UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Lightweight and Direct Document Relevance Optimization for Generative Information Retrieval

Mekonnen, K.A.; Tang, Y.; de Rijke, M.

DOI
10.1145/3726302.3730023

Publication date
2025

Document Version
Final published version

Published in
SIGIR '25

License
CC BY

Link to publication

Citation for published version (APA):
Mekonnen, K. A., Tang, Y., & de Rijke, M. (2025). Lightweight and Direct Document Relevance Optimization for Generative Information Retrieval. In *SIGIR '25: Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval : July 13-18, 2025, Padua, Italy* (pp. 1327-1338). Association for Computing Machinery. https://doi.org/10.1145/3726302.3730023

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:12 Sep 2026

Lightweight and Direct Document Relevance Optimization for Generative Information Retrieval

Kidist Amde Mekonnen
University of Amsterdam
Amsterdam, The Netherlands
k.a.mekonnen@uva.nl

Yubao Tang
University of Amsterdam
Amsterdam, The Netherlands
y.tang3@uva.nl

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
m.derijke@uva.nl

Abstract

Generative information retrieval (GenIR) is a promising neural retrieval paradigm that formulates document retrieval as a document identifier (docid) generation task, allowing for end-to-end optimization toward a unified global retrieval objective. However, existing GenIR models suffer from token-level misalignment, where models trained to predict the next token often fail to capture document-level relevance effectively. While reinforcement learning-based methods, such as reinforcement learning from relevance feedback (RLRF), aim to address this misalignment through reward modeling, they introduce significant complexity, requiring the optimization of an auxiliary reward function followed by reinforcement fine-tuning, which is computationally expensive and often unstable. To address these challenges, we propose *direct document relevance optimization* (DDRO), which aligns token-level docid generation with document-level relevance estimation through direct optimization via pairwise ranking, eliminating the need for explicit reward modeling and reinforcement learning. Experimental results on benchmark datasets, including MS MARCO document and Natural Questions, show that DDRO outperforms reinforcement learning-based methods, achieving a 7.4% improvement in MRR@10 for MS MARCO and a 19.9% improvement for Natural Questions. These findings highlight DDRO's potential to enhance retrieval effectiveness with a simplified optimization approach. By framing alignment as a direct optimization problem, DDRO simplifies the ranking optimization pipeline of GenIR models while offering a viable alternative to reinforcement learning-based methods.

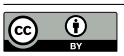
CCS Concepts

• **Information systems** → **Retrieval models and ranking**; **Learning to rank**; **Language models**.

Keywords

Generative information retrieval, Document relevance optimization, Ranking optimization, Learning to rank, Supervised fine-tuning

ACM Reference Format:
Kidist Amde Mekonnen, Yubao Tang, and Maarten de Rijke. 2025. Lightweight and Direct Document Relevance Optimization for Generative Information Retrieval. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3726302.3730023



This work is licensed under a Creative Commons Attribution 4.0 International License.
SIGIR '25, Padua, Italy
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3730023

1 Introduction

Building on the success of transformer-based pre-trained language models, recent research has explored various neural retrieval approaches [69]: learned sparse retrieval [17, 18], dense retrieval [22, 68, 72, 73], and cross-encoders [37]. A new paradigm has recently been added to this palette, *generative information retrieval* (GenIR) [34, 60]. This approach uses pre-trained encoder-decoder models as differentiable search indexes (DSI). It has inspired the development of several models [4, 33, 43, 50, 52–55, 64, 70, 71, 79, 80]

Generative information retrieval. GenIR models represent documents as sequences of *unique document identifiers* (docids), generated autoregressively, where each token is conditioned on the query encoding and previously generated tokens. The generation process is controlled through (constrained) beam search [33, 60, 64, 70, 71, 80]. Docids could be predefined and remain static during training, making their careful design crucial for optimal retrieval performance [58]. We classify docids into two categories based on their generation methodology and abstraction level. The first category, referred to as *content-derived* docids, includes identifiers that are extracted directly from document elements such as titles [12, 13, 16, 25, 51, 56, 57, 59], n-grams [4, 11, 28, 29, 65], URLs [43, 76, 79, 82], and key terms [75]. These docids preserve surface-level textual characteristics and are closely tied to the original document content. In contrast, the second category, termed *computationally-generated* are derived using techniques like quantization [10, 42, 70, 71, 79] or hierarchical clustering algorithms [33, 48, 60, 64] to encode deeper semantics by abstracting raw document content into conceptual features.

During training, GenIR models learn to associate document text with corresponding docids, embedding semantic information directly into its parameters. During retrieval, docids are sequentially generated based on learned representations. By unifying indexing and retrieval within a transformer-based architecture, these models optimize both processes simultaneously [34, 60].

Challenges in GenIR. Despite recent advancements, GenIR models face key limitations that hinder their effectiveness. These models typically rely on an auto-regressive loss function that optimizes the generation of individual docid tokens. However, this token-level optimization approach does not align with the broader goal of ranking tasks, which requires assessing the overall relevance of a document to the query. As a result, this misalignment often leads to suboptimal ranking performance. *To address these challenges, it is crucial to align token generation with document-level relevance estimation to ensure more accurate, well-rounded retrieval outcomes.*

Direct document-level relevance optimization. Existing approaches aimed at aligning token-level docid generation with document-level relevance estimation, such as reinforcement learning from relevance feedback (RLRF), use reinforcement learning to align predictions with relevance judgments through reward modeling [76]. However, RLRF introduces significant complexity, requiring the optimization of an auxiliary reward function followed by reinforcement fine-tuning, which is computationally expensive and often unstable. To address these challenges, we introduce a *direct document relevance optimization* (DDRO) method that employs a pairwise ranking approach into GenIR models to improve document retrieval performance.

Our approach has two key phases. (i) First, we employ supervised fine-tuning (SFT) to train a language model (LM) capable of generating docids that are most relevant to a given query. (ii) Next, we directly refine the model through pairwise ranking, where the model learns to differentiate between relevant and irrelevant docids for a specific query based on labeled data. This refinement ensures that the retrieval system ranks documents based on their relevance, aligning the model more closely with the query. The SFT phase serves as a pretraining step, aligning the model with initial relevance signals from training data and providing a strong foundation for the pairwise ranker to further fine-tune document ranking effectiveness.

Experiments conducted on the MS MARCO document ranking [3] and Natural Questions (NQ) [23] benchmarks demonstrate the effectiveness of DDRO in improving retrieval accuracy, outperforming multiple baselines. Moreover, DDRO maintains competitive performance with established baselines on broader metrics like R@10, demonstrating robustness across evaluation criteria. An ablation study further highlights the contributions of pairwise ranking optimization to the observed performance improvements.

Main contributions. We introduce direct document relevance optimization (DDRO), a pairwise ranking approach that aligns docid generation with document-level relevance judgments. This approach ensures that docids are generated not only based on token-level likelihood but also according to their relevance to the user's query. DDRO unifies training objectives within a single framework, optimizing directly for document-level relevance. Experimental results demonstrate improvements in retrieval accuracy, highlighting the effectiveness of the proposed approach in enhancing generative retrieval models for relevance-based ranking.

Reproducibility. To promote reproducibility in GenIR, we open-source our codebase and make checkpoints publicly available.[1]

2 Related Work

Reward modeling. Recent efforts in GenIR have sought to bridge the gap between token-level optimization and document-level relevance. GenRRL [76] addresses this issue using RLRF to align docid generation with query relevance. While effective, this approach requires a robust reward model training and reinforcement learning fine-tuning, both of which are resource-intensive and prone to instability. Developing a reliable reward model demands substantial labeled data, and reinforcement learning fine-tuning involves extensive hyperparameter tuning [40], contributing to training instability and scalability challenges for large-scale applications. In contrast, we propose DDRO, a direct document-level relevance optimization method that eliminates the need for explicit reward model training and reinforcement learning fine-tuning, thereby reducing computational overhead and improving optimization efficiency.

Dense-generative integration. Ranking-oriented generative retrieval (ROGER) [77] combines dense and generative retrieval by using dense retrievers as intermediaries to provide relevance signals, bridging the gap between document ranking and docid generation. ROGER employs knowledge distillation from dense retrievers to enhance the generative model's ranking capabilities, combining the strong relative ranking signals of dense retrieval with the flexibility of generative models. However, it relies on external dense retrievers and does not directly optimize for document-level relevance within the generative model's training objectives. In contrast, DDRO eliminates this dependency by incorporating pairwise ranking directly into the generative model's optimization pipeline, ensuring alignment with document-level relevance.

Learning to rank in generative retrieval models. Similarly, LTRGR [29] incorporates a learning-to-rank (LTR) framework to address the gap between docid generation and document ranking. It introduces an additional training phase where the model is optimized using a margin-based ranking loss, eliminating the need for a separate ranking step during inference. However, LTRGR focuses on optimizing passage ranking during the second phase, treating docid generation as a step toward this goal rather than fully integrating document-level relevance into the generative process. Consequently, the challenge of embedding document-level relevance directly into docid generation remains unaddressed. In contrast, DDRO integrates pairwise ranking directly into the generative model's optimization pipeline, ensuring docid generation inherently aligns with document-level relevance.

Our approach. Building on the advancements of GenRRL, ROGER, and LTRGR, we propose a framework that combines SFT for docid generation with pairwise ranking optimization to better align GenIR objectives with ranking goals. DDRO addresses token-level misalignment by incorporating document-level relevance optimization into the training process, enhancing existing GenIR systems by enabling them to learn to rank more effectively.

State-of-the-art baselines. SOTA baselines in GenIR, such as RIPOR [70] and PAG [71], employ multi-stage optimization approaches. E.g., RIPOR refines relevance-based docids through iterative pre-training and fine-tuning, while PAG introduces a hybrid decoding strategy that combines simultaneous and sequential decoding to enhance ranking efficiency. Both methods achieve strong results on the MS MARCO passage ranking dataset. Our work simplifies optimization with a single-framework approach, offering an alternative to multi-stage methods. A direct comparison with these baselines is deferred to future work to assess how DDRO can complement and extend these approaches while evaluating its scalability in large-scale retrieval tasks.

[1] https://github.com/kidist-amde/DDRO-Direct-Document-Relevance-Optimization/tree/main

3 Preliminaries and Motivations

3.1 Generative Information Retrieval (GenIR)

GenIR models build on large pre-trained language models, such as T5 [41] and BART [27], and are designed to take a query string and generate a ranked list of document identifiers (docids) based on their generation probabilities, ordered in descending sequence. Each document d is assigned a unique identifier docid = $(\text{docid}_1, \text{docid}_2, \ldots, \text{docid}_L)$, where L is the length of the identifier, and the model processes the query q to autoregressively generate the corresponding docid using a scoring function defined as:

$$score(\text{docid} \mid q) = \prod_{i=1}^{L} p_\theta(\text{docid}_i \mid \text{docid}_{1:i-1}, q), \tag{1}$$

where p_θ denotes the generative retrieval model parameterized by θ, and docid_i is the i-th token of the docid for document d. The generation continues until a special end-of-sequence (EOS) token is decoded.

Training is performed using a multi-task setup that combines indexing and fine-tuning, which yields better results than training these tasks separately [60]. During indexing, the model memorizes the document collection and maps each document's text to its corresponding docid. Fine-tuning then refines this mapping by optimizing query-to-docid associations. The model is optimized via the following loss L^{θ}_{DSI} with teacher forcing [66]:

$$\sum_{d_i \in D} \log P(docid_i \mid T5_\theta(d_i)) + \sum_{q_j \in Q} \log P(docid_i \mid T5_\theta(q_j)), \tag{2}$$

where D represents the document set, and Q denotes the query set. This loss function enables parameter updates during both indexing and fine-tuning, enhancing the model's ability to generate the most relevant docid for a given query or document. The retrieval phase employs a (constrained) beam search algorithm to decode the most probable docid, with their generation probabilities determining the final ranking [33, 60, 64, 70, 71, 80].

3.2 Learning to Rank (L2R)

L2R aims at training models to rank documents based on their relevance to a given query [6, 9, 63]. L2R methods can be classified into point-wise, pair-wise, and list-wise approaches based on their learning objectives. (i) Point-wise methods [19] frame ranking as a classification problem by scoring individual documents independently: $L_{\text{point}} = \sum_i \mathcal{L}(\hat{s}(d_i, q), s(d_i, q))$, where $\hat{s}(d_i, q)$ and $s(d_i, q)$ denote the predicted relevance score and ground truth relevance score, respectively. In GenIR, generated probabilities serve as relevance scores, aligning with this approach [60]. And the retrieval term in Eq. 2 belongs to this type. (ii) Pair-wise approaches [6, 9, 15, 63] compare document pairs to determine relative preferences: $L_{\text{pair}} = \sum_{(d_i,d_j)} \log\left(1 + \exp\left(-\left(\hat{s}(d_i, q) - \hat{s}(d_j, q)\right)\right)\right)$, where d_i and d_j are used as pairs to compare. DDRO shares similarities with traditional pairwise L2R methods such as RankNet [5] and LambdaRank [7], in that it optimizes a margin between relevant and non-relevant documents. It differs in that the ranking signal is used to supervise the generation of structured docid sequences via a generative decoder. Unlike typical L2R approaches that score documents retrieved by an external system, DDRO learns to produce docids directly, making it end-to-end generative. This integration of sequence modeling and pairwise supervision is a key distinction from prior L2R pipelines. (iii) List-wise approaches [24, 67], optimize the entire ranked list: $L_{\text{list}} = \sum_q \mathcal{L}\left(\text{Softmax}(\hat{s}(q, \hat{\pi})), \text{Softmax}(s(q, \pi))\right)$, where $\hat{\pi}$ and π are the predicted and ground-truth lists, respectively. In GenIR, Tang et al. [53] introduces a position-aware list-level objective to learn the relevance. As we focus on pair-wise approaches, comparisons with list-wise methods are left for future work.

A fundamental challenge in GenIR stems from the inherent misalignment between the optimization objectives of autoregressive models and the overarching objectives of document ranking tasks. Training GenIR models solely to generate docids can be treat as the point-wise approach, which is often insufficient for achieving effective ranking. Addressing this challenge necessitates the development of a robust framework that enables GenIR models to directly learn to rank [29, 76, 77].

3.3 Reinforcement Learning from Relevance Feedback (RLRF)

To address the aforementioned limitations, Zhou et al. [76] propose GenRRL, a generative retrieval model based on RLRF to optimize generative models for alignment with document-level relevance. RLRF optimizes rewards while ensuring alignment with human preferences using a KL divergence constraint [2, 14, 30, 35, 39, 47, 76]. This method refines model predictions using a learned reward function, as formalized in prior research [20, 21]:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}\left[r_\phi(x, y)\right] - \beta\mathbb{D}_{\text{KL}}\left(\pi_\theta(y|x) \| \pi^{\text{ref}}(y|x)\right), \tag{3}$$

where β is a parameter controlling deviation from the base reference policy π^{ref}, which is typically the initial supervised fine-tuned model. This constraint prevents the model from straying too far from the data distribution used to train the reward function, preserving output diversity and avoiding overfitting to high-reward responses. Since language generation is non-differentiable, reinforcement learning (RL) techniques are widely used. A widely recognized approach [39, 47, 81] optimizes the reward function using Proximal Policy Optimization (PPO) [46], and defines the reward function as:

$$r(x, y) = r_\phi(x, y) - \beta(\log \pi_\theta(y|x) - \log \pi^{\text{ref}}(y|x)). \tag{4}$$

GenRRL [76] trains a reward model using relevance-annotated data derived from BM25 [44], DPR [22], and LLaMA-13b [61]. This reward model guides reinforcement learning to optimize the language model's policy for generating high-reward outputs, with a KL divergence constraint ensuring alignment with the original supervised fine-tuned model or the reference policy π^{ref}. The optimization process involves supervised fine-tuning using negative log-likelihood, pairwise ranking loss for the reward model, and reinforcement learning techniques incorporating pointwise, pairwise, and listwise approaches to enhance ranking performance. While effective, this approach introduces considerable complexity, requiring the training of multiple models and sampling from the policy during training, which substantially increases computational costs.

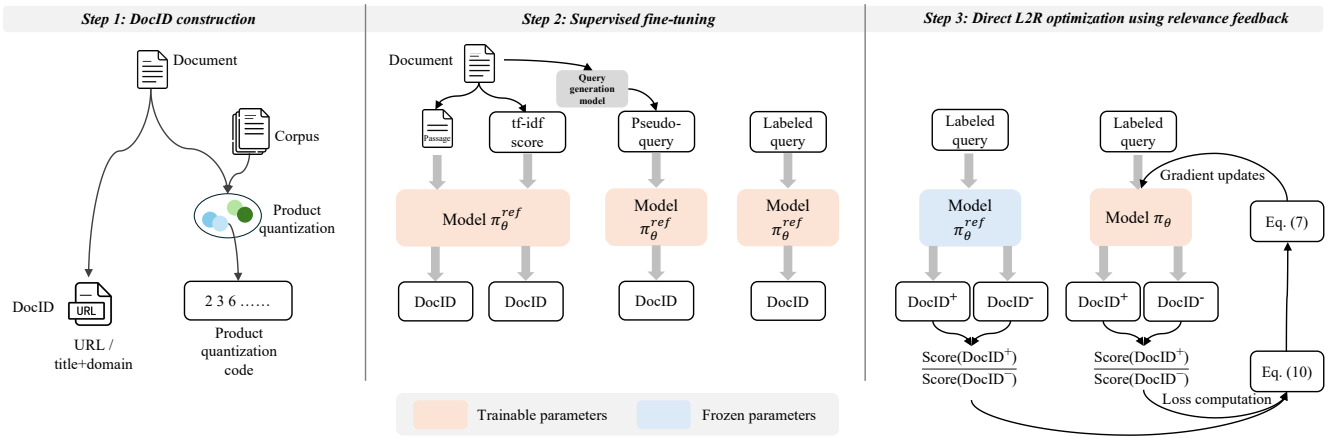


Figure 1: The proposed workflow comprises three key stages: (1) Construction of document identifiers (docids) including URL/title, domain, and product quantization codes; (2) Supervised fine-tuning of the retrieval model π_θ^{ref} using diverse data pairs; and (3) Freezing the trained reference policy model π_θ^{ref} and performing direct learning-to-rank (L2R) optimization on a policy model π_θ.

Inspired by the work of Rafailov et al. [40], we propose *direct document relevance optimization* (DDRO), a streamlined approach designed to enhance the ranking capabilities of GenIR models. DDRO directly optimizes GenIR models to learn document-level ranking without relying on explicit reward modeling or reinforcement learning. While our optimization is inspired by the DPO framework [40], its adaptation to GenIR is non-trivial. Unlike preference alignment for open-ended generation, our task involves optimizing structured docid generation under beam decoding constraints. Additionally, our method differs in both architecture (encoder-decoder vs. decoder-only) and objective (document ranking vs. preference alignment), requiring novel integration into GenIR pipelines. To the best of our knowledge, DDRO is the first method to apply preference-style optimization directly to generative document retrieval by extending DPO-style training to constrained generative settings, leveraging pairwise query-docid relevance supervision and constrained decoding.

4 Method

We provide a comprehensive explanation of the proposed DDRO method. As depicted in Figure (1), the method initiates with the generation of two categories of docids, designed to encapsulate diverse semantic and contextual features of the documents (Section 4.1). Subsequently, the retrieval model is trained through a combination of self-supervised and supervised learning techniques (Section 4.2). Finally, direct learning-to-rank (L2R) optimization is applied, using relevance feedback to refine the model's ranking quality and align its outputs with document-level relevance (Section 4.3).

4.1 Docid Construction

The methodology for constructing docids in this work is grounded in established frameworks [70, 71, 76, 79]. This approach uses keyword-based identifiers to effectively encapsulate the semantic and contextual information of the documents.

URL and Title (TU). Titles in web search results are typically crafted to be descriptive, closely aligning with user search intent, while URLs often contain structured tokens, such as keywords or domains, that are highly indicative of relevance for web-based queries [79]. The structure of the URL reverses to prioritize semantically meaningful segments. When a URL lacks descriptive content (e.g., uses numeric IDs or generic paths), we fall back to a combination of the document's title and domain name as an alternative identifier. Formally, this docid variant is defined as:

$$docid_{\text{TU}} = \begin{cases} \text{reverse(URL)}, & \text{if the URL is semantically rich,} \\ \text{title + domain}, & \text{otherwise.} \end{cases} \tag{5}$$

Product quantization codes (PQ). Building on prior work [10, 42, 70, 79], we adopt product quantization (PQ) to reduce the dimensionality of document representations while maintaining their semantic integrity. PQ compresses document vectors into latent semantic tokens by employing K-means clustering to partition the latent vector space into clusters. Each document is then represented by the corresponding cluster center, forming a compact identifier that preserves the document's core semantic features. The resulting docid is defined as:

$$docid_{PQ} = PQ(\text{Encoder}(d)), \tag{6}$$

where the encoder is based on a pre-trained T5 model [36]. The clustering process generates $\boldsymbol{k}$ cluster centers across $\boldsymbol{n}$ groups, expanding the vocabulary by $\boldsymbol{n} \times \boldsymbol{k}$ new tokens. This approach produces a semantically rich and efficient representation of each document.

4.2 Supervised Fine-tuning

Supervised fine-tuning (SFT) enhances the retrieval capabilities of pre-trained language models by aligning them with task-specific data [1, 39, 47]. Based on the two basic operations of DSI [60], i.e., indexing and retrieval tasks, diverse data pairs are curated and optimized using a teacher forcing policy [66] to achieve alignment with the ground truth.

Indexing task. To memorize the corpus, the indexing task learns associations between documents and docids, making the document input format a crucial factor. Inspired by the indexing strategy proposed in [79], we generate self-supervised learning signals directly from the document corpus.

$$\nabla_\theta \mathcal{L}_{\text{DDRO}}\left(\pi_\theta; \pi^{\text{ref}}\right) = -\beta \mathbb{E}_{(q,docid^+,docid^-)\sim\mathcal{D}}[\underbrace{\sigma\left(\hat{r}_\theta\left(q, docid^-\right) - \hat{r}_\theta\left(q, docid^+\right)\right)}_{\text{higher weight when reward estimate is wrong}} \times [\underbrace{\nabla_\theta \log \pi\left(docid^+ \mid q\right)}_{\text{increase likelihood of } docid^+} - \underbrace{\nabla_\theta \log \pi\left(docid^- \mid q\right)}_{\text{decrease likelihood of } docid^-}]], \tag{7}$$

Figure 2: Gradient for direct learning-to-rank optimization using relevance feedback.

Text segments are mapped to their corresponding docids, enabling the model to link document passages with their broader context [8, 79]. Each document is divided into fixed-size passages, paired with the document's docid to create passage-to-docid pairs. For a document containing N terms, $\{w_1, w_2, \ldots, w_N\}$, multiple passage-to-docid pairs are generated as follows:

$$\text{passage} : \{w_i, w_{i+1}, \ldots, w_{i+m-1}\} \rightarrow docid, \tag{7}$$

where i is the starting term of a passage, and m is the fixed passage length. To emphasize a document's core semantic content, terms are prioritized by their $tf\text{-}idf$ scores [44], with a subset of high-scoring terms selected to form a compressed representation, which is then mapped to the document's docid:

$$\text{terms} : \{w_a, w_b, w_c\} \rightarrow docid, \tag{8}$$

where w_a, w_b, w_c are key terms selected based on their $tf\text{-}idf$ scores.

Retrieval task. During retrieval, a pre-trained language model is fine-tuned with supervised query-docid pairs to learn semantic mappings between queries and their corresponding docids. A key challenge in this process is the scarcity of labeled click data, which limits the ability to establish effective query-to-docid associations. To mitigate this, pseudo-queries are generated directly from the document corpus [64, 80]. Specifically, the docTTTTTquery [38] model is fine-tuned using supervised click data from the MS MARCO document and NQ datasets. For each document, an initial passage serves as input, and the model produces k predicted queries using beam search, denoted as $Q = \{q_1, \ldots, q_k\}$.

These diverse datasets, including passage-to-docid pairs, supervised query-docid pairs (derived from real-world relevance judgments), and synthetic pseudo-queries, are collectively used to train the π_θ^{ref} model. Using these comprehensive and diverse training data, the SFT model acquires a robust understanding of query-to-document mappings and learns to generate relevant docids by optimizing token-level generation probabilities for a given query.

Training objective. The model is trained with a sequence-to-sequence objective, aiming to maximize the likelihood of the target sequence through teacher forcing [66]. Given an input sequence s, which can be any of the document formats or queries described above, the objective is defined as:

$$L_{\text{SFT}}^{\theta} = \arg\max_{\theta} \log P(docid^* \mid s, \pi_\theta^{\text{ref}}(docid)), \tag{9}$$

where $docid^*$ represents the ground truth sequence, $\pi_\theta^{\text{ref}}(docid)$ denotes the sequence generated by the SFT model, and $P(docid^* \mid s, \pi_\theta^{\text{ref}}(docid))$ corresponds to the conditional probability of the ground truth given the input sequence and the model's generated sequence.

4.3 Direct L2R Optimization Using Relevance Feedback

DDRO simplifies the complexities associated with reward modeling and reinforcement learning used in RLRF approaches. Instead, it directly optimizes the likelihood that relevant docids ($docid^+$) are assigned higher scores over non-relevant ones ($docid^-$) for a given query, as illustrated in Figure (3). The corresponding optimization objective is formulated as:

$$\mathcal{L}_{\text{DDRO}}(\pi_\theta; \pi^{\text{ref}}) = -\mathbb{E}_{(q,docid^+,docid^-)\sim D} \left[\log \sigma\left(\beta \log \frac{\pi_\theta(docid^+ \mid q)}{\pi^{\text{ref}}(docid^+ \mid q)} - \beta \log \frac{\pi_\theta(docid^- \mid q)}{\pi^{\text{ref}}(docid^- \mid q)}\right)\right], \tag{10}$$

where $\pi_\theta(docid|q)$ is the policy that is being optimized, while $\pi^{\text{ref}}(docid|q)$ is the reference policy, typically the fine-tuned model (SFT). This formulation ensures that the optimized model remains close to the reference policy while improving relevance-based ranking. The DDRO update guides the model toward producing outputs better aligned with relevance by utilizing pairwise comparisons, offering a simplified alternative to RL-based approaches. The gradient w.r.t. the model parameters θ is defined as in Eq. 7, where

$$\hat{r}_\theta(q, docid) = \beta \log \frac{\pi_\theta(docid \mid q)}{\pi^{\text{ref}}(docid \mid q)} \tag{11}$$

is the reward implicitly defined by the model π_θ and the reference model π^{ref}. The examples are weighted according to how much the implicit reward model $\hat{r}_\theta$ overestimates the ranking of the non-relevant docid compared to the relevant docid. This weighting is scaled by β, reflecting the degree of misjudgment while considering the strength of the KL divergence constraint. The sigmoid function $\sigma(\cdot)$ ensures smooth optimization, and the model parameters θ are adjusted to increase the likelihood of the relevant docids over non-relevant ones. This reparameterization streamlines the training process by eliminating the need for an explicit reward model and iterative fine-tuning, providing a more stable and efficient framework for aligning the retrieval model with document-level relevance objectives.

5 Experimental Settings

5.1 Datasets and Evaluation Metrics

Datasets. We conduct our experiments using two widely recognized benchmarks: the **MS MARCO Document Ranking** dataset[2] [3] and the **Natural Questions (NQ)** dataset[3] [23]. The MS MARCO document ranking dataset is widely used for document ranking tasks and contains a large collection of queries and web pages. Following prior work [65, 76, 78, 79], we use a subset with 320k documents and 360k query-document pairs for training. The NQ dataset,

[2] https://microsoft.github.io/msmarco/Datasets.html#document-ranking-dataset
[3] https://ai.google.com/research/NaturalQuestions/download

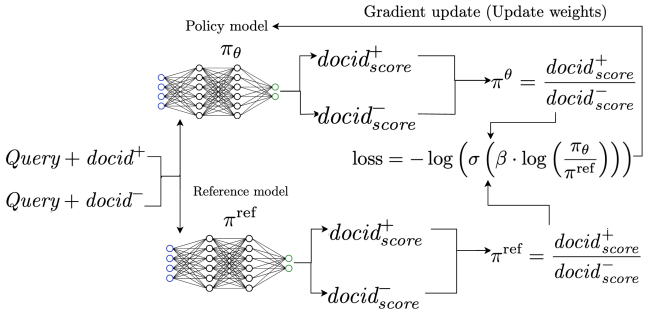


Figure 3: Architecture of the DDRO model, which fine-tunes the retrieval model through direct learning-to-rank (L2R) optimization using relevance feedback. Unlike GenRRL [76], DDRO directly optimizes with relevance judgment data, avoiding reinforcement learning, explicit reward modeling, and extensive hyperparameter tuning. *For clarity*, the model π_θ^{ref} from the SFT phase is referred to as π^{ref}, with its parameters frozen during this phase.

introduced by Google, is a widely used benchmark in question-answering research. In this study, we use the NQ320k version, which includes 320k query-document pairs sourced from Wikipedia, with queries formulated in natural language. To ensure reliable evaluation and improve performance [26], we deduplicate documents by title and utilize the predefined training and validation splits.

Evaluation metrics. Following [65, 76–79], we assess model performance using standard document retrieval metrics: Recall (R@1/5/10) and Mean Reciprocal Rank (MRR@10). Statistical significance is determined using paired t-tests with a threshold of $p < 0.05$.

5.2 Baselines

We evaluate our approach against three types of baseline: term-based retrieval, dense retrieval, and generative retrieval.

Term-based retrieval. (i) **BM25** [45], a probabilistic retrieval model commonly used as a standard baseline, implemented using Pyserini.[4] (ii) **DocT5Query** [38], which generates synthetic queries from documents using the T5 model [41], appending them to the original document text.

Dense retrieval. (i) **DPR** [22], which utilizes a BERT-based dual encoder to produce dense embeddings for queries and documents. PseudoQ [49] improves DPR by generating pseudo-queries using K-means clustering over document embeddings. (ii) **ANCE** [68], a RoBERTa-based dual encoder that incorporates hard negatives retrieved from an asynchronously updated approximate nearest neighbor (ANN) index. (iii) **RepBERT** [74], a BERT-based model that generates fixed-length contextualized embeddings, with query–document relevance computed via inner product similarity. (iv) **Sentence-T5** [36], which applies a T5-based architecture to generate sentence embeddings using encoder-only and encoder-decoder models with contrastive learning.

Generative retrieval. (i) **DSI** [60], which represents docids using hierarchical k-means cluster IDs and trains with the DSI-Num objective. (ii) **DSI-QG** [80], which augments training data with synthetic queries generated using a query generation model [38] and represents documents with arbitrary unique numerical docids. (iii) **NCI** [64], which assigns semantically structured numeric docids paired with pseudo-queries. (iv) **SEAL** [16], which retrieves docids represented as arbitrary n-grams extracted from document text using an FM-index. (v) **Ultron** [79], which employs keyword and semantic-based docids, using a three-stage training approach: general pre-training, search-oriented pre-training, and supervised fine-tuning. (vi) **ROGER** [77], which transfers document relevance knowledge from a dense retriever to a generative retriever via knowledge distillation. (vii) **MINDER** [28], which assigns multiple identifiers, including titles, n-grams, and synthetic queries, to documents and pairs them for indexing. (viii) **LTRGR** [29], which trains on pairwise relevance objectives using margin-based ranking loss for optimization. (ix) **GenRRL** [76], which incorporates pointwise, pairwise, and listwise relevance optimization through reinforcement learning, using document summaries and URLs as docids. We exclude document summaries as docids due to their dependence on external summarization models, such as LLaMA-13b used in GenRRL. These models introduce preprocessing overhead and variability in identifier quality. Instead, DDRO employs product quantization (PQ) to generate compact, structured docids, ensuring consistency and scalability.

Note on result sourcing. Baseline results for methods such as GenRRL [76] and ROGER [77] are taken from their original papers due to the unavailability of public code, ensuring consistency and avoiding potential reproducibility issues. Other results were reproduced using publicly available code and the dataset configurations described in this work.

5.3 Implementation Details

SFT. The SFT model is based on the T5-base pretrained model [41], trained with a learning rate of `1e-3` and a batch size of 128. All experiments involving various docid types were conducted on 8 NVIDIA RTX A6000 GPUs.

Pseudo queries. The DocT5Query model [38], fine-tuned on the target dataset with document-query pairs, was used to generate 10 pseudo-queries per document.

Contrastive data pair construction. Training triples were generated using stratified sampling for diversity. Positive samples were selected based on qrels relevance judgments, while negatives were drawn from the top 1000 BM25-retrieved documents, stratified into top (1–100), mid (101–500), and lower (501–1000) ranks. Negatives were randomly sampled in roughly equal proportions, with 8 per query for NQ and 16 for MS MARCO.

DDRO. The DDRO model was initialized with the pre-trained autoregressive SFT model (see Section 4.2) and fine-tuned using the proposed direct learning-to-rank framework (see Section 4.3). Training was performed using a modified Hugging Face TRL `DPOTrainer` [62], adapted for encoder-decoder models. A cosine learning rate scheduler with 1000 warm-up steps and early stopping was applied. The learning rate was set to `5e-6` for PQ-based docids and `1e-5` for URL-based docids, with a batch size of 64 and a regularization parameter β of 0.4 to balance chosen and rejected responses. All experiments were conducted on a single NVIDIA A100 GPU.

[4] https://github.com/castorini/pyserini

Table 1: Performance comparison of GenRRL and DDRO on the MS MARCO document ranking (MS300K) dataset. The best results are in bold. Results for cited models are sourced from their original papers. Abbreviations: PQ – Product Quantization; TU – Title + URL; Sum – document summary.

Model	R@1	R@5	R@10	MRR@10
GenRRL (TU) [76]	33.01	63.62	74.91	45.93
GenRRL (Sum) [76]	33.23	64.48	**75.80**	46.62
DDRO (PQ)	32.92	64.36	73.02	45.76
DDRO (TU)	**38.24**	**66.46**	74.01	**50.07**

Constrained beam search. During inference, constrained beam search generates valid docids by using a prefix tree [60] to enforce valid token sequences.

6 Experimental Evaluation and Results

Our evaluation of DDRO focuses on the following questions:

RQ1 How does DDRO compare to RLRF-based methods, such as GenRRL, in terms of retrieval performance while avoiding the complexities of reward modeling and reinforcement learning?

RQ2 How does DDRO perform relative to established baselines in terms of retrieval accuracy and ranking consistency on benchmark datasets?

RQ3 What is the impact of pairwise ranking optimization on the performance of DDRO?

RQ4 How robust is DDRO across datasets with varying characteristics?

RQ5 How does DDRO balance relevance across the ranked list in generative retrieval models, and what impact does it have on overall ranking quality?

6.1 Comparison with Reinforcement Learning-Based Methods

To address RQ1, we compare DDRO with GenRRL [76] on both datasets. Results are presented in Table 1 and Table 2.

We observe the following: (i) On MS MARCO, DDRO (TU) achieves the highest scores in early precision-focused metrics. Specifically, it outperforms GenRRL(Sum) by 15.06% in R@1 and 7.4% in MRR@10, highlighting its effectiveness in ranking the most relevant document at the top. While GenRRL (Sum) performs better in the R@10 metric, DDRO (TU) achieves comparable results with a simplified optimization process, avoiding the need for a reward model and reinforcement learning. (ii) On NQ, DDRO (PQ) outperforms GenRRL (Sum) by 34.69% in R@1 and 19.87% in MRR@10, further validating its effectiveness in retrieving relevant documents within the top ranks. GenRRL variants achieve higher R@10 scores, likely benefiting from multi-signal learning and listwise optimization strategies, potentially improving broader document retrieval. Summary-based docids may aid performance on knowledge-intensive queries. (iii) DDRO (TU) and DDRO (PQ) show different trends across datasets: TU performs better on MS300K, while PQ excels on NQ. This disparity likely stems from higher document quality in NQ, where the rich semantic content allows the generated PQs to convey more meaningful information. In contrast, MS300K, which is derived from web search logs, contains noisy content such as ads, resulting in lower-quality PQs. Consequently, TU, which prioritizes keyword-based features, effectively captures the core content of MS300K, where the shorter, keyword-focused queries align well with surface-level signals such as titles and URLs.

Table 2: Performance comparison of GenRRL and DDRO on the NQ320K dataset. The best results are in bold. Results for cited models are sourced from their original papers. Abbreviations: PQ – Product Quantization; TU – Title + URL; Sum – document summary.

Model	R@1	R@5	R@10	MRR@10
GenRRL (TU) [76]	35.79	56.49	70.96	45.73
GenRRL (Sum) [76]	36.32	57.42	**71.49**	46.31
DDRO (TU)	40.86	53.12	55.98	45.99
DDRO (PQ)	**48.92**	**64.10**	67.31	**55.51**

6.2 Comparison with Established Baselines

To address RQ2, Table 3 presents a comprehensive comparison of DDRO with baselines on the MS300K dataset. We can observe the followings: (i) The performance of dense retrieval baselines is generally better than that of sparse retrieval baselines, likely because the former uses dense vectors to capture richer semantic information, which is consistent with earlier findings [31, 32]. (ii) The best-performing dense retrieval baseline, ANCE, outperforms others such as SEAL, NCI, and DSI-QG. This could be due to the fact that these generative retrieval baselines rely solely on maximum likelihood estimation (MLE) to learn relevance, which may not fully capture the relevance patterns. However, ANCE lags behind models like Ultron, ROGER, and LTGR, which employ more advanced optimization strategies. This highlights the need for fine-grained relevance modeling to enhance generative retrieval ranking performance. (iii) Our proposed DDRO (TU) outperforms these generative retrieval baselines, achieving 15.63% and 8.03% higher R@1 and MRR@10, respectively, compared to the best-performing baseline, ROGER-Ultron. These results demonstrate the effectiveness of our document-level relevance optimization approach. While ROGER-Ultron achieves the highest R@10, DDRO (TU) delivers comparable performance within a more efficient and lightweight framework.

6.3 Ablation Study

To address RQ3, an ablation study was conducted to assess the impact of pairwise ranking optimization on DDRO performance. From Table 4, we can find: (i) On the MS MARCO dataset, removing pairwise ranking optimization significantly reduces intermediate and broader recall metrics (R@5 and R@10) for both DDRO variants, highlighting its critical role in improving retrieval performance. (ii) On the NQ dataset, excluding pairwise ranking optimization leads to consistent declines across all metrics, with a more pronounced impact on DDRO (PQ), particularly in early precision and broader recall. This underscores the importance of pairwise ranking in enhancing retrieval effectiveness at various ranking depths.

These findings confirm that pairwise ranking optimization effectively aligns model predictions with document relevance, contributing to improved performance across different ranking levels.

Effect of KL Constraint Strength β. We evaluate the impact of the β parameter, which controls the KL divergence constraint

Table 3: Comparison of retrieval model performance on the MS MARCO document ranking (MS300K) dataset. The best results are highlighted in bold. Statistical significance is assessed using a paired t-test with a $p < 0.05$ threshold, where improvements are marked with the dagger symbol (†) to indicate statistical significance. The second-best values are underlined. Results for cited models are sourced from their original papers. Abbreviations: SI – Semantic ID; PQ – Product Quantization; NG – N-grams; TU – Title + URL.

Model	R@1	R@5	R@10	MRR@10
Term-based retrieval				
BM25	18.94	42.82	55.07	29.24
DocT5Query	23.27	49.38	63.61	34.81
Dense retrieval				
DPR	29.08	62.75	73.13	43.41
ANCE	29.65	63.43	<u>74.28</u>	44.09
RepBERT	25.25	58.41	69.18	38.48
Sentence-T5	27.27	58.91	72.15	40.69
Generative retrieval				
DSI (SI)	25.74	43.58	53.84	33.92
DSI-QG (SI)	28.82	50.74	62.26	38.45
NCI (SI)	29.54	57.99	67.28	40.46
SEAL (NG)	27.58	52.47	61.01	37.68
Ultron (TU)	29.82	60.39	68.31	42.53
Ultron (PQ)	31.55	63.98	73.14	45.35
ROGER-NCI (SI) [77]	30.61	59.02	68.78	42.02
ROGER-Ultron (TU) [77]	33.07	63.93	**75.13**	46.35
MINDER (SI)	29.98	58.37	71.92	42.51
LTRGR (SI)	32.69	64.37	72.43	<u>47.85</u>
Ours				
DDRO (PQ)	32.92	64.36	73.02	45.76
DDRO (TU)	**38.24**†	**66.46**†	<u>74.01</u>	**50.07**†

between the DDRO policy π_θ and the reference policy π^{ref}, on retrieval performance. Figure 4 shows results for different β values on MS MARCO and Natural Questions (NQ). A moderate setting ($\beta = 0.4$) consistently yields the best MRR@10 across both datasets. Smaller values (e.g., $\beta = 0.2$) lead to under-regularization, resulting in unstable and suboptimal learning. In contrast, larger values (e.g., $\beta = 0.6$) impose excessive regularization, restricting the model's ability to adapt, and thus degrading performance. These results highlight the sensitivity of DDRO to the KL constraint, suggesting the importance of tuning β to balance learning flexibility and regularization.

6.4 Robustness Analysis Across Datasets

To address RQ4, additional experiments were conducted NQ dataset to evaluate the robustness of DDRO across datasets with varying characteristics. The analysis focuses on two aspects: (i) comparing DDRO retrieval performance with baseline models across different categories Table 5, and (ii) examining the impact of various docid design choices on retrieval effectiveness.

Comparison to baseline retrieval models. The performance comparison on the NQ320k across different retrieval baselines is as follow: (i) Term-based baselines, such as BM25 and DocT5Query,

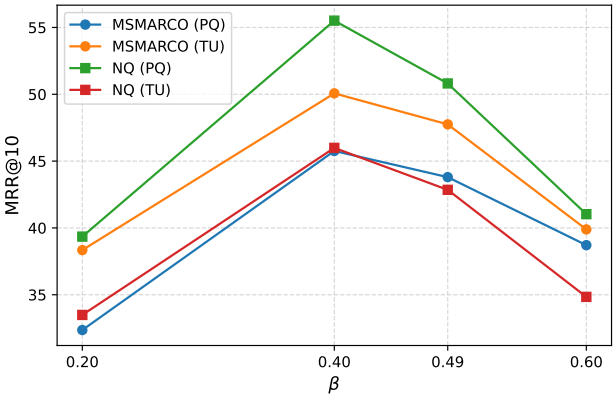


Figure 4: Effect of KL constraint strength (β) on DDRO performance. A moderate value ($\beta = 0.4$) yields the best MRR@10, while under- or over-regularization degrades performance.

Table 4: Ablation study evaluating the impact of pairwise ranking optimization on DDRO performance across the MS300K and NQ320K datasets. Statistical significance is assessed using a paired t-test with a significance threshold of $p < 0.05$. Statistically significant improvements ($p < 0.05$) are marked with a dagger symbol (†), while non-significant improvements are underlined. Abbreviations: PQ – Product Quantization; TU – Title + URL.

MS MARCO doc				
Model	**R@1**	**R@5**	**R@10**	**MRR@10**
DDRO (PQ)	<u>32.92</u>	64.36†	73.02†	<u>45.76</u>
w/o pairwise ranking	32.18	62.62	71.29	44.79
DDRO (TU)	<u>38.24</u>	66.46†	74.01†	<u>50.07</u>
w/o pairwise ranking	38.12	64.60	72.90	49.18
Natural Questions				
Model	**R@1**	**R@5**	**R@10**	**MRR@10**
DDRO (PQ)	48.92†	64.10†	67.31†	55.51†
w/o pairwise ranking	44.19	58.44	62.23	50.48
DDRO (TU)	40.86†	53.12†	55.98†	45.99†
w/o pairwise ranking	39.58	50.50	53.53	44.32

show lower early ranking performance, but remain competitive at broader levels. (ii) Dense retrieval baselines, including DPR and ANCE, improve early ranking metrics over term-based methods. (iii) Generative retrieval baselines, such as ROGER-Ultron (TU) and LTRGR, perform well across all metrics. The proposed DDRO achieves the highest overall performance. Specifically, DDRO (PQ) surpasses the best-performing baseline, ROGER-Ultron (TU), by 36.27% in R@1 and 23.58% in MRR@10.

Impact of Dataset Characteristics and Docid Selection. The docid design is a critical factor influencing performance in GenIR, we further analyze the differences in performance among various designs within our proposed DDRO. Our analysis underscores the critical impact of docid design on retrieval performance across datasets. DDRO (TU) excels on MS MARCO, where shorter, keyword-driven queries align well with title and URL-based docids that capture

Table 5: Comparison of retrieval model performance on the NQ320K dataset. The best-performing results are shown in bold. Statistical significance is determined using a paired t-test with a significance threshold of $p < 0.05$, with a dagger symbol (†) indicating statistical significance. Results for cited models are drawn from their respective original publications. Abbreviations used: SI – Semantic ID; PQ – Product Quantization; NG – N-grams; TU – Title + URL.

Model	R@1	R@5	R@10	MRR@10
Term-based retrieval				
BM25	14.06	36.91	47.93	23.60
DocT5Query	19.07	43.88	55.83	29.55
Dense retrieval				
DPR	22.78	53.44	68.58	35.92
ANCE	24.54	54.21	69.08	36.88
RepBERT	22.57	52.20	65.65	35.13
Sentence-T5	22.51	52.00	65.12	34.95
Generative retrieval				
DSI (SI)	27.42	47.26	56.58	34.31
DSI-QG (SI)	30.17	53.20	66.37	38.85
NCI (SI)	32.69	55.82	69.20	42.84
SEAL (NG)	29.30	54.12	68.53	40.34
Ultron (TU)	33.78	54.20	67.05	42.51
Ultron (PQ)	25.64	53.09	65.75	37.12
ROGER-NCI (SI) [77]	33.20	56.34	69.80	43.45
ROGER-Ultron (TU) [77]	35.90	55.59	**69.86**	44.92
MINDER (SI)	31.00	55.50	65.79	43.50
LTRGR (SI)	32.80	56.20	68.74	44.80
Ours				
DDRO (TU)	40.86	53.12	55.98	45.99
DDRO (PQ)	**48.92**†	**64.10**†	67.31	**55.51**†

surface-level lexical features for efficient retrieval. In contrast, DDRO (PQ) performs better on NQ, which features longer, complex queries requiring deeper semantic understanding. PQ-based docids effectively capture latent relationships, making them well-suited for NQ's informational queries. These findings suggest that aligning docid strategies with dataset-specific characteristics enhances retrieval effectiveness and model adaptability.

6.5 Analysis of Relevance Distribution

The relevance distribution could, to some extent, reflect the retrieval model's ability to recognize relevance. Therefore, we analyze the retrieval performance of DDRO at different top positions in the generated docid list to address RQ5. We have the following observations: (i) Tables (3) and (5) demonstrate DDRO's effectiveness across datasets, showcasing its adaptability. On MS MARCO, DDRO (TU) achieves the highest early precision, with statistically significant improvements in R@1 and MRR@10, and excels in intermediate ranking (R@5), outperforming models like ROGER-Ultron (TU) and LTRGR (SI). This highlights its ability to identify relevant documents early while maintaining strong intermediate performance. However, at broader recall levels (R@10), ROGER-Ultron (TU) shows a slight advantage over DDRO (TU). (ii) On NQ, DDRO (PQ) achieves the highest early precision (R@1, R@5), effectively ranking relevant documents for complex queries. The PQ-based docids capture deeper semantic relationships, leading to strong intermediate performance and competitive broader recall (R@10) against hybrid models such as ROGER-NCI (SI).

Overall, DDRO consistently achieves strong performance without relying on auxiliary reward models, reinforcement learning, or dense retrieval signals. Instead, it employs SFT followed by pairwise ranking optimization to refine early-stage precision while maintaining effectiveness across different ranking depths.

7 Conclusion

We introduced DDRO, a novel approach for enhancing GenIR systems by directly aligning docid generation with document-level relevance estimation. This alignment allows GenIR systems to effectively learn to rank and improve accuracy. We experimented with two types of docid designs and designed a lightweight direct L2R optimization algorithm on top of SFT training. Unlike existing RL-based methods, DDRO simplifies optimization through pairwise ranking, eliminating the need for auxiliary reward modeling and RL fine-tuning. Experiments conducted on benchmark datasets demonstrate that DDRO offers a lightweight optimization process while achieving high performance and demonstrating robustness. However, DDRO has limitations that offer opportunities for further improvement. (i) Our evaluation has primarily focused on a subset of the MS MARCO document dataset, leaving its scalability to larger and more diverse corpora, including domain-specific and multilingual datasets, yet to be explored. Addressing this limitation will be crucial to assess DDRO's adaptability in broader applications. (ii) Our current pairwise ranking formulation relies on binary relevance judgments (relevant vs. non-relevant), which may limit its expressiveness for queries with nuanced or graded relevance levels. Future work may explore listwise ranking objectives and graded relevance supervision to more accurately capture complex retrieval intents. (iii) Potential improvements could include advanced hard negative mining to enhance relevance discrimination and multi-objective optimization to balance relevance with efficiency, fairness, and diversity. (iv) A comprehensive comparison with scalable GenIR baselines such as RIPOR and PAG remains an avenue for future research to assess DDRO's scalability and integration potential in large-scale retrieval scenarios.

Acknowledgments

We thank our colleagues at the IRLab for their support and Hansi Zeng at UMass Amherst for helpful discussions and technical guidance.

Experiments for this work were supported by the Dutch Research Council (NWO) under project EINF-9550; computations were performed on the Snellius supercomputer (SURF). This research was (partially) supported by Ahold Delhaize, through AIRLab, by the Dutch Research Council (NWO), under project numbers 024.004.022, NWA.1389.20.183, and KICH3.LTP.20.006, and by the European Union's Horizon Europe program under grant agreement No 101070212.

All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

[1] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. https://doi.org/10.48550/ARXIV.2204.05862

[2] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional AI: Harmlessness from AI Feedback. *arXiv preprint arXiv:2212.08073* N/A, N/A (2022), N/A.

[3] 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* N/A, N/A (2016), N/A.

[4] Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Wen-tau Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive Search Engines: Generating Substrings as Document Identifiers. https://doi.org/10.48550/ARXIV.2204.10628

[5] 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* (Bonn, Germany) *(ICML '05)*. Association for Computing Machinery, New York, NY, USA, 89–96. https://doi.org/10.1145/1102351.1102363

[6] Christopher JC Burges. 2010. From RankNet to LambdaRank to LambdaMart: An Overview. *Learning* 11, 23-581 (2010), 81.

[7] Christopher J. C. Burges, Robert Ragno, and Quoc Viet Le. 2006. Learning to Rank with Nonsmooth Cost Functions. In *Proceedings of the 20th International Conference on Neural Information Processing Systems* (Canada) *(NIPS'06)*. MIT Press, Cambridge, MA, USA, 193–200.

[8] Jamie Callan. 1994. Passage-level Evidence in Document Retrieval. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Springer, N/A, Dublin, Ireland, 302–310.

[9] 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*. ACM, Corvallis, Oregon, USA, 129–136.

[10] Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. 2023. Continual Learning for Generative Retrieval over Dynamic Corpora. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. ACM, Birmingham, UK, 306–315.

[11] Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yiqun Liu, Yixing Fan, and Xueqi Cheng. 2023. A Unified Generative Retriever for Knowledge-Intensive Language Tasks via Prompt Learning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Taipei, Taiwan, 2372–2382. https://doi.org/10.1145/3539618.3591665

[12] Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2022. GERE: Generative Evidence Retrieval for Fact Verification. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Madrid, Spain, 2108–2112. https://doi.org/10.1145/3477495.3531864

[13] Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yiqun Liu, Yixing Fan, and Xueqi Cheng. 2022. CorpusBrain: Pre-train a Generative Retrieval Model for Knowledge-Intensive Language Tasks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. ACM, Atlanta, GA, USA, 199–208. https://doi.org/10.1145/3511808.3557475

[14] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep Reinforcement Learning from Human Preferences. *Advances in neural information processing systems* 30 (2017).

[15] Faïza Dammak, Hager Kammoun, and Abdelmajid Ben Hamadou. 2017. Improving Pairwise Learning to Rank Algorithms for Document Retrieval. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, Honolulu, HI, USA, 1–8. https://doi.org/10.1109/SSCI.2017.8285207

[16] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive Entity Retrieval. https://doi.org/10.48550/ARXIV.2010.00904

[17] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval. https://doi.org/10.48550/ARXIV.2109.10086

[18] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 2288–2292. https://doi.org/10.1145/3404835.3463098

[19] Muhammad Ibrahim and Mark Carman. 2016. Comparing Pointwise and Listwise Objective Functions for Random-Forest-based Learning-to-Rank. *ACM Transactions on Information Systems (TOIS)* 34, 4 (2016), 1–38.

[20] Natasha Jaques, Shixiang Gu, Dzmitry Bahdanau, José Miguel Hernández-Lobato, Richard E. Turner, and Douglas Eck. 2017. Sequence Tutor: Conservative Fine-tuning of Sequence Generation Models with KL-control. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70* (Sydney, NSW, Australia) *(ICML'17)*. JMLR.org, Sydney, NSW, Australia, 1645–1654.

[21] Natasha Jaques, Judy Hanwen Shen, Asma Ghandeharioun, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. 2020. Human-centric Dialog Training via Offline Reinforcement Learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 3985–4003. https://doi.org/10.18653/v1/2020.emnlp-main.327

[22] 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 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

[23] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc V. Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics* 7 (2019), 453–466. https://api.semanticscholar.org/CorpusID:86611921

[24] Yanyan Lan, Yadong Zhu, Jiafeng Guo, Shuzi Niu, and Xueqi Cheng. 2014. Position-aware ListMLE: A Sequential Learning Process for Ranking. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence* (Quebec City, Quebec, Canada) *(UAI'14)*. AUAI Press, Arlington, Virginia, USA, 449–458.

[25] Hyunji Lee, JaeYoung Kim, Hoyeon Chang, Hanseok Oh, Sohee Yang, Vladimir Karpukhin, Yi Lu, and Minjoon Seo. 2023. Nonparametric Decoding for Generative Retrieval. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 12642–12661. https://doi.org/10.18653/v1/2023.findings-acl.801

[26] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating Training Data Makes Language Models Better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 8424–8445. https://doi.org/10.18653/v1/2022.acl-long.577

[27] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

[28] Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2023. Multiview Identifiers Enhanced Generative Retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 6636–6648. https://doi.org/10.18653/v1/2023.acl-long.366

[29] Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2024. Learning to Rank in Generative Retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 8716–8723.

[30] Zihao Li, Zhuoran Yang, and Mengdi Wang. 2023. Reinforcement Learning with Human Feedback: Learning Dynamic Choices via Pessimism. https://doi.org/10.48550/ARXIV.2305.18438

[31] Shuqi Lu, Di He, Chenyan Xiong, Guolin Ke, Waleed Malik, Zhicheng Dou, Paul Bennett, Tie-Yan Liu, and Arnold Overwijk. 2021. Less is More: Pretrain a Strong Siamese Encoder for Dense Text Retrieval Using a Weak Decoder. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2780–2791. https://doi.org/10.18653/v1/2021.emnlp-main.220

[32] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, and Xueqi Cheng. 2022. Pre-train a Discriminative Text Encoder for Dense Retrieval via Contrastive Span Prediction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 848–858. https://doi.org/10.1145/3477495.3531772

[33] Sanket Vaibhav Mehta, Jai Gupta, Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Jinfeng Rao, Marc Najork, Emma Strubell, and Donald Metzler. 2023. DSI++: Updating Transformer Memory with New Documents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino,

and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 8198–8213.

[34] Donald Metzler, Yi Tay, Dara Bahri, and Marc Najork. 2021. Rethinking search: making domain experts out of dilettantes. In *ACM SIGIR Forum*, Vol. 55. ACM New York, NY, USA, 1–27.

[35] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2021. WebGPT: Browser-assisted Question-answering with Human Feedback. https://doi.org/10.48550/ARXIV.2112.09332

[36] Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-T5: Scalable Sentence Encoders from Pretrained Text-to-Text Models. In *Findings of the Association for Computational Linguistics: ACL 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 1864–1874. https://doi.org/10.18653/v1/2022.findings-acl.146

[37] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. https://doi.org/10.48550/ARXIV.1901.04085

[38] Rodrigo Nogueira and Jimmy Lin. 2019. From Doc2query to DocTTTTTquery. (2019). https://cs.uwaterloo.ca/~jimmylin/publications/Nogueira_Lin_2019_docTTTTTquery-v2.pdf.

[39] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training Language Models to Follow Instructions with Human Feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, Article 2011, 15 pages.

[40] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS)* (New Orleans, LA, USA) *(NeurIPS '23)*. Curran Associates, Inc., Red Hook, NY, USA, Article 2338, 14 pages. https://doi.org/10.48550/arXiv.2305.18290

[41] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21 (2020), 1–67.

[42] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Keshavan, Trung Vu, Lukasz Heidt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. 2023. Recommender Systems with Generative Retrieval. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '23)*. Curran Associates Inc., Red Hook, NY, USA, Article 452, 17 pages.

[43] Ruiyang Ren, Wayne Xin Zhao, Jing Liu, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2023. TOME: A Two-stage Approach for Model-based Retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 6102–6114. https://doi.org/10.18653/v1/2023.acl-long.336

[44] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (April 2009), 333–389. https://doi.org/10.1561/1500000019

[45] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference (TREC-3)*. National Institute of Standards and Technology (NIST), Gaithersburg, Maryland, USA, 109–126. https://www.microsoft.com/en-us/research/publication/okapi-at-trec-3/

[46] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. https://doi.org/10.48550/ARXIV.1707.06347

[47] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to Summarize from Human Feedback. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 253, 14 pages.

[48] 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. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '23)*. Curran Associates Inc., Red Hook, NY, USA, Article 2010, 17 pages.

[49] Hongyin Tang, Xingwu Sun, Beihong Jin, Jingang Wang, Fuzheng Zhang, and Wei Wu. 2021. Improving Document Representations by Generating Pseudo Query Embeddings for Dense Retrieval. 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)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 5054–5064. https://doi.org/10.18653/v1/2021.acl-long.392

[50] Yubao Tang, Ruqing Zhang, Jiafeng Guo, Jiangui Chen, Zuowei Zhu, Shuaiqiang Wang, Dawei Yin, and Xueqi Cheng. 2023. Semantic-Enhanced Differentiable Search Index Inspired by Learning Strategies. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, New York, NY, USA, 4904–4913.

[51] Yubao Tang, Ruqing Zhang, Jiafeng Guo, and Maarten de Rijke. 2023. Recent Advances in Generative Information Retrieval. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*. 294–297.

[52] Yubao Tang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, and Xueqi Cheng. 2024. Generative Retrieval Meets Multi-Graded Relevance. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 72790–72817. https://proceedings.neurips.cc/paper_files/paper/2024/file/853e781cb2af58956ed5c89aa59da3fc-Paper-Conference.pdf

[53] Yubao Tang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, and Xueqi Cheng. 2024. Listwise Generative Retrieval Models via a Sequential Learning Process. *ACM Transactions on Information Systems* 42, 5 (2024), 1–31.

[54] Yubao Tang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024. Bootstrapped Pre-training with Dynamic Identifier Prediction for Generative Retrieval. In *Findings of the Association for Computational Linguistics ACL 2024*. 10303–10317.

[55] Yubao Tang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Shihao Liu, Shuaiqing Wang, Dawei Yin, and Xueqi Cheng. 2025. Generative Retrieval for Book Search. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

[56] Yubao Tang, Ruqing Zhang, Zhaochun Ren, Jiafeng Guo, and Maarten de Rijke. 2024. Recent Advances in Generative Information Retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval)*. Association for Computing Machinery, New York, NY, USA, 3005–3008. https://doi.org/10.1145/3626772.3661379

[57] Yubao Tang, Ruqing Zhang, Zhaochun Ren, Jiafeng Guo, and Maarten de Rijke. 2024. Recent Advances in Generative Information Retrieval. In *Advances in Information Retrieval*, Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis (Eds.). Springer Nature Switzerland, Cham, 363–368.

[58] Yubao Tang, Ruqing Zhang, Weiwei Sun, Jiafeng Guo, and Maarten de Rijke. 2024. Recent Advances in Generative Information Retrieval. In *Companion Proceedings of the ACM Web Conference 2024* (Singapore, Singapore) *(WWW '24)*. Association for Computing Machinery, New York, NY, USA, 1238–1241. https://doi.org/10.1145/3589335.3641239

[59] Yubao Tang, Ruqing Zhang, Weiwei Sun, Jiafeng Guo, and Maarten de Rijke. 2024. Recent Advances in Generative Information Retrieval. In *Companion Proceedings of the ACM Web Conference 2024* (Singapore, Singapore) *(The Web Conference 2024)*. Association for Computing Machinery, New York, NY, USA, 1238–1241. https://doi.org/10.1145/3589335.3641239

[60] Yi Tay, Vinh Q. Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. 2022. Transformer Memory as A Differentiable Search Index. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, Article 1587, 13 pages.

[61] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. https://doi.org/10.48550/ARXIV.2302.13971

[62] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. TRL: Transformer Reinforcement Learning. https://github.com/huggingface/trl.

[63] 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* (Torino, Italy) *(CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 1313–1322. https://doi.org/10.1145/3269206.3271784

[64] Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Hao Sun, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, Xing Xie, Hao Allen Sun, Weiwei Deng, Qi Zhang, and Mao Yang. 2022. A Neural Corpus Indexer for Document Retrieval. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, Article 1856, 15 pages.

[65] Zihan Wang, Yujia Zhou, Yiteng Tu, and Zhicheng Dou. 2023. NOVO: Learnable and Interpretable Document Identifiers for Model-Based IR. In *Proceedings of the*

32nd ACM International Conference on Information and Knowledge Management (Birmingham, United Kingdom) *(CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 2656–2665. https://doi.org/10.1145/3583780.3614993

[66] Ronald J. Williams and David Zipser. 1989. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation* 1 (1989), 270–280.

[67] 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* (Helsinki, Finland) *(ICML '08)*. Association for Computing Machinery, New York, NY, USA, 1192–1199. https://doi.org/10.1145/1390156.1390306

[68] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. https://doi.org/10.48550/ARXIV.2007.00808

[69] Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. Pretrained Transformers for Text Ranking: BERT and Beyond. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*, Greg Kondrak, Kalina Bontcheva, and Dan Gillick (Eds.). Association for Computational Linguistics, Online, 1–4.

[70] 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 Web Conference 2024* (Singapore, Singapore) *(WWW '24)*. Association for Computing Machinery, New York, NY, USA, 1441–1452. https://doi.org/10.1145/3589334.3645477

[71] Hansi Zeng, Chen Luo, and Hamed Zamani. 2024. Planning Ahead in Generative Retrieval: Guiding Autoregressive Generation through Simultaneous Decoding. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 469–480. https://doi.org/10.1145/3626772.3657746

[72] 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* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 1979–1983. https://doi.org/10.1145/3477495.3531791

[73] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing Dense Retrieval Model Training with Hard Negatives. https://doi.org/10.48550/ARXIV.2104.08051

[74] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. RepBERT: Contextualized Text Embeddings for First-Stage Retrieval. https://doi.org/10.48550/ARXIV.2006.15498

[75] Peitian Zhang, Zheng Liu, Yujia Zhou, Zhicheng Dou, Fangchao Liu, and Zhao Cao. 2023. Generative Retrieval via Term Set Generation. https://doi.org/10.48550/ARXIV.2305.13859

[76] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2023. Enhancing Generative Retrieval with Reinforcement Learning from Relevance Feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 12481–12490.

[77] Yujia Zhou, Jing Yao, Zhicheng Dou, Yiteng Tu, Ledell Wu, Tat-Seng Chua, and Ji-Rong Wen. 2024. ROGER: Ranking-Oriented Generative Retrieval. *ACM Trans. Inf. Syst.* 42, 6, Article 155 (Oct. 2024), 25 pages. https://doi.org/10.1145/3603167

[78] Yujia Zhou, Jing Yao, Zhicheng Dou, Ledell Wu, and Ji-Rong Wen. 2022. DynamicRetriever: A Pre-training Model-based IR System with Neither Sparse nor Dense Index. https://doi.org/10.48550/ARXIV.2203.00537

[79] Yujia Zhou, Jing Yao, Zhicheng Dou, Ledell Wu, Peitian Zhang, and Ji-Rong Wen. 2022. Ultron: An Ultimate Retriever on Corpus with a Model-based Indexer. https://doi.org/10.48550/ARXIV.2208.09257

[80] Shengyao Zhuang, Houxing Ren, Linjun Shou, Jian Pei, Ming Gong, Guido Zuccon, and Daxin Jiang. 2022. Bridging the Gap Between Indexing and Retrieval for Differentiable Search Index with Query Generation. https://doi.org/10.48550/ARXIV.2206.10128

[81] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-Tuning Language Models from Human Preferences. https://doi.org/10.48550/ARXIV.1909.08593

[82] Noah Ziems, Wenhao Yu, Zhihan Zhang, and Meng Jiang. 2023. Large Language Models are Built-in Autoregressive Search Engines. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 2666–2678. https://doi.org/10.18653/v1/2023.findings-acl.167