

Distillation Enhanced Generative Retrieval

Yongqi Li¹, Zhen Zhang², Wenjie Wang^{2*}, Liqiang Nie³, Wenjie Li^{1*}, Tat-Seng Chua²

¹The Hong Kong Polytechnic University

²National University of Singapore ³Harbin Institute of Technology (Shenzhen)

{liyongqi0,cristinzhang7,wenjiewang96,nieliqiang}@gmail.com
cswjli@comp.polyu.edu.hk dcscts@nus.edu.sg

Abstract

Generative retrieval is a promising new paradigm in text retrieval that *generates* identifier strings of relevant passages as the retrieval target. This paradigm leverages powerful generative language models, distinct from traditional sparse or dense retrieval methods. In this work, we identify a viable direction to further enhance generative retrieval via distillation and propose a feasible framework, named DGR. DGR utilizes sophisticated ranking models, such as the cross-encoder, in a teacher role to supply a passage rank list, which captures the varying relevance degrees of passages instead of binary hard labels; subsequently, DGR employs a specially designed *distilled RankNet* loss to optimize the generative retrieval model, considering the passage rank order provided by the teacher model as labels. This framework only requires an additional distillation step to enhance current generative retrieval systems and does not add any burden to the inference stage. We conduct experiments on four public datasets, and the results indicate that DGR achieves state-of-the-art performance among the generative retrieval methods. Additionally, DGR demonstrates exceptional robustness and generalizability with various teacher models and distillation losses. The code is released at <https://github.com/liyongqi67/DGR>.

1 Introduction

Text retrieval is a crucial task in information retrieval and has a significant impact on various language systems, including search ranking (Nogueira and Cho, 2019), open-domain question answering (Chen et al., 2017), and retrieval augmented generation (RAG) (Lewis et al., 2020). In recent years, dense retrieval (Lee et al., 2019; Karpukhin et al., 2020) has been the dominant approach for text retrieval based on the advancements in encoder-based language models, like BERT (Kenton and Toutanova, 2019).

With the advancement of generative large language models (Brown et al., 2020), generative retrieval emerges as an alternative paradigm to dense retrieval. Generative retrieval leverages autoregressive language models to *generate* identifier strings of target passages, such as Wikipedia page titles, to complete the retrieval process. Current approaches focus on exploring various identifiers to better represent passages. Initially, generative retrieval utilized page titles (De Cao et al., 2020) as identifiers but was limited to specific retrieval domains, such as Wikipedia. Subsequently, a range of identifier types were introduced, including numeric IDs (Tay et al., 2022), substrings (Bevilacqua et al., 2022), codebooks (Sun et al., 2023), and multiview identifiers (Li et al., 2023c), to consistently enhance generative retrieval for broader search scenarios and larger retrieval corpus.

Despite its rapid development and substantial potential, generative retrieval still has limitations. Generative retrieval relies on query-passage pairs for training, but the relevance judgments between queries and passages are typically incomplete. On the one hand, for a given query, only a few passages (or even just one) are judged, while the judgments of other passages are missing; on the other hand, the judgments typically provide binary labels, which often neglect the reality that different passages typically exhibit varying levels of relevance. During generative training, generative retrieval treats the target passage (identifier) as positive and all other passages (identifiers) as equally negative. This introduces substantial noise and disrupts the training of generative retrieval, exacerbating the limitations of incomplete judgments.

Tackling the aforementioned issues is challenging, as they are intrinsically inherent to the generative training of generative retrieval. In this work, we propose to tackle the above problem by innovating the training paradigm via knowledge distillation. We aim to employ a powerful rank

*Corresponding authors.

(teacher) model to give multi-level judgments of query-passage pairs for the generative retrieval (student) model. While previous studies have explored distillation for dense retrieval, its direct application to generative retrieval is not feasible due to significant differences in their training paradigms. To successfully introduce knowledge distillation into generative retrieval, two fundamental issues must be redefined and emphasized: what type of knowledge should be distilled from the teacher model, and how to effectively leverage the knowledge to benefit the student model?

In pursuit of this goal, we introduce a Distillation-enhanced Generative Retrieval framework, dubbed DGR, as illustrated in Figure 1. DGR characterizes knowledge as the rank order of passages provided by the teacher model and specially designs the distilled RankNet loss to efficiently incorporate this knowledge into the generative retrieval model. Specifically, we first train a typical generative retrieval model to retrieve passages. Subsequently, we utilize a sophisticated teacher model that excels in ranking passages better than generative retrieval, such as a cross-encoder, to rerank the retrieved passages. The rank order given by the teacher model contains fine-grained supervisor signals, which reflect the varying relevance degrees rather than simple binary labels. The custom-designed distilled RankNet is then employed to optimize the generative retrieval model based on the rank order provided by the teacher. During inference, we use the trained model to retrieve passages as in the typical generative retrieval. Therefore, the DGR framework only requires an additional distillation step and does not add any burden to the inference stage. We evaluate our proposed method on four widely used datasets, and the results demonstrate that DGR achieves the best performance in generative retrieval.

The key contributions are summarized:

- We are the first to introduce distillation into generative retrieval, identifying a viable direction and opening doors for potential advancements in generative retrieval.
- We propose the DGR framework, which formulates the distilled RankNet loss to effectively incorporate knowledge from advanced rank models into generative retrieval models.
- DGR achieves state-of-the-art performance in generative retrieval on four widely used

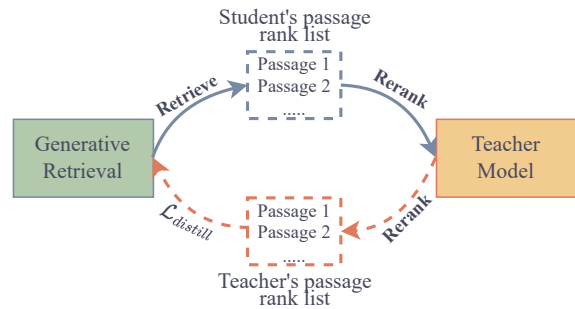


Figure 1: The illustration of distillation enhanced generative retrieval (DGR) framework. Sophisticated ranking models serve as teacher models to rerank the passages, and the custom-designed distilled RankNet loss is utilized to optimize the generative retrieval model.

datasets without any burden on the inference stage. DGR demonstrates exceptional robustness and generalizability with various teacher models and distillation losses.

2 Related Work

2.1 Generative Retrieval

Generative retrieval is an emerging new retrieval paradigm, which generates identifier strings of passages as the retrieval target. Instead of generating entire passages, generative retrieval uses identifiers to represent passages for reducing the amount of useless information (Li et al., 2024). Previous approaches have explored different types of identifiers for different search scenarios, including titles (URLs) (De Cao et al., 2020; Li et al., 2023a), numeric IDs (Tay et al., 2022), substrings (Bevilacqua et al., 2022), codebook (Sun et al., 2023; Zeng et al., 2024), and multiview identifiers (Li et al., 2023c). Despite the above different identifiers, generative retrieval still lags behind advanced dense retrievers. In this work, we aim to introduce knowledge distillation to further enhance generative retrieval.

2.2 Knowledge Distillation in Text Retrieval

Knowledge distillation is proposed by Hinton et al. to compress the knowledge in a powerful model into another one. In the text retrieval task, researchers also explored distilling knowledge from cross-encoder rankers into dense retrievers to enhance performance (Hofstätter et al., 2020, 2021; Qu et al., 2021; Zeng et al., 2022), because cross-encoder rankers are more powerful in ranking passages via fine-grained interactions. However, it is challenging to distill knowledge from cross-encoder rankers into generative retrieval, due to

their different training paradigms and architectures. In this work, we propose a feasible framework to facilitate knowledge distillation in generative retrieval.

2.3 Dense Retrieval

Dense retrieval (Lee et al., 2019; Karpukhin et al., 2020) is currently the de facto implementation of text retrieval. This method benefits from the powerful representation abilities of encoder-based language models and the MIPS algorithm (Shrivastava and Li, 2014), allowing for efficient passage retrieval from a large-scale corpus. Dense retrieval has been constantly developed through knowledge distillation, hard negative sampling, and better pre-training design (Chang et al., 2019; Wang et al., 2022a). Compared to dense retrieval, which relies on the dual-encoder architecture, the arising generative retrieval shows promise in overcoming the missing fine-grained interaction problem through the encoder-decoder paradigm. Despite the huge potential, generative retrieval still lags behind the state-of-the-art dense retrieval method and leaves much room for investigation.

3 Method

In this section, we begin by giving a preliminary of a standard generative retrieval system. Following this, we outline our distillation framework as it applies to the generative retrieval system.

3.1 Preliminary: Generative Retrieval

We present MINDER (Li et al., 2023c), an advanced generative retrieval system, as the base model to demonstrate the generative retrieval scheme. MINDER leverages multiview identifiers, enabling robust performance across diverse search scenarios and achieving advanced performance.

Identifiers. MINDER employs three types of identifiers to represent a passage: title, substring, and pseudo-query. The title generally reflects the main topic of the passage, while the substring is a randomly selected excerpt. The pseudo-query is generated based on the passage’s content. These identifier types complement each other and adapt to different scenarios, contributing to MINDER’s robustness across various search contexts.

Training. MINDER optimizes an autoregressive language model, denoted as **AM**, via typical sequence-to-sequence loss. The input text is the query text, and the output is the identifier of the

corresponding passage that is relevant to the query. During training, the three identifiers of the samples are randomly shuffled to train the autoregressive model. For each training sample, the objective is to minimize the sum of the negative loglikelihoods of the tokens $\{i_1, \dots, i_l\}$ in a target identifier I , whose length is l . The generation loss is formulated as,

$$\mathcal{L}_{gen} = - \sum_{j=1}^l \log p_{\theta}(i_j|q; I_{<j}), \quad (1)$$

where $I_{<j}$ denotes the partial identifier sequence $\{i_0, \dots, i_{j-1}\}$, q is the question text, and θ is the trainable parameters in the autoregressive model.

Inference. During the inference process, given a query text, the trained autoregressive language model **AM** could generate several predicted identifiers via beam search, denoted as \mathcal{I} . A heuristic function is employed to transform the predicted identifiers \mathcal{I} into a ranked list of passages. This function selects the predicted identifier $i_p \in \mathcal{I}_p$ for a given passage p if i_p appears at least once in the identifiers of that passage. The similarity score of the passage p corresponding to the query q is then calculated as the sum of the scores of its covered identifiers, as follows,

$$s_{stu} = \sum_{i_p \in \mathcal{I}_p} s_{i_p}, \quad (2)$$

where s_{i_p} represents the language model score of the identifier i_p , and \mathcal{I}_p is the set of selected identifiers that appear in the passage p . We just give a brief overview here, and please refer to the original paper for details.

By sorting the similarity score s_{stu} of the passage, we are able to obtain a rank list of passages from the student model, denoted as $\mathcal{R}_{stu} = (p_1, \dots, p_N)$, where N is the number of passages in \mathcal{R}_{stu} .

3.2 Distillation enhanced Generative Retrieval

To introduce distillation into generative retrieval, we must address two key questions. First, we should determine the teacher model and what type of knowledge to be distilled into the student model (the generative retrieval model). Second, we should choose an effective distillation loss for distilling the knowledge into the student model.

Teacher model and knowledge type. The teacher model should be more powerful in ranking passages than the student model. We consider existing cross-encoder rankers, which enable

fine-grained interaction between queries and passages, as the teacher models for generative retrieval. These rankers are trained using contrastive loss and produce logits as similarity scores. However, since generative retrieval is trained using generative loss, it cannot directly mimic the teacher’s similarity scores that have different distributions.

We propose using the rank order of the passages provided by the teacher model as the distilled knowledge. Specifically, the teacher model can rerank the retrieved passages \mathcal{R}_{stu} from the student model and give a new rank list $\mathcal{R}_{tea} = (p_1, \dots, p_M)$, where M is the number of passages in R_{tea} . The \mathcal{R}_{tea} has the better rank orders attributed to the superior teacher model. In practice, the value of M is typically set to be smaller than N limited by GPU memory. As a result, we sample M passages from \mathcal{R}_{stu} for the teacher model to rerank. We explore different sample strategies in Section 5.3.

We believe the rank order of passages in \mathcal{R}_{tea} is effective knowledge that could be used for distillation. Unlike the binary “1” and “0” relevance labels, the passage ranks reflect the varying degrees of relevance from the teacher model. This is evident in the different positions of negative passages in the ranking list, as opposed to having the same “0” label in binary labels.

Distillation loss. Selecting an appropriate distillation loss is the key to transferring knowledge from the teacher model to the generative retrieval model. Since we aim to distill the rank order of passages, pair-wise and list-wise rank losses seem to be natural choices. We could regard the rank order given by the teacher model as the true labels to apply pair-wise and list-wise rank losses. However, our practical findings, as detailed in Section 5.1, indicate that these rank losses may not effectively serve as distillation losses.

Therefore, we propose the distilled RankNet loss to better suit the DGR framework, as it can effectively leverage the rank order provided by the teacher model. For the rank list $\mathcal{R}_{tea} = (p_1, \dots, p_M)$, we denote $r_i \in [1, \dots, M]$ as the rank of the passage p_i . The distilled RankNet loss is formulated as follows,

$$\begin{cases} \mathcal{L}_{distill} = \sum_{i=1}^M \sum_{j=1}^M \mathbb{1}_{r_i < r_j} \max(0, s_{stu}^j - s_{stu}^i + m_{ij}), \\ m_{ij} = m_{base} + m_{gap} * (r_j - r_i - 1), \end{cases} \quad (3)$$

where s_{stu}^i represents the i th passage’s score provided by the student model, and m_{ij} denotes the margin value determined by the rank of the two pas-

sages in the rank list R_{tea} provided by the teacher model. m_{base} is the base margin value and m_{gap} is the incremental margin value.

Similar to RankNet (Burgess et al., 2005), our distilled RankNet loss also forms $M * (M - 1)$ pairs. Differently, we adjust the RankNet loss from the following two aspects. 1) The order in which passages are ranked is determined by the teacher model rather than by the true labels. 2) The RankNet loss function only requires that the former passage have a higher score than the latter. But the distilled RankNet introduces a margin for each pair, and the margin increases as the samples are further apart in the teacher’s rank list \mathcal{R}_{tea} . As our goal is to distill the rank orders of documents from the teacher model, the distillation loss must utilize the ranking list rather than just a pair. Therefore, documents ranked lower in the ranking list will be assigned a larger margin, allowing the student model to learn to push these documents further away, thus approaching the sorting results provided by the teacher. In this way, our distilled RankNet loss guarantees list-wise optimization under the guidance of the teacher model.

Training and inference. The final loss is defined as the interpolation of the generation loss and distillation loss: $\mathcal{L} = \alpha \mathcal{L}_{gen} + \mathcal{L}_{distill}$. It is noted that the gradients will be backpropagated to the generative retrieval model through the S_{stu} in Equation (3). More importantly, our distillation framework optimizes the neural model and does not add any additional burden to the original inference stage. After training, the autoregressive model can be used to retrieve passages as introduced in Section 3.1.

4 Experiments

4.1 Setup

Datasets and evaluation. We conducted experiments on two important retrieval scenarios, QA and Web search. For the QA scenarios we adopted the widely used NQ (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017) datasets under the DPR (Karpukhin et al., 2020) setting. The two datasets are based on about 21 million Wikipedia passages. As for the Web search scenarios, we adopted the MSMARCO dataset (Nguyen et al., 2016) and TREC Deep Learning (DL) Track 2019 (Craswell et al., 2020) and 2020 (Craswell et al., 2021). For evaluation metrics, we used the traditional metrics for each dataset: hits@5, @20,

Methods	Natural Questions			TriviaQA		
	@5	@20	@100	@5	@20	@100
BM25	43.6	62.9	78.1	67.7	77.3	83.9
DPR(Karpukhin et al., 2020)	68.3	80.1	86.1	72.7	80.2	84.8
GAR(Mao et al., 2021)	59.3	73.9	85.0	73.1	80.4	85.7
DSI-BART(Tay et al., 2022)	28.3	47.3	65.5	-	-	-
SEAL-LM(Bevilacqua et al., 2022)	40.5	60.2	73.1	39.6	57.5	80.1
SEAL-LM+FM(Bevilacqua et al., 2022)	43.9	65.8	81.1	38.4	56.6	80.1
SEAL(Bevilacqua et al., 2022)	61.3	76.2	86.3	66.8	77.6	84.6
MINDER(Li et al., 2023c)	65.8	78.3	86.7	68.4	78.1	84.8
LTRGR(Li et al., 2023b)	68.8	80.3	87.1	70.2	79.1	85.1
DGR	71.4[†]	81.4[†]	87.4[†]	72.6 [†]	80.4[†]	85.8[†]
% improve	3.78%	1.37%	0.34%	3.42%	1.64%	0.82%

Table 1: Retrieval performance on NQ and TriviaQA. Inapplicable results are marked by “-”. The best results in each group are marked in Bold, and [†] denotes the best result in generative retrieval. % improve represents the relative improvement achieved by DGR over the previously best generative retrieval method.

and @100 for NQ and TriviaQA; Recall and MRR for MSMARCO; nDCG for TREC DL. As we know, we are the pioneers in evaluating generative retrieval on such comprehensive benchmarks.

Baselines. We compared DGR with several generative retrieval methods, including DSI (Tay et al., 2022), DSI (scaling up) (Pradeep et al., 2023), NCI (Wang et al., 2022c), SEAL (Bevilacqua et al., 2022), MINDER (Li et al., 2023c), and LTRGR (Li et al., 2023b). Additionally, we included the term-based method BM25, as well as DPR (Karpukhin et al., 2020) and GAR (Mao et al., 2021). It is noted that the results of NCI on MSMARCO are from the work (Pradeep et al., 2023) rather than the work (Zhang et al., 2024). All baseline results were obtained from previous papers.

Implementation details. To ensure a fair comparison with previous work, we utilized BART-large as our backbone. In practice, we loaded the trained autoregressive model, MINDER (Li et al., 2023c), and continued training it using our proposed distillation framework. We used the released SimLM (cross-encoder)¹ as the teacher model. We set the M , N , m_{base} , m_{gap} , α , as 6, 200, 300, 100,100, and 500, respectively. We have trained the model several times to confirm that the improvement is not a result of random chance and present the mid one. Our experiments were conducted on four NVIDIA A5000 GPUs with 24 GB of memory.

¹<https://github.com/microsoft/unilm/tree/master/simlm>

4.2 Retrieval Results on QA

We summarized the results on NQ and TriviaQA in Table 1. Upon analyzing the results, we had the following findings:

(1) Among the generative retrieval methods, it was observed that DSI lags behind other generative retrieval approaches due to the lack of semantic information in numeric identifiers, requiring the model to memorize the mapping from passages to their numeric IDs. Consequently, DSI faces challenges with datasets like NQ and TriviaQA, which contain over 20 million passages. MINDER surpasses SEAL by using multiview identifiers to comprehensively represent a passage. LTRGR outperforms MINDER by leveraging additional learning-to-rank training. Notably, DGR significantly outperforms all generative retrieval approaches, including LTRGR, with improvements of 2.6 and 2.4 in hits@5 on NQ and TriviaQA, respectively. DGR, based on MINDER, notably enhances its performance through the distillation framework, improving hits@5 from 65.8 to 71.4 on NQ. This underscores the effectiveness of our proposed distillation-enhanced generative retrieval framework.

(2) Most generative retrieval approaches fall behind classical sparse and dense retriever baselines such as BM25 and DPR. While LTRGR only slightly outperforms DPR on NQ, it still performs worse on TriviaQA. However, our proposed DGR has propelled generative retrieval to a new level. DGR significantly outperforms DPR on NQ and slightly surpasses DPR on TriviaQA, marking the first time that generative retrieval has surpassed DPR in the QA scenario. It is important to acknowl-

Methods	Model Size	MSMARCO				TREC DL 19	TREC DL 20
		R@5	R@20	R@100	M@10	nDCG@10	nDCG@10
BM25	-	28.6	47.5	66.2	18.4	51.2	47.7
SEAL(Bevilacqua et al., 2022)	BART-Large	19.8	35.3	57.2	12.7	-	-
MINDER(Li et al., 2023c)	BART-Large	29.5	53.5	78.7	18.6	50.6	39.2
NCI(Wang et al., 2022c)	T5-Base	-	-	-	9.1	-	-
DSI(Pradeep et al., 2023)	T5-Base	-	-	-	17.3	-	-
DSI(Pradeep et al., 2023)	T5-Large	-	-	-	19.8	-	-
LTRGR(Li et al., 2023b)	BART-Large	40.2	64.5	85.2	25.5	58.7	54.7
DGR	BART-Large	42.9	67.6	86.5	26.6	59.5	58.3
% improve	-	6.71%	4.81%	1.53%	4.31%	1.36%	6.58%

Table 2: Retrieval performance on the MSMARCO dataset and TREC dataset. R and M denote Recall and MRR, respectively. “-” means the result not reported in the published work. The best results in each group are marked in Bold. The baselines’ results are from their respective papers. % improve represents the relative improvement achieved by DGR over the previously best generative retrieval method.

Teacher Models	Architecture	Natural Questions			TriviaQA		
		@5	@20	@100	@5	@20	@100
E5(Wang et al., 2022b)	Dual-encoder	70.7	81.3	87.5	72.8	80.6	85.9
SimLM(Wang et al., 2023)	Cross-encoder	71.4	81.4	87.4	72.6	80.4	85.8

Table 3: Analysis on DGR with different teacher models. Different teacher models produce the ranking list \mathcal{R}_{tea} .

Methods	Natural Questions		
	@5	@20	@100
Base model(Li et al., 2023c)	65.8	78.3	86.7
KL divergence	68.9	80.6	87.0
ListMLE(Xia et al., 2008)	68.5	80.1	87.1
ListNet(Cao et al., 2007)	68.9	80.6	87.0
approxNDCG(Qin et al., 2010)	68.5	79.9	87.0
RankNet(Burges et al., 2005)	68.1	80.2	87.0
LambdaLoss(Wang et al., 2018)	68.7	80.3	87.0
MarginMSE(Hofstätter et al., 2020)	35.8	52.3	66.1
Distilled RankNet	71.4	81.4	87.4

Table 4: Analysis on distillation losses. We applied KL divergence, various pair-wise and list-wise rank losses, and our proposed distilled RankNet, as the distillation loss, to the same base MINDER model.

edge that current generative retrieval approaches still exhibit a performance gap with state-of-the-art dense retrievers in the benchmarks. As a new paradigm in text retrieval, generative retrieval requires further research, and we have identified a promising direction to enhance its capabilities.

4.3 Retrieval Results on Web Search

We conducted experiments on the MSMARCO and TREC DL datasets to assess the performance of generative retrieval approaches on Web search. It is worth noting that evaluations of generative retrieval methods on commonly used benchmarks in this field are scarce, and we aim to advance the evaluation of generative retrieval in general search benchmarks.

Upon analyzing the results presented in Table 2,

several key findings have emerged. 1) Most generative retrieval methods exhibit lower performance compared to the basic BM25, and MINDER and DSI (T5-large) only marginally outperform BM25. This is attributed to the fact that web search passages are sourced from diverse web pages, often of lower quality and lacking important metadata such as titles. 2) LTRGR surpasses other generative retrieval baselines, benefiting from its learning-to-rank scheme. 3) DGR, based on the same MINDER backbone, outperforms LTRGR, which is reasonable given that DGR incorporates additional knowledge from the teacher models. 4) Current generative retrieval methods demonstrate a significant performance gap compared to the leading methods on the MSMARCO and TREC DL benchmarks, indicating the need for further efforts to address the challenges in general web search domains.

5 Analysis

5.1 Analysis on Distillation Loss

In this work, we introduced the distilled RankNet, as outlined in Eqn.(3), to distill the ranking order from the teacher model to a generative retrieval model. Our framework also allows other distillation losses to achieve the same goal. The KL divergence is a commonly used loss in knowledge distillation, and we could also optimize the generative retrieval model using pair and list-wise rank losses, treating the given ranking order as labels. In our experiments, we applied various distillation

Methods	Natural Questions		
	@5	@20	@100
Random	69.4	81.1	87.6
Top	70.4	80.4	86.7
Top&random	71.4	81.4	87.4

Table 5: Retrieval performance of different sampling strategies.

losses, including ListMLE, ListNet, approxNDCG, RankNet, LambdaLoss, and MarginMSE, to the base model MINDER, and the results are summarized in Table 4.

We obtained the following findings: 1) Our proposed distillation enhanced generative retrieval framework exhibits high generalizability. Both the KL divergence and various pair-wise and list-wise losses work within our framework. We have observed that the performance of the base model could be significantly enhanced by applying our framework with different distillation losses. 2) There is a noticeable disparity between our proposed distilled RankNet loss and others. While KL divergence aims to minimize the distance between two distributions, generative retrieval and cross-encoder rankers represent distinct retrieval paradigms with differing distributions. RankNet only necessitates that the positive item has a higher similarity score than the negative one in each pair, but it lacks list-wise optimization. As a result, we have devised the distilled RankNet loss, which establishes the incremental margins for the passages in the ranking list provided by the teacher model. Our experimental results confirm the importance of our proposed distilled RankNet loss, which significantly enhances performance.

5.2 Analysis on Teacher Models

In our DGR framework, we selected the cross-encoder ranker, specifically SimLM (cross-encoder), as the teacher model due to its effective ranking of passages through cross-attention between queries and passages. However, there are also dense retrievers with a dual-encoder architecture that outperforms generative retrieval. This raises the question of whether these advanced dense retrievers could serve as teacher models in the DGR framework. Consequently, we have presented the retrieval performance of the DGR framework with different teacher models featuring various architectures in Table 3.

The results demonstrate the robustness of the

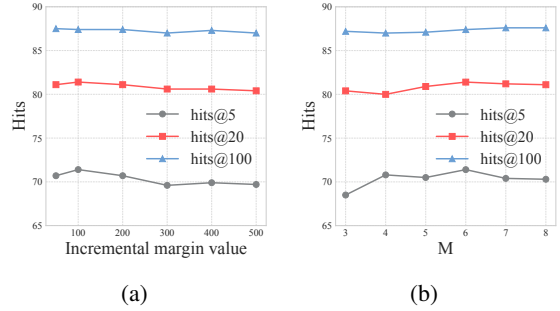


Figure 2: Retrieval performances of DGR on the NQ test set are depicted in (a) and (b) with respect to the incremental margin values m_{gap} and the number of passages, M , in \mathcal{R}_{tea} .

DGR framework across different teacher model architectures. We observed performance improvements when applying our DGR framework to the base model using both cross-encoder and dual-encoder architectures. This is attributed to our approach of utilizing the ranking order of the teacher model as the knowledge type, which ensures the framework’s resilience irrespective of the teacher model’s architecture. Additionally, we noticed similar performance improvements with different teacher models. While cross-encoder rankers are expected to have more powerful ranking abilities, the observed improvement was not significantly larger. We believe this may be due to the length of \mathcal{R}_{tea} . As a result of GPU resource constraints, we set the length to 6, which may not fully capture the varying ranking abilities of different teachers.

5.3 Analysis on Sampling Strategies

In DGR, we selected M passages from the passage list \mathcal{R}_{stu} to be reranked by the teacher model. We investigated the following sampling strategies:

- **-Random:** Randomly selecting M passages from \mathcal{R}_{stu} .
- **-Top:** Choosing the top-ranked M passages from \mathcal{R}_{stu} .
- **-Top&Random:** Selecting one top-ranked passage and randomly selecting $M - 1$ passages from \mathcal{R}_{stu} .

The results are summarized in Table 5, and we have made several key findings. 1) When comparing the “Random” and “Top” strategies, we observed that “Random” performs better in hits@100, while “Top” achieves better hits@5 performance. This is reasonable as the “Random” strategy evenly selects passages from \mathcal{R}_{stu} , benefiting hits@100,

Methods	Natural Questions		
	@5	@20	@100
SEAL	61.3	76.2	86.3
SEAL-LTR	63.7	78.1	86.4
SEAL-DGR	68.4	80.2	87.0

Table 6: Retrieval performance of SEAL, SEAL-LTR and SEAL-DGR on NQ. SEAL-LTR and SEAL-DGR represents applying our proposed DGR and LTRGR(Li et al., 2023b) framework to the SEAL model.

while the “Top” strategy selects top-ranked passages better for hits@5. 2) The “Top&random” strategy balances the two strategies and achieves better overall performance on hits@5, 20, and 100. It is worth noting that “Top&random” also outperforms “Top” in hits@5, as only considering the top M passages is too extreme, especially when M is set to 6 in our experiment.

5.4 In-depth Analysis

Incremental margin value m_{gap} . Our proposed distilled RankNet introduces a margin for each pair, with the margin increasing as the samples are further apart in the rank list \mathcal{R}_{tea} , as defined in Eqn. (3). To evaluate the impact of incremental value m_{gap} on retrieval performance, we manually set margin values ranging from 50 to 500 in Eq. 3 and summarized the results in Figure 2(a). Our findings indicate that the distilled RankNet with a margin of 100 performs better than the value of 50. However, as the margin value increases from 100 to 500, performance gradually declines. This suggests that a larger margin value indicates larger similarity gaps among passages in the rank list, but may not be suitable as many passages are also relevant to the query. Therefore, we looped the m_{gap} values to find the optimal one.

Number of passages M . The number of passages in \mathcal{R}_{tea} indicates how many passages the teacher model ranks. To determine the optimal length of \mathcal{R}_{tea} , we conducted a tuning experiment with different M values, and the results are summarized in Figure 2(a). It is observed that the retrieval performance gradually increases as M increases. This is reasonable as more passages in \mathcal{R}_{tea} represent more knowledge provided by the teacher model. However, it is important to note that a larger length requires more GPU memory, and we were only able to set it to 8.

Generalization of DGR. Our DGR builds on the generative retrieval model MINDER and continues to train it via the distillation loss. This leads

to the question of whether DGR can be generalized to other generative retrieval models. To address this, we replaced MINDER with SEAL as the basic model and enhanced it using our proposed distillation framework. Additionally, we compared it with another additional training framework (Li et al., 2023b). The results are presented in Table 6. The performance of SEAL showed significant improvement after applying our DGR framework. The hits@5 metric improved from 61.3 to 68.4, confirming the effectiveness of DGR on other base generative retrieval models. Furthermore, when compared with the learning-to-rank (LTR) framework, DGR also demonstrated significant improvement, with a 4.7-point increase in hits@5. This improvement can be attributed to DGR’s introduction of extra teacher models, allowing it to learn knowledge from these models.

Inference speed. DGR enhances base generative retrieval models, such as MINDER, through the distillation loss without impacting the inference process of the base generative retrieval model. As a result, the speed of inference remains the same as the underlying generative retrieval model. On the NQ test set, using one V100 GPU with 32GB memory, DGR, LTRGR, and MINDER took approximately 135 minutes to complete the inference process, while SEAL took only 115 minutes. It is worth noting that SEAL’s speed is comparable to that of the typical dense retriever, DPR, as reported in (Bevilacqua et al., 2022).

6 Conclusion and Future Work

In this study, we proposed the distillation enhanced generative retrieval (DGR) framework, which could significantly improve generative retrieval models by distilling knowledge from advanced rank models. Importantly, DGR is highly adaptable and can effectively work with different teacher models and distillation losses. Additionally, we introduced the distilled RankNet loss, which is tailored to generative retrieval and has been evident to be more effective than existing distillation losses. We conducted extensive experiments on four widely used benchmarks to verify the effectiveness and robustness of DGR. The results verify that distillation is a promising direction for improving generative retrieval systems.

In the future, we aim to improve DGR from the following aspects. We have investigated the use of cross-encoder and dual-encoder as teacher models,

and we will also explore the possibility of advanced generative retrieval models as teacher models for outdated generative retrieval models. Additionally, DGR has only begun to validate the feasibility of distillation in generative retrieval, leaving ample room for further research. Further studies, including more sampling strategies and progressive training methods, are worth exploring.

Acknowledgments

The work described in this paper was supported by National Natural Science Foundation of China (62076212), Research Grants Council of Hong Kong (PolyU/5210919, PolyU/15207821, and PolyU/15207122), and PolyU internal grants (ZVQ0).

Limitations

1) Due to limitations in GPU memory, we only assessed DGR using the maximum length of 8 for \mathcal{R}_{tea} . It is important to note that a longer length typically implies a greater amount of knowledge from the teacher model. Therefore, the performance of DGR could potentially be enhanced with a longer length, but we cannot verify this. 2) DGR achieves the best performance among the current generative retrieval methods, but it still falls short of the current state-of-the-art on the leaderboards. This is attributed to the model’s autoregressive generation method, which generates from left to right and may not fully capture the entire content of a passage. Further techniques should be explored for generative retrieval. Fortunately, distillation appears to be a promising direction, and we have assessed its feasibility. More studies following this direction are expected.

Ethics Statement

The datasets used in our experiment are publicly released and labeled through interaction with humans in English. In this process, user privacy is protected, and no personal information is contained in the dataset. The scientific artifacts that we used are available for research with permissive licenses. And the use of these artifacts in this paper is consistent with their intended use. Therefore, we believe that our research work meets the ethics of ACL.

References

- Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Wen-tau Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers. *arXiv preprint arXiv:2204.10628*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96.
- Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136.
- Wei-Cheng Chang, X Yu Felix, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2019. Pre-training tasks for embedding-based large-scale retrieval. In *International Conference on Learning Representations*.
- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1870–1879.
- Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the trec 2020 deep learning track. *arXiv preprint arXiv:2102.07662*.
- Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820*.
- Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. In *International Conference on Learning Representations*.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- 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, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with

- balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122.
- Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing*, pages 6769–6781. ACL.
- Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.
- 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 Association for Computational Linguistics*, 7:452–466.
- Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096. ACL.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.
- Yongqi Li, Xinyu Lin, Wenjie Wang, Fuli Feng, Liang Pang, Wenjie Li, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2024. A survey of generative search and recommendation in the era of large language models. *arXiv preprint arXiv:2404.16924*.
- Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2023a. Generative retrieval for conversational question answering. *Information Processing & Management*, 60(5):103475.
- Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2023b. Learning to rank in generative retrieval. *arXiv preprint arXiv:2306.15222*.
- Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2023c. Multiview identifiers enhanced generative retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6636–6648. ACL.
- Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4089–4100.
- 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 and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.
- Ronak Pradeep, Kai Hui, Jai Gupta, Adam Lelkes, Honglei Zhuang, Jimmy Lin, Donald Metzler, and Vinh Tran. 2023. How does generative retrieval scale to millions of passages? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1305–1321.
- Tao Qin, Tie-Yan Liu, and Hang Li. 2010. A general approximation framework for direct optimization of information retrieval measures. *Information retrieval*, 13:375–397.
- Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 5835–5847.
- Anshumali Shrivastava and Ping Li. 2014. Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips). *Advances in neural information processing systems*, 27.
- Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, Maarten de Rijke, and Zhaochun Ren. 2023. Learning to tokenize for generative retrieval. *arXiv preprint arXiv:2304.04171*.
- Yi Tay, Vinh Q Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. 2022. Transformer memory as a differentiable search index. *arXiv preprint arXiv:2202.06991*.
- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022a. Simlm: Pre-training with representation bottleneck for dense passage retrieval. *arXiv preprint arXiv:2207.02578*.
- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022b. Text embeddings by weakly-supervised contrastive pre-training.

- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2023. [Simlm: Pre-training with representation bottleneck for dense passage retrieval](#).
- Xuanhui Wang, Cheng Li, Nadav Golbandi, Michael Bendersky, and Marc Najork. 2018. The lambdaloss framework for ranking metric optimization. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 1313–1322.
- Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, et al. 2022c. A neural corpus indexer for document retrieval. *Advances in Neural Information Processing Systems*, 35:25600–25614.
- Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*, pages 1192–1199.
- Hansi Zeng, Chen Luo, Bowen Jin, Sheikh Muhammad Sarwar, Tianxin Wei, and Hamed Zamani. 2024. Scalable and effective generative information retrieval. In *Proceedings of the ACM on Web Conference 2024*, pages 1441–1452.
- Hansi Zeng, Hamed Zamani, and Vishwa Vinay. 2022. Curriculum learning for dense retrieval distillation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1979–1983.
- Hailin Zhang, Yujing Wang, Qi Chen, Ruiheng Chang, Ting Zhang, Ziming Miao, Yingyan Hou, Yang Ding, Xupeng Miao, Haonan Wang, et al. 2024. Model-enhanced vector index. *Advances in Neural Information Processing Systems*, 36.