knowledge distillation for fine-grained training. For the sake of fair comparison, all methods (baselines and Reinforced-IR) are applied to the same set of synthetic queries and the same generator and retriever backbones.

3.1.1 Overall Evaluation

The overall evaluation is demonstrated in Table 1, where the following analysis is made.

- *Improvement over base retrievers.* Reinforced-IR substantially improves the base retrievers’ cross-domain retrieval performances across all datasets. This effect is particularly evident with Contriever, a pre-trained model from massive unlabeled data. Specifically, it enables the average performance to be improved from 35.4 to 52.3 on BEIR. Moreover, it achieves even larger improvements on AIR-Bench, with the average performance raised from 28.6 to 41.2. Although another base retriever, BGE M3, has been broadly fine-tuned with various question answering datasets, Reinforced-IR still contributes to its performance, increasing its average performance from 46.7 to 55.6 on BEIR, and from 41.8 to 45.0 on AIR-Bench, respectively. The improvement on BGE M3 is remarkable, considering that a broadly fine-tuned retriever has already gained a good command of necessary knowledge on the target domain, where traditional domain adaptation methods struggle to make further improvements (Gao et al., 2022).

- *Improvements over domain-adaptation baselines.* Reinforced-IR also demonstrates significant advantages over existing domain-adaptation methods. Notably, it outperforms both generative augmentation methods (HyDE, Doc2Query) and continual fine-tuning methods (QGen, GPL) individually, as well as the combination of the two methods (HyDE+QGen, HyDE+GPL). A closer analysis of the experimental results reveals that the baseline methods struggle to deliver consistent improvements across different datasets and base retrievers. For instance, while HyDE enhances Contriever’s performance on BEIR, it contributes little

	BEIR									AIR-Bench					
	FiQA	Scidocs	Fever	Argu.	Scifact	T-Covid	Touche	DBPedia	AVG	Law	News	Health	Fin.	ArXiv	AVG
Contriever	24.5	14.9	68.2	37.9	64.9	27.3	16.7	29.2	35.5	13.2	36.2	34.3	36.2	23.0	28.6
HyDE (Mist, Ctrv)	25.3	12.4	69.5	38.6	69.2	55.0	19.7	35.8	40.7	11.3	33.4	32.5	28.6	21.6	25.5
HyDE (Qwen, Ctrv)	24.6	13.2	63.1	39.0	68.2	52.1	17.8	31.6	38.7	11.0	32.2	33.8	29.2	22.8	25.8
HyDE (Llama, Ctrv)	26.2	12.4	69.5	40.9	66.5	59.7	15.8	33.9	40.6	11.0	31.4	33.9	27.7	21.8	25.2
Ours (Mist, Ctrv)	38.0	19.5	81.0	52.4	70.5	78.3	30.6	47.5	52.2	29.7	48.2	46.3	46.8	38.3	41.9
Ours (Qwen, Ctrv)	36.7	19.4	78.2	52.0	71.1	75.7	31.2	46.2	51.3	30.1	46.9	45.5	46.1	37.8	41.3
Ours (Llama, Ctrv)	36.8	19.2	81.3	52.6	70.9	78.6	31.1	47.5	52.3	28.4	47.6	45.3	46.1	38.4	41.2
RetroMAE	31.6	15.0	77.3	43.4	65.3	77.2	23.7	39.0	46.6	14.5	45.7	44.7	41.0	34.5	36.1
HyDE (Llama, Ret)	26.4	13.7	75.9	37.4	62.8	72.0	25.4	38.1	44.0	11.8	44.2	29.2	37.4	26.5	29.8
Ours (Llama, Ret)	37.5	17.9	86.2	56.2	70.0	82.6	32.7	46.2	53.7	29.7	52.0	46.1	47.3	38.2	42.6
Contriever-ft	32.9	16.5	75.8	44.6	67.7	59.6	20.4	41.3	44.9	13.3	46.3	45.3	43.0	32.8	36.1
HyDE (Llama, C-ft)	31.2	15.8	77.4	40.6	67.7	72.9	26.6	41.8	46.8	12.4	45.3	39.1	40.3	29.4	33.3
Ours (Llama, C-ft)	38.7	18.7	84.4	52.1	70.4	78.5	34.3	46.4	52.9	30.4	48.3	49.6	48.3	37.4	42.8
GTE-large	44.6	23.4	84.5	57.3	74.3	70.2	25.5	42.4	52.8	16.1	46.0	51.5	43.0	36.7	38.9
HyDE (Llama, gte)	43.5	23.6	81.0	53.9	75.4	75.5	22.5	44.8	52.5	13.6	44.5	47.6	40.5	34.3	36.1
Ours (Llama, gte)	46.1	23.2	84.1	66.8	73.8	84.8	31.7	47.3	57.2	30.3	52.8	56.4	48.3	42.7	46.1
Stella-base-en-v2	38.6	18.6	79.1	60.7	72.5	64.7	21.9	39.7	49.5	15.9	42.7	50.0	40.7	30.4	36.0
HyDE (Llama, stella)	37.8	21.2	71.5	55.1	73.6	80.4	25.3	42.2	50.9	13.3	43.3	46.4	39.2	30.8	34.6
Ours (Llama, stella)	43.7	22.3	84.2	64.7	74.1	84.2	30.1	45.4	56.1	27.3	49.4	53.2	47.1	37.8	43.0
BGE-M3	41.1	16.4	81.0	54.1	64.2	54.7	22.3	39.8	46.7	25.6	50.8	49.1	46.0	37.4	41.8
HyDE (Llama, M3)	39.2	16.9	75.8	53.2	67.2	71.7	19.6	42.5	48.3	20.7	45.6	44.1	42.7	31.6	36.9
Ours (Llama, M3)	45.8	19.2	84.7	65.1	68.2	83.9	32.4	45.5	55.6	32.5	52.6	51.5	48.9	39.7	45.0

Table 2: Extended evaluation based on additional generators and retrievers.

to BGE M3, as the latter has already undergone extensive fine-tuning. Besides, the benefits of more advanced fine-tuning operations, such as those employed in GPL, are significantly diminished when applied to BGE M3. Additionally, the native combination of HyDE and fine-tuning based methods yields minimal benefit, probably due to the discrepancy between the two strategies. These observations highlight the limitations of existing domain-adaptation methods, particularly regarding their applicability and significance. In contrast, Reinforced-IR addresses these challenges through its self-boosting mechanism, which effectively mitigates such issues and drives substantial progress.

3.2 Extended Evaluation

We conduct the extended experiments to explore Reinforced-IR’s effectiveness under more situations.

- *Analysis of different backbones.* We study the impact from using different generators and retrievers, as demonstrated in Table 2. Our experiment compares three LLMs of similar sizes, including Llama-3 8B (Llama), Qwen-2.5 7B (Qwen), and Mistral 7B (Mist). Besides, we also consider the following types of retrievers: 1) pre-trained model: Contriever, 2) models fine-tuned only with MS MARCO: Contriever-ft, RetroMAE, and 3) broadly fine-tuned models with various datasets: BGE M3, GTE (large), and Stella-v2. From these evaluations, we derive several key observations.

First, Reinforced-IR consistently outperforms

both the base retriever and HyDE baseline when working with different LLM backbones. Despite some underlying differences, e.g., Llama-3 and Mistral are pre-trained more comprehensively than Qwen-2.5, and Qwen-2.5 is more of a bi-lingual LLM compared to the other two models, all methods converge to a superior performance through Reinforced-IR. In contrast, none of the HyDE alternatives surpasses Contriever on AIR-Bench, underscoring the incapability of existing generative augmentation methods in dealing with new tasks.

Second, Reinforced-IR achieves a significant advantage over the baselines when applied to pre-trained and MS MARCO-finetuned retrievers. This result highlights Reinforced-IR’s effect in enhancing cross-domain retrieval performance. Moreover, Reinforced-IR also makes substantial contributions to the broadly finetuned retrievers, particularly on AIR-Bench datasets, which demonstrates its generally applicability across diverse application scenarios.

- *Analysis of larger LLMs.* We incorporate three powerful LLMs to the experiment (Table 5): Llama3-70B and Qwen2.5-72B, and GPT-4o-mini. This allows us to explore the optimal effect of the existing generative augmentation methods. Our evaluation includes both HyDE and its combinations with other approaches.

Our experimental results reveal that the use of powerful LLMs can enhance baseline performance in certain scenarios, such as Contriever and BGE

	BEIR									AIR-Bench					
	FiQA	Scidocs	Fever	Argu.	Scifact	T-Covid	Touche	DBPedia	AVG	Law	News	Health	Fin.	ArXiv	AVG
Contriever															
<i>HyDE</i>															
Llama3-70B	28.1	14.6	74.4	40.6	69.6	51.1	19.7	36.0	41.8	12.8	36.5	35.8	32.2	24.6	28.4
Qwen2.5-72B	25.5	14.1	77.7	46.6	70.1	56.8	18.3	35.1	43.0	11.6	31.2	32.3	28.3	23.2	25.3
GPT-4o-mini	26.1	13.2	76.7	44.4	68.2	57.3	18.8	33.4	42.3	12.0	34.1	34.6	31.7	24.0	27.3
<i>HyDE+QGen</i>															
Llama3-70B	35.9	18.3	75.9	51.1	72.8	69.8	24.1	40.5	48.6	24.0	44.0	37.9	40.0	32.4	35.7
Qwen2.5-72B	35.7	18.0	79.3	55.1	72.4	71.0	22.6	38.4	49.1	22.4	37.7	33.7	34.8	30.2	31.8
GPT-4o-mini	35.8	17.5	78.4	53.4	71.2	70.3	23.8	36.7	48.4	24.5	43.0	36.0	39.4	30.9	34.8
<i>HyDE+GPL</i>															
Llama3-70B	36.3	18.4	80.9	42.9	62.9	73.7	29.0	44.1	48.5	21.1	41.9	42.4	34.7	34.4	34.9
Qwen2.5-72B	35.9	18.5	83.2	47.1	63.9	73.0	25.6	41.3	48.6	19.7	35.8	38.3	31.2	32.8	31.6
GPT-4o-mini	35.2	17.5	83.2	44.9	63.3	73.7	27.6	40.9	48.3	20.9	41.1	41.6	34.7	34.2	34.5
Reinforced-IR	36.8	19.2	81.3	52.6	70.9	78.6	31.1	47.5	52.3	28.4	47.6	45.3	46.1	38.4	41.2
BGE-M3															
<i>HyDE</i>															
Llama3-70B	40.9	17.5	80.8	54.4	70.2	68.9	22.5	43.6	49.9	22.1	46.2	44.2	40.7	32.8	37.2
Qwen2.5-72B	40.6	16.9	83.9	55.1	71.2	68.6	20.8	42.9	50.0	20.9	39.5	41.9	36.2	30.4	33.8
GPT-4o-mini	40.3	16.9	83.9	52.7	69.5	72.4	21.7	42.8	50.0	22.9	44.7	44.8	41.1	31.9	37.1
<i>HyDE+QGen</i>															
Llama3-70B	43.8	19.8	79.6	62.9	71.9	75.6	26.4	43.5	52.9	28.0	45.8	42.2	44.4	33.3	38.7
Qwen2.5-72B	42.3	20.0	82.9	60.4	71.1	74.5	25.9	43.0	52.5	21.0	39.3	42.0	36.8	31.1	34.0
GPT-4o-mini	42.7	19.5	83.1	57.5	71.1	81.0	28.4	42.4	53.2	22.7	44.6	45.2	40.0	31.8	36.9
<i>HyDE+GPL</i>															
Llama3-70B	42.8	20.4	79.9	60.4	70.3	77.7	28.7	44.0	53.0	22.4	45.1	45.0	40.8	32.4	37.1
Qwen2.5-72B	39.2	20.0	75.6	60.3	63.8	69.0	25.4	42.5	49.5	21.3	39.7	41.6	36.0	31.2	34.0
GPT-4o-mini	40.2	19.6	74.3	57.2	68.1	74.3	27.2	42.6	50.4	22.6	44.9	44.7	40.3	32.0	36.9
Reinforced-IR	45.8	19.2	84.7	65.1	68.2	83.9	32.4	45.5	55.6	32.5	52.6	51.5	48.9	39.7	45.0

Table 3: Extended evaluation based on larger LLMs.

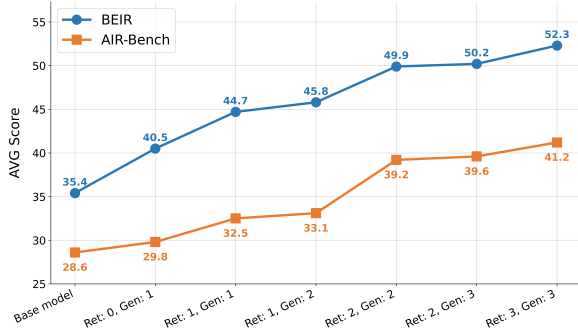


Figure 3: Impact from iterative optimization of generator (Gen) and retriever (Ret).

M3’s retrieval performance on BEIR. However, these improvements are inconsistent across different datasets. Besides, there remains a large performance gap between these methods and Reinforced-IR in most cases. These results highlight that it’s not enough to simply count on the increased capacity of LLMs. Instead, they underscore the necessity of jointly adapting LLMs and retrievers to optimize the cross-domain retrieval performance.

- *Analysis of self-boosting.* We evaluate the impact of self-boosting by analyzing Reinforced-IR’s performance growth throughout the training process (Figure 3, as detailed in Appendix D). Specifically, the complete set of training queries is divided into three subsets. For each subset, we conduct

one self-boosting iteration, consisting of a round of generator optimization via RLRF (Gen- i), followed by a round of retriever optimization through RLGF (Ret- i).

The experimental results demonstrate that both self-boosting operations contribute substantially to the improvement of retrieval performance. In each iteration, the optimization of the generator enables the production of more effective query augmentations for the current retriever, improving the performance from “Ret- i , Gen- i ” to “Ret- i , Gen- $(i+1)$ ”. While the optimization of retriever allows it to make better use of the augmented queries from the current generator, which further improves the performance from “Ret- i , Gen- $(i+1)$ ” to “Ret- $(i+1)$, Gen- $(i+1)$ ”. This iterative refinement ultimately results in Reinforced-IR’s superior performance across the entire training dataset.

3.2.1 Ablation Study

We make detailed analysis for Reinforced-IR’s technical factors with the ablation study in Table 4.

- *Training methods.* In our experiment, we replace the original DPO with supervised fine-tuning for the generator’s training, using the winning candidate as the supervision label (w/o DPO). Additionally, we substituted basic contrastive learning

	BEIR	Δ	AIR-Bench	Δ
Reinforced-IR (default)	52.3	-	41.2	-
w/o DPO	50.6	-1.7	40.6	-0.6
w/o Distillation	49.3	-3.0	38.9	-2.3
w/o Proximity	52.3	-0.0	40.2	-1.0
w/o Filtering rule-1	49.0	-3.3	41.1	-0.1
w/o Filtering rule-2	49.7	-2.6	40.9	-0.3

Table 4: Ablation studies.

for knowledge distillation during retriever’s training (w/o Distillation). The experiment result shows that both modifications lead to significant decline of empirical performance on the two evaluation benchmarks. This decline is attributed to the alternative methods’ inability to incorporate the fine-grained feedback from the generator and retriever, specifically, the usability of augmented queries and document relevance, thereby hindering the effective utilization of training data.

- *Impact of proximity objective.* We further replace the proximity objective in Eq. 6 with the basic objective: $\alpha_0 * \mathbf{v}_q^T \mathbf{v}_d + \sum_L \alpha_i * \mathbf{v}_{h_i}^T \mathbf{v}_d$. As discussed, this alternative form requires the model to accomplish both query-to-doc and doc-to-doc matching, thus increasing the training difficulty. Our experiment result verifies proximity objective’s overall effectiveness in general scenarios, as the alternative method (w/o Proximity) significantly reduces the performance on AIR-Bench, which solely comprises question-answering style tasks, while leading to a neural impact on BEIR, which constitutes miscellaneous tasks.

- *Candidate filtering.* We disable the filtering rules in Eq. 3 and make direct use of the unfiltered candidates (w/o Filtering rule-1, w/o Filtering rule-2). The experiment result highlights the significance of both rules on BEIR’s performance. This can be attributed to BEIR’s diverse retrieval tasks, which increases the likelihood of generating unsuitable query augmentation from the generator. As such, the filtering operations are essential to optimizing the performance. In contrast, AIR-Bench focuses solely on question-answering tasks, allowing for more reliable query augmentation and diminishing the need for candidate filtering.

4 Related Work

4.1 Cross-domain Retrieval

Cross-domain retrieval is an important but challenging problem for existing techniques. As demonstrated by the popular benchmarks in this field (Thakur et al., 2021), the pre-trained retrievers are

prone to inferior performances when they are applied directly for a new working scenario. To tackle this challenge, one common strategy is to perform multi-task training, where a pre-trained retriever is broadly fine-tuned using extensive labeled datasets. By learning from diverse tasks during training, the retriever develops a stronger ability to handle new tasks during testing (Wang et al., 2022; Xiao et al., 2024; Su et al., 2022). However, multi-task retrievers often make trade-offs between individual tasks to optimize overall performance, leading to significant performance gaps when compared to specialized retrievers in target domains.

Another line of research focuses on the continual fine-tuning of pre-trained retrievers using synthetic data generated from a target domain (Ma et al., 2020; Thakur et al., 2021; Wang et al., 2021). These approaches leverage generators to produce synthetic queries for unlabeled documents, which creates training samples to fine-tune the pre-trained models. Thanks to the popularity of language models, it’s made possible to produce synthetic queries at scale, (Bonifacio et al., 2022; Jeronymo et al., 2023; Wang et al., 2023b), enabling the corresponding approaches to be easily conducted in practice. However, these approaches could deliver limited performance gains due to the potential mismatch between the synthetic data and actual scenarios.

Different from the fine-tuning methods, generation augmented retrieval (GAR) makes direct use of generation models to address the cross-domain problems (Mao et al., 2020). These methods enrich query and document with extra information, enabling relevant data to identified in a straightforward way. Nowadays, large language models are widely adopted as the backbone generator (Gao et al., 2022; Wang et al., 2023c), which contributes to the performance and applicability of corresponding methods. Although GAR is widely perceived as a promising strategy, it’s not enough solely rely on general LLMs, as they still lack necessary knowledge required to generate effective query augmentations for highly specialized problems.

4.2 Joint Optimization

For complex tasks, multiple models often need to work together. For example, Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) first employs a retriever to search for relevant documents from a large-scale knowledge base based on the input query, then utilizes a generator to produce an-

swers based on the retrieved documents. Similarly, reranking (Liu et al., 2025) involves using a computationally efficient bi-encoder model for initial retrieval, followed by a more precise cross-encoder model to rerank candidate documents. In these tasks, different models serve distinct sub-functions, and optimizing only one while neglecting the collaborative of them often fails to achieve comprehensive improvements. Consequently, several studies propose joint optimization strategies aimed at increasing overall efficiency and performance (Ke et al., 2024; Liu et al., 2021; Jiang et al., 2023b; Miech et al., 2021).

For GAR tasks, two primary components are introduced: a retriever and a generator. Recent works tend to fix their capabilities or optimize only the retriever (Gao et al., 2022), where the generator cannot learn new knowledge to generate effective augmented queries, and decoupled training of retrievers and generators overlooks their potential synergistic effects, which ultimately constrains overall performance.

In this paper, we propose joint optimization for the retriever and the generator, allowing them to work collaboratively to efficiently accomplish the retrieval task.

5 Conclusion

In this paper, we introduce Reinforced-IR, a novel self-boosting framework for cross-domain retrieval. Our method employs two advanced learning algorithms: RLRF and RLGF. These algorithms enable the generator and retriever to mutually reinforce each other through feedback, leading to a progressive enhancement of retrieval performance. The effectiveness of Reinforced-IR is thoroughly validated, as it outperforms existing domain adaptation methods by a huge advantage, delivering superior performance across various application scenarios.

Limitations

Although Reinforced-IR has demonstrated significant improvements, there are two limitations in this work. Firstly, the current training data generation process employs a relatively simplistic approach. Future research could optimize the training data generation process, which may yield further improvements in retrieval performance. Secondly, we utilize 7B or 8B parameter models as generators. For computational efficiency considerations, smaller models (1B-2B parameters) could be ex-

plored in future research.

Acknowledgements

This work is supported by the National Science and Technology Major Project (2022ZD0116315), National Natural Science Foundation of China (Nos. 62272054, 62192784, U24A20253), Beijing Nova Program (No. 20230484319), and Xiaomi Young Talents Program.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Anthropic. 2024. *The Claude 3 Model Family: Opus, Sonnet, Haiku*.
- 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*.
- Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2387–2392.
- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.
- Jianlv Chen, Nan Wang, Chaofan Li, Bo Wang, Shitao Xiao, Han Xiao, Hao Liao, Defu Lian, and Zheng Liu. 2024b. Air-bench: Automated heterogeneous information retrieval benchmark. *arXiv preprint arXiv:2412.13102*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*.
- Yuan Gao, Xiang Wang, Xiangnan He, Huamin Feng, and Yongdong Zhang. 2023. *Rumor detection with self-supervised learning on texts and social graph*. *Frontiers of Computer Science*, 17(4):174611.

- Binyuan Hui, Jian Yang, Zeyu Cui, Jiayi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.
- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.
- Vitor Jeronimo, Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, Jakub Zavrel, and Rodrigo Nogueira. 2023. Inpars-v2: Large language models as efficient dataset generators for information retrieval. *arXiv preprint arXiv:2301.01820*.
- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.
- Fan Jiang, Qionghai Xu, Tom Drummond, and Trevor Cohn. 2023b. Boot and switch: Alternating distillation for zero-shot dense retrieval. *arXiv preprint arXiv:2311.15564*.
- Zixuan Ke, Weize Kong, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Bridging the preference gap between retrievers and llms. *arXiv preprint arXiv:2401.06954*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich K ttler, Mike Lewis, Wen-tau Yih, Tim Rockt schel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.
- Chaofan Li, Minghao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Yingxia Shao, Defu Lian, and Zheng Liu. 2024. Making text embedders few-shot learners. *arXiv preprint arXiv:2409.15700*.
- Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.
- Fangyu Liu, Yunlong Jiao, Jordan Massiah, Emine Yilmaz, and Serhii Havrylov. 2021. Trans-encoder: Unsupervised sentence-pair modelling through self-and mutual-distillations. *arXiv preprint arXiv:2109.13059*.
- Zheng Liu, Chaofan Li, Shitao Xiao, Chaozhuo Li, Chen Jason Zhang, Hao Liao, Defu Lian, and Yingxia Shao. 2025. Fitting into any shape: A flexible llm-based re-ranker with configurable depth and width. In *Proceedings of the ACM on Web Conference 2025*, pages 3942–3951.
- Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. 2020. Zero-shot neural passage retrieval via domain-targeted synthetic question generation. *arXiv preprint arXiv:2004.14503*.
- Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2020. Generation-augmented retrieval for open-domain question answering. *arXiv preprint arXiv:2009.08553*.
- Antoine Miech, Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Andrew Zisserman. 2021. Thinking fast and slow: Efficient text-to-visual retrieval with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9826–9836.
- Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. 2022. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*.
- Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.
- Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, and Tao Yu. 2022. One embedder, any task: Instruction-finetuned text embeddings. *arXiv preprint arXiv:2212.09741*.
- Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agents. *arXiv preprint arXiv:2304.09542*.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Nandan Thakur, Nils Reimers, Andreas R ckl , Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.
- Jiajia Wang, Weizhong Zhao, Xinhui Tu, and Tingting He. 2023a. [A novel dense retrieval framework for long document retrieval](#). *Frontiers of Computer Science*, 17(4):174609.
- 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*.

- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.
- Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023b. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.
- Liang Wang, Nan Yang, and Furu Wei. 2023c. Query2doc: Query expansion with large language models. *arXiv preprint arXiv:2303.07678*.
- Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. 2022. [Retromae: Pre-training retrieval-oriented language models via masked auto-encoder](#).
- Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muenighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- Dun Zhang et al. 2024a. Jasper and stella: distillation of sota embedding models. *arXiv preprint arXiv:2412.19048*.
- Zhigao Zhang, Fanfei Song, Bin Wang, and Chuan-sheng Dong. 2024b. Extract implicit semantic friends and their influences from bipartite network for social recommendation. *Data Science and Engineering*, 9(3):278–293.
- Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4):1–60.
- Xuanhe Zhou, Zhaoyan Sun, and Guoliang Li. 2024. Db-gpt: Large language model meets database. *Data Science and Engineering*, 9(1):102–111.

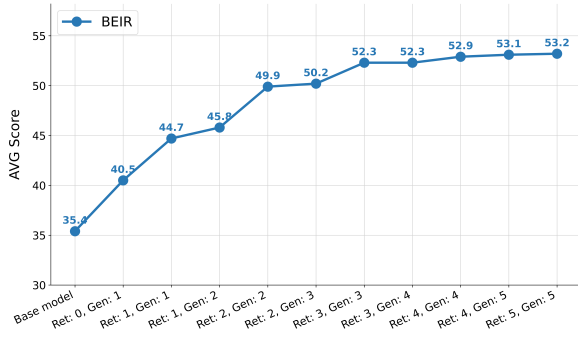


Figure 4: Impact of iterative optimization between the generator (Gen) and retriever (Ret) over multiple iterations.

A Implementation Details

When get query embeddings, we set α to 0.8. In RLRF, γ is sequentially set to 1.05, 1.08, and 1.1 over three iterations. Meanwhile, in RLGF, both α_0 and α_i are defined as 0.5.

For LLM, we use the following version:

- 1) Meta-Llama-3-8B-Instruct,
- 2) Qwen-2.5-7B-Instruct,
- 3) Mistral-7B-Instruct-v0.3,
- 4) Meta-Llama-3-70B-Instruct,
- 5) Qwen-2.5-72B-Instruct,
- 6) gpt-4o-mini-2024-07-18

B Query Generation

To enhance the effectiveness of query generation, we have implemented refinements to QGen, which encompasses two distinct phases: query generation and quality control. During the query generation phase, we provide the LLM with relevant documents and tasks, prompting it to produce corresponding queries. In the quality control phase, the tasks, documents, and generated queries are presented to the LLM again, enabling it to evaluate whether the generated queries are relevant to the documents. Queries deemed irrelevant are then filtered out.

Taking Trec-covid dataset as an example, for query generation, we use the following prompt:

Here is a retrieval task (Task) and a document (Passage):

Task: Given a query on COVID-19, retrieve documents that answer the query.

Passage: {passage}

Given the retrieval task and the document, your mission is:
- Generate a query on COVID-19 that the document can answer.

Note:

- The generated query should not contain the pronouns such as "this", "that", "it", "there", "here", etc.
- The generated query should be clear and 5 to 10 words.
- The generated query should be common and formal in terms of language style.

Your output should be a string of the generated query. Remember do not explain your output.

Your output:

For quality control, we use the following prompt:

Given a retrieval task (Task), a query (Query), and a document (Passage), your mission is Judge whether the document can answer the query..

Task: Given a query on COVID-19, retrieve documents that answer the query.

Query: {query}
Passage: {passage}

Your output must be one of the following:

- 0: No, the document cannot answer the query.
- 1: Yes, the document can answer the query.

Do not explain your answer in the output. Your output must be a single number.

Your output:

C Hypothetical Document

For hypothetical document, we use the following prompt:

Given a retrieval task and a query, your mission is to generate a brief document for the query in the context of the retrieval task.

Task: Given a query on COVID-19, retrieve documents that answer the query.

Query: {query}

Your output:

D Impact of Iterative optimization

Table 5 presents the detailed results on the BEIR and AIR-Bench benchmarks. With each iteration of the self-boosting process, the performance on every dataset improves significantly.

We conduct an additional 4-5 iterations of experiments on the BEIR benchmark using Contriever and Llama-3-8B as base models, as shown in Figure 4. The experimental results indicate that performance continues to improve as training extends beyond three iterations, which is consistent with our previous observations. However, the additional iterations yield only marginal gains, considering resource limitation, three iterations is enough.

	BEIR									AIR-Bench					
	FiQA	Scidocs	Fever	Argu.	Scifact	T-Covid	Touche	DBPedia	AVG	Law	News	Health	Fin.	ArXiv	AVG
Base model	24.5	14.9	68.2	37.9	64.9	27.3	16.7	29.2	35.4	13.2	36.2	34.3	36.2	23.0	28.6
Ret: 0, Gen: 1	28.9	15.6	74.0	40.3	67.1	42.9	20.3	34.9	40.5	13.6	39.3	36.8	34.2	24.9	29.8
Ret: 1, Gen: 1	29.8	17.6	72.2	43.2	69.8	64.6	25.3	35.1	44.7	18.1	36.6	38.4	40.7	28.7	32.5
Ret: 1, Gen: 2	30.8	17.7	74.9	42.6	69.9	69.4	25.2	35.7	45.8	18.5	38.0	38.3	41.2	29.6	33.1
Ret: 2, Gen: 2	33.5	18.8	78.2	49.1	70.4	74.8	30.4	43.9	49.9	25.6	45.0	43.3	44.8	37.1	39.2
Ret: 2, Gen: 3	33.6	19.1	79.1	49.7	70.5	73.6	31.0	45.2	50.2	26.2	45.8	43.6	44.8	37.4	39.6
Ret: 3, Gen: 3	36.8	19.2	81.3	52.6	70.9	78.8	31.1	47.5	52.3	28.4	47.6	45.3	46.1	38.4	41.2

Table 5: Detailed impact from iterative optimization of generator (Gen) and retriever (Ret).

	FiQA	Scidocs	Fever	Arguana	Scifact	T-Covid	Touche	DBPedia	AVG
QGen	31.6	17.8	66.9	50.1	69.0	63.3	19.0	34.3	44.0
GPL	33.2	17.7	77.5	44.5	58.4	67.1	21.6	42.7	45.3
QGen + HyDE (w/o doc ft)	34.5	17.0	70.9	49.9	68.2	69.3	21.2	38.1	46.1
GPL + HyDE (w/o doc ft)	35.0	17.3	76.6	42.3	57.3	74.9	25.1	42.3	46.4
QGen + HyDE (w/ doc ft)	35.7	19.8	69.1	56.7	70.0	63.2	17.9	39.6	46.5
GPL + HyDE (w/ doc ft)	36.1	19.7	74.6	50.6	65.1	69.1	20.6	42.8	47.3
Reinforced-IR	36.8	19.2	81.3	52.6	70.9	78.6	31.1	47.5	52.3

Table 6: Performance of optimized baselines on the BEIR benchmark.

Generator	Gen: 1	Gen: 2	Gen: 3
BEIR	40.5	41.4	42.1

Table 7: The performance on the BEIR benchmark while keeping the retriever fixed.

Generator	Gen: 1	Gen: 2	Gen: 3
Length	205	270	290

Table 8: The average lengths of augmented queries on the BEIR benchmark.

reflects enhanced augmented query quality. The average lengths of these augmented queries also grow consistently, as demonstrated in Table 8. Upon examining the detailed content, we find that the updated augmented queries provide more useful context to the original queries.

E Optimizing Baselines with Hyde

We optimize baselines by incorporating hypothetical documents during training for GPL and QGen methods on the BEIR benchmark. Table 6 presents results using Contriever and Llama-3-8B as base models, with "doc ft" indicating hypothetical document (augmented query) fine-tuning. As observed, integrating hypothetical documents during training for both the GPL and QGen methods leads to improvements. However, the overall impact on performance remains limited. This reinforces the effectiveness of our joint training approach, underscoring its advantages over these alternative methods.

F Effectiveness of Augmented Query

We investigate the effectiveness of augmented queries from different generators by maintaining a fixed retriever (Contriever) while updating the generator (Llama-3-8B). Table 7 shows steady performance increases on the BEIR benchmark, which