Reinforcement Learning-Driven Generative Retrieval with Semantic-aligned Multi-Layer Identifiers

Bo Xu
xubo@dlut.edu.cn
School of Computer Science and Technology, Dalian University of Technology
Dalian, Liaoning, China

Yicen Tian
yicentian@mail.dlut.edu.cn
School of Computer Science and Technology, Dalian University of Technology
Dalian, Liaoning, China

Xiaokun Zhang
dawnkun1993@gmail.com
City University of Hong Kong
Hong Kong, China

Erchen Yu
yuerchen0809@mail.dlut.edu.cn
School of Computer Science and Technology, Dalian University of Technology
Dalian, Liaoning, China

Dailin Li
ldlbest@mail.dlut.edu.cn
School of Computer Science and Technology, Dalian University of Technology
Dalian, Liaoning, China

Linlin Zong*
llzong@dlut.edu.cn
School of Software, Dalian University of Technology
Dalian, Liaoning, China

Hongfei Lin
hflin@dlut.edu.cn
School of Computer Science and Technology, Dalian University of Technology
Dalian, Liaoning, China

Abstract

Generative retrieval enhances retrieval effectiveness by generating natural language represented document identifiers. However, current methods often struggle with two major challenges: limited identifier quality and insufficient query-document interaction, leading to limited retrieval performance. To tackle these challenges, we propose a novel generative retrieval framework integrated with semantic-aligned multi-layer identifiers and reinforcement learning. To improve identifier quality, we design a prompt-driven multi-task learning strategy to generate three types of hierarchical identifiers: summary, keyword, and pseudo-query, to capture multi-granularity document semantics. Furthermore, we adopt supervised fine-tuning to integrate these identifiers. To improve query-document interaction, we devise a multi-view ranking fusion mechanism that combines retrieval results across multi-layer identifiers. We further employ a GRPO-based reinforcement learning based on dense similarity rewards and a difficulty-aware negative sampling strategy to optimize the generated identifiers. Experiments on multiple benchmark datasets show that our framework significantly outperforms existing generative retrieval methods, offering a promising solution for building more effective and semantically aligned retrieval systems. The code for our model is publicly available at https://github.com/yicentian02/GRAM-RL.

CCS Concepts

• **Information systems** → **Retrieval models and ranking**.

Keywords

Generative Retrieval; Muti-layer Identifier; Reinforcement Learning

ACM Reference Format:
Bo Xu, Yicen Tian, Xiaokun Zhang, Erchen Yu, Dailin Li, Linlin Zong, and Hongfei Lin. 2025. Reinforcement Learning-Driven Generative Retrieval with Semantic-aligned Multi-Layer Identifiers. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746252.3761136

1 Introduction

Information retrieval plays a vital role in web applications, such as search ranking [20], question answering [3], and retrieval-augmented generation [13]. Traditional retrieval paradigms include the sparse retrieval based on the bag-of-words assumption (e.g., BM25 [23]), and the dense retrieval leveraging semantic embeddings (e.g., DPR [9]). However, both paradigms rely on independent stages of representation and retrieval, which largely limit retrieval performance.

Recently, generative retrieval (GR) has redefined the retrieval process as an end-to-end paradigm that directly maps queries to document identifiers (DocIDs) [6, 27]. The core challenge of this paradigm lies in the design and optimization of DocIDs [1, 34]. By eliminating the traditional multi-stage pipeline of indexing and ranking, GR enables a single-step generation process from a query to its target documents, significantly improving retrieval performance [1, 5, 14, 16, 31]. Recent studies have developed a variety of DocID representation methods, such as numeric identifiers [27, 34], textual identifiers [6, 26, 31], and semantic identifiers [14, 27, 29].

Although these methods have achieved advanced retrieval performance by bypassing explicit query-document matching, generative retrieval still confronts two major challenges: **limited identifier quality** and **insufficient query-document interaction**.

[*] Corresponding author.

CIKM '25, Seoul, Republic of Korea

ACM ISBN 979-8-4007-2040-6/2025/11
https://doi.org/10.1145/3746252.3761136

For *limited identifier quality*, existing methods often rely on single-mode or shallow semantic extraction to generate DocIDs with limited semantic understanding [1, 27, 34, 35]. This results in identifiers that do not fully capture the complex semantics and context of the documents. As a consequence, these low-quality identifiers may mislead the retrieval process. When users issue queries, the system may retrieve documents that are only superficially related to the query, thereby missing most relevant documents. Enhancing identifier quality is a crucial step for a retrieval system to understand in-depth document content and match it with queries for providing the most relevant search results. Although some recent methods have attempted to enhance identifier quality by incorporating titles, term sets, body substrings and urls [1, 14, 31, 34], the generated identifiers still lack sufficient contextual awareness to serve as robust semantic representations.

For *insufficient query-document interaction*, most existing methods treat query-document interaction as a one-way or weak connection, neglecting hierarchical and fine-grained interactions between queries and documents [1, 11, 14, 28, 28, 31]. Since most models rely on identifier generation probabilities to reflect relevance, they struggle to capture the combined contribution of multiple semantic fragments in long documents or ambiguous queries. This underscores the need for retrieval strategies that can explicitly model multi-dimensional query-document interactions beyond the identifier mediation. For instance, the generated identifiers may be too generic and lack the ability to specifically respond to the query intent. Even if the system retrieves documents based on the generated identifiers, it may not be able to accurately prioritize the most relevant documents. Recent models, such as ROGER [33] and DOGR [17], have attempted to enhance query-document interaction by incorporating dense retriever feedback and contrastive ranking loss. However, these models rely heavily on external retrieval signals, which results in incomplete alignment between query intent and document semantics.

To address these two challenges, we propose a novel Generative Retrieval framework with semantic-Aligned Multi-layer identifiers via Reinforcement Learning (GRAM-RL). To improve *semantic richness and contextual coverage of DocIDs*, GRAM-RL first establishes three hierarchical identifier layers, namely summary, keyword, and pseudo-query, to capture query topics, fine-grained semantic details, and latent user intents, respectively. To jointly generate three-layer identifiers, GRAM-RL fine-tunes a unified pre-trained language model via a prompt-driven multi-task learning strategy, enhancing the semantic and contextual relevance of DocIDs. Furthermore, a multi-layer fusion mechanism is proposed to combine the isolated identifiers by constructing separate vector spaces from document titles, bodies, and generated identifiers. This mechanism then integrates the vector spaces via a multi-view ranking fusion function based on position decay, view-specific weighting, and paragraph frequency. Addressing the challenge of limited identifier quality, GRAM-RL substantially improves the accuracy and robustness of retrieval in our experiments.

To model *fine-grained query-document interactions* in DocIDs, GRAM-RL introduces a Group Relative Policy Optimization (GRPO)-based reinforcement learning method for generating discriminative DocIDs. Tailored for retrieval scenarios, the reward function is designed to maximize the average margin between vector similarities of positive and negative documents relative to the generated identifiers, thereby guiding the model to produce semantically aligned DocIDs. Furthermore, a difficulty-aware negative sampling strategy is integrated, which strategically incorporates a balanced mixture of hard, medium, and easy negative examples. This strategy strengthens contrastive learning by forcing the model to discern subtle semantic differences, thereby enhancing its ability to capture fine-grained distinctions and improving the overall discriminative power of the learned DocIDs.

In summary, our contributions are as follows:

- We propose a novel generative retrieval framework GRAM-RL via semantic-aligned multi-layer identifiers and reinforcement learning to generate high-quality fine-grained DocIDs for advanced retrieval performance.
- We design a prompt-driven multi-task fine-tuning for DocID generation, a multi-view ranking fusion for DocID integration, and a GRPO-based reinforcement optimization for query-document interaction-aware DocID learning.
- Extensive experiments demonstrate that GRAM-RL outperforms state-of-the-art models, and effectively improves the quality of document identifiers, providing a more efficient and reliable solution for generative retrieval.

2 The Proposed Framework

2.1 Overview of the GRAM-RL

As illustrated in Figure 1, the proposed GRAM-RL framework consists of two core components: (1) Semantic-aligned multi-layer identifier learning employs supervised fine-tuning process to generate hierarchical high-quality document identifiers, capturing query topics, fine-grained semantics, and latent user intents; (2) Multi-view interaction-aware identifier optimization utilizes GRPO-based reinforcement learning to incorporate fine-trained query-document interactions into identifiers for better relevance matching.

2.2 Semantic-aligned Multi-layer Identifier Learning

2.2.1 *Identifier Creation.*

To improve the quality of initially generated identifiers, we propose to extract three-layer identifiers for each document, namely a summary, a set of keywords, and a set of pseudo-queries. These identifiers capture the document semantics at multiple granularities for document representation and retrieval.

Top-layer Summary Identifier. This type of DocIDs consists of a concise document summary s with no more than ten words, which serves as a high-level representation that captures the core theme of the document, acting as the top-layer identifier in our framework.

Middle-layer Keyword Identifier. This type of DocIDs involves a set of more representative and discriminative keywords extracted from the summary and expanded using LLMs, enabling effective paragraph-level semantic matching.

Bottom-layer Pseudo-query Identifier. This type of DocIDs involves a set of query-like questions generated by recombining keywords and prompting with an LLM, serving as potential user queries. Pseudo-queries project document semantics into the query

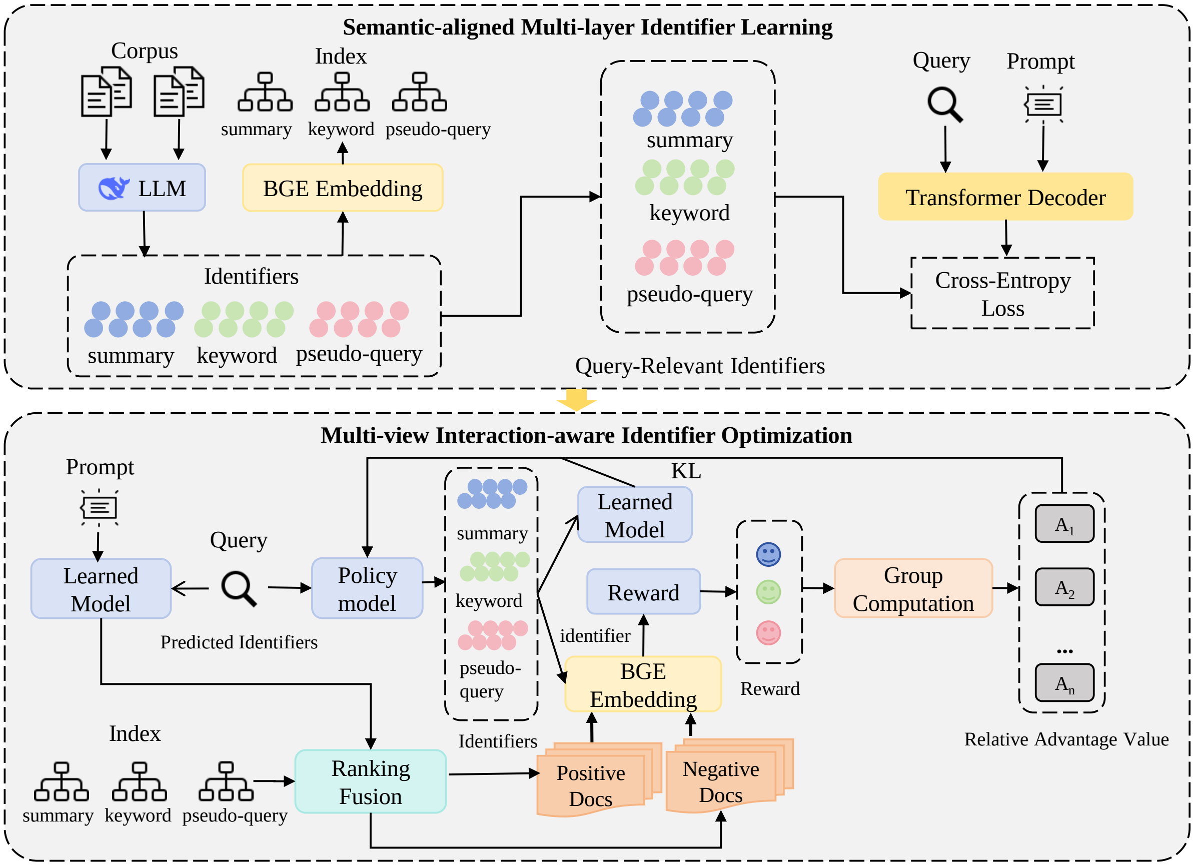

Figure 1: Architecture of the GRAM-RL framework with two core components. Semantic-aligned multi-layer identifier learning generates hierarchical document identifiers via supervised fine-tuning to capture query topics, fine-grained semantics, and latent user intents. Multi-view interaction-aware identifier optimization employs GRPO-based reinforcement learning to integrate fine-grained query-document interactions into identifiers, optimizing their relevance for retrieval.

space, simulating realistic information needs and enhancing retrieval diversity. To illustrate the three-layer identifiers, we provide an example in Table 1. By extracting these identifiers, document semantics are better aligned for relevance ranking.

2.2.2 Identifier Learning via Supervised Fine-tuning. To accurately fit DocIDs with semantic information, we fine-tuned a pre-trained auto-regressive language model to generate each type of identifiers based on a given query q. We crafted task-specific prompts for generating the gold-standard summary, keyword, and pseudo-query with respect to each document. Specifically, given a training sample (q, s^*, k^*, z^*), where q is the input query, and s^*, k^*, and z^* are the corresponding high-quality generated summary, keywords, and pseudo-queries extracted from highly relevant documents, we construct three input-output training pairs: (x_s, y_s), (x_k, y_k), and (x_z, y_z), where x_* is the constructed prompt and y_* is the expected output. We train the generation model $f(\cdot)$ using the auto-regressive cross-entropy loss[6]. For each target sequence $y = (y_1, y_2, \ldots, y_T)$ and its corresponding input x, the loss is computed as:

$$\mathcal{L}_{\text{CE}}(f(x), y) = -\sum_{t=1}^{T} \log P(y_t \mid y_{<t}, x; \theta), \qquad (1)$$

where $P(y_t \mid y_{<t}, x; \theta)$ is the probability assigned by the model f with parameters θ to the token y_t given the previous tokens $y_{<t}$ and the input x. This encourages the model to generate summaries, keywords, and pseudo-queries that are close to the gold-standard references under a unified auto-regressive learning framework.

2.3 Multi-view Interaction-aware Identifier Optimization

2.3.1 Multi-view Retrieval. To enhance relevance recall and retrieval robustness, we propose a Hierarchical Ranking Fusion (HRF) mechanism that leverages the complementary semantic coverage of multiple identifiers. This mechanism retrieves documents from multiple indexes, and fuses mlutiple ranking results into a unified ranking.

Identifier-Guided Query Expansion. For a given query q, we concatenate it with each type of identifiers (summary s, keywords k

Table 1: An example on three-layer document identifiers.

Query: Who was the British Prime Minister in 1953?
Document (https://en.wikipedia.org/wiki/Winston_Churchill) **title:** Winston Churchill **text:** Winston Churchill Sir Winston Leonard Spencer-Churchill (30 November 187424 January 1965) was a British politician, statesman, army officer, and writer, who was Prime Minister of the United Kingdom from 1940 to 1945 and again from 1951 to 1955. As Prime Minister, Churchill led Britain...
Multi-layer Identifiers **Summary:** Winston Churchill: WWII British Prime Minister and writer. **keywords:** Winston Churchill, British Prime Minister, WWII, writer, Conservative Party, Member of Parliament, statesman, army officer, economic liberal, British imperialist **Pseudo-queries:** - Who was Winston Churchill? - Winston Churchill writings? - What was Winston Churchill's role during World War II? - Which political parties did Winston Churchill belong to? - When did Winston Churchill serve as Prime Minister of the UK?

and pseudo-queries z), to construct three expanded queries:

$$q_s = [q; s] \tag{2}$$

$$q_k = [q; k] \tag{3}$$

$$q_z = [q; z] \tag{4}$$

These three forms of expanded queries are used independently to retrieve documents from three-view vector indexes.

Multi-View Index Construction. To match the semantic characteristics of each identifier type, we construct three dense retrieval vector indexes with tailored document encodings: (1) Summary-view index, where documents are encoded as [Title; Text; Summary]; (2) Keyword-view index, where documents are encoded as [Title; Text; Keywords]; (3) Pseudo-query-view index, where documents are encoded as [Title; Text; Pseudo-queries]. Each index is built using a pre-trained embedding model, ensuring compatibility between identifier queries and document vectors.

Hierarchical Ranking Fusion. Once retrieval is performed on all three-view indexes, we obtain three ranked lists: $R_s(q)$, $R_k(q)$, and $R_z(q)$, corresponding to q_s, q_k, and q_z respectively. These lists are merged using a frequency-aware weighted reciprocal rank fusion function for each document d, computed as follows.

$$S(q, d) = \sum_{t \in \{s,k,z\}} w_t \cdot \frac{\mathbb{I}[r_t(q, d) \text{ exists}]}{\log_2(1 + r_t(q, d))} \cdot f_{q,d} \tag{5}$$

where $r_t(q, d)$ is the rank position of d under view t, if it appears (1-based indexing). w_t is the learned or heuristic weight for view t. $\mathbb{I}[\cdot]$ is the indicator function, ensuring documents not retrieved in view t do not contribute. $f_{q,d} = \sum_t \mathbb{I}[r_t(q, d) \text{ exists}]$ is the appearance frequency of document d across views. This scoring function ensures higher-ranked documents contribute more due to reciprocal log decay, and view weights w_t enable prioritizing more reliable identifier types. This fusion-based framework unifies multiple semantic perspectives of the query, allowing the system to recall documents that may match one or more facets of the intended meaning. It is particularly effective for complex or under-specified queries, where a single identifier may be insufficient.

2.3.2 Query-document Interaction-aware Reinforcement Learning. Relying solely on supervised signals to generate positive identifiers is often insufficient for training a model with strong recall capabilities. In particular, under weakly supervised settings, unstable identifier quality may frequently lead to the retrieval of irrelevant documents.Therefore, to further capture the query-document interactions, we introduce the Group Relative Policy Optimization (GRPO)-based reinforcement learning [24] to encode interaction-aware information into the generated DocIDs, thereby enhancing the model's practical utility in retrieval tasks.

Specifically, for each query q_i in the training set, we use the current generation model π_θ to generate one candidate identifier z_i, and employ the learned retrieval model to retrieve a ranked list of documents $P_i = \{p_i^1, p_i^2, \ldots, p_i^3\}$. Each P_i includes both relevant (positive) documents closely related to the query, as well as irrelevant or incorrectly retrieved (negative) documents. Our learning aims to encourage the model to generate identifiers that prioritize retrieving positive documents. To this end, we define a reward function based on vector similarity to measure the retrieval quality of each generated identifier:

$$r_i = \lambda \cdot \left(\text{Sim}_{\text{pos}}(z_i) - \text{Sim}_{\text{neg}}(z_i)\right), \tag{6}$$

$$\text{Sim}_{\text{pos}}(z_i) = \frac{1}{|P_i^+|} \sum_{p \in P_i^+} \text{Sim}(z_i, p), \tag{7}$$

$$\text{Sim}_{\text{neg}}(z_i) = \frac{1}{|P_i^-|} \sum_{p \in P_i^-} \text{Sim}(z_i, p) \tag{8}$$

where $\text{Sim}(z, p)$ denotes the cosine similarity computed using the BGE-v1.5[30] embedding model. P_i^+ and P_i^- represent the sets of positive and negative documents in the retrieved list, respectively. λ is a reward scaling factor, typically set to 100.0.

Stratified Construction of Positive and Negative Samples. To ensure the reward signal is stable and discriminative, we use stratified sampling to form positive (P_i^+) and negative (P_i^-) sets. For P_i^+, we select top-ranked retrieved documents that are either relevant or contain answers, ensuring semantic alignment with the query. For P_i^-, we craft a multi-level negative set: hard negatives with high similarity but irrelevance, medium negatives related to the topic but lacking answers, and easy negatives fully unrelated to the query. Sampling from each negative category by predefined ratios, such as 30% hard, 40% medium, and 30% easy, enhances the model's capability to handle difficult cases and avoid semantic confusion, with the ratio determined empirically through grid search experiments.

Advantage Estimation and Policy Update. To reduce variance during training, we adopt intra-group normalized relative advantage estimation in GRPO. For a group of generated identifiers, denoted as G, we normalize each reward r_i using:

Table 2: Performance comparisons on NQ and TrivaQA. All results are from our own implementation.

Category	**Method**	**NQ**				**TrivaQA**			
		R@1	R@5	R@10	MRR@10	R@1	R@5	R@10	MRR@10
Sparse	BM25	0.266	0.511	0.602	0.371	0.466	0.647	0.703	0.543
	DocT5Query	0.383	0.619	0.693	0.484	0.493	0.708	0.766	0.586
Dense	DPR	0.449	0.631	0.683	0.528	0.437	0.612	0.679	0.512
	Sentence-T5	0.364	0.588	0.655	0.457	0.442	0.659	0.721	0.534
	GTR-base	0.419	0.636	0.694	0.510	0.510	0.705	0.764	0.595
	BGE	0.452	0.659	0.720	0.541	0.504	0.701	0.754	0.587
Generative	SEAL	0.246	0.476	0.562	0.343	0.250	0.437	0.525	0.329
	MINDER	0.352	0.536	0.613	0.499	0.377	0.570	0.642	0.459
	DGR	0.423	0.597	0.656	0.499	0.460	0.646	0.703	0.539
	LTRGR	0.431	0.616	0.670	0.509	0.460	0.658	0.715	0.543
Ours	**GRAM-RL**	**0.495**	**0.687**	**0.736**	**0.576**	**0.600**	**0.752**	**0.792**	**0.665**

$$\hat{r}_i = \frac{r_i - \mu_r}{\sigma_r}, \quad \text{where} \quad \mu_r = \frac{1}{G}\sum_{j=1}^{G} r_j, \quad \sigma_r = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(r_j - \mu_r)^2}. \tag{9}$$

We treat the normalized reward $\hat{r}_i$ as a shared advantage value $A_{i,t} = \hat{r}_i$ for all tokens in the generated sequence z_i, and construct the GRPO loss as:

$$\begin{aligned}\mathcal{L}_{\text{GRPO}} = \mathbb{E}_{(q,z_i)}\Big[&\min\left(\rho_i \cdot \hat{r}_i,\ \text{clip}(\rho_i, 1-\epsilon, 1+\epsilon)\cdot \hat{r}_i\right)\\ &- \beta \cdot \text{KL}(\pi_\theta \| \pi_{\text{ref}})\Big]\end{aligned} \tag{10}$$

where $\rho_i = \frac{\pi_\theta(z_i|q)}{\pi_{\theta_{\text{old}}}(z_i|q)}$ is the policy ratio, ϵ is a clipping threshold, and π_{ref} is a frozen reference model after SFT. Using this GRPO-based reinforcement learning, fine-grained query-document interactions are encoded into DocIDs, enriching the identifiers with stronger contextual semantics for more effective retrieval.

3 Experiment

3.1 Research Questions

To evaluate the performance of our GRAM-RL framework, we focus on answering the following research questions:

- RQ1: How does our GRAM-RL perform compared with existing state-of-the-art methods?
- RQ2: What is the effect of each designed component in handling the task?
- RQ3: How do negative sample selection strategies affect retrieval performance?
- RQ4: Why is the hierarchical ranking fusion important for improving retrieval performance?
- RQ5: How well does the proposed framework work in real-world instances?

3.2 Datasets and Preprocessing

We evaluate our model on three widely used benchmarks: Natural Questions (NQ) [10], TriviaQA [8], and BEIR NFCorpus [2]. NQ contains 140k documents and 30k queries from real Google search logs, while TriviaQA includes 220k documents and 35k queries from Wikipedia. NFCorpus is a biomedical dataset with 11,823 documents and 2,591 queries. For evaluation, we mix relevant and irrelevant documents (top-10 BM25 candidates) to create a challenging setup. Additionally, we use DeepSeek-v3 [7] to generate <= 10-word summaries for each document, from which we extract up to 10 keywords and construct five pseudo-queries to capture user intent.We choose DeepSeek-v3 for its strong summarization and keyword generation capability.

3.3 Baselines and Metrics

We compare our approach against a range of generative retrieval methods, including SEAL [1], MINDER [14], LTRGR [15], DGR [16], GENRE [6], and GLEN [11]. In addition, we include traditional sparse retrieval baselines such as BM25 [23] and DocT5Query [21], as well as dense retrieval methods including DPR [9], SentenceT5 [18], and GTR-base [19]. Most baseline results are reproduced using publicly available implementations, while others are reported directly from their respective papers when models are not released.

Following previous works, we adopt widely used retrieval metrics, including Recall@1, Recall@5, Recall@10, and MRR@10 for NQ and TriviaQA, and NDCG@10, Recall@5, Recall@20, and Recall@100 for NFCorpus, with cutoff values chosen based on dataset size and query characteristics to balance top-ranked precision and overall recall.

3.4 Implementation Details

All experiments are conducted on a single NVIDIA A800 GPU (80GB). For model configuration, we use Qwen2.5-7B-Instruct [22] as our main model, which uses the Transformers framework. It has a learning rate of 5×10^{-5} and uses the Adam optimizer with a weight decay of 0.01. The batch size is set at 64. Its LoRA rank/Alpha

Table 3: Performance comparisons on NFCorpus. The methods marked with † are from their official implementation; others are from our implementation. Results not available are denoted as '–'.

Method	NDCG@10	R@5	R@20	R@100
BM25	0.311	0.100	0.164	0.204
DocT5Query	0.305	0.103	0.149	0.198
Sentence-T5	0.295	0.119	0.167	0.209
GTR-base	0.315	0.118	0.171	0.211
BGE	0.366	0.137	0.199	0.256
SEAL	0.239	0.065	0.145	0.194
MINDER	0.280	0.064	0.151	0.207
GENRE†	0.200	–	–	–
GLEN†	0.159	–	–	–
GRAM-RL (Ours)	**0.373**	**0.144**	**0.220**	**0.270**

values are 64/16. For fair comparison, we additionally train a BART-large [12] model that generates only hierarchical identifiers, which utilizes the Fairseq framework. The learning rate is set at 3×10^{-5}, and the optimizer employed is Adam with a weight decay of 0.01. The maximum number of tokens is specified as 4192. For BART large inference, the beam size is set to 5,15,15 for NFCorpus, NQ, TrivaQA, respectively. For GRPO-based reinforcement learning, We apply GRPO with up to 10 top-ranked positives per query. For negative sampling, we adopt a 3-level difficulty strategy, selecting 3 hard, 4 medium, and 3 easy negatives per query. GRPO is trained with a learning rate of 1×10^{-5}, batch size 256, LoRA rank 64, $\alpha = 16$, and weight decay 0.01.

4 Results and Analysis

4.1 Overall Performance (RQ1)

We summarize our main experimental results on NQ, TriviaQA in Table 2, and NF-Corpus in Table 3, where bolded numbers are the best performance of each column and the second best method is underlined. The following findings highlight the effectiveness of our proposed generative retrieval framework.

Our method significantly outperforms existing generative retrieval models across all datasets. On NQ and TriviaQA, GRAM-RL improves MRR@10 by +6.7% and +12.2% over LTRGR, and on NFCorpus, it surpasses MINDER by +9.3% in NDCG@10. While prior models use shallow document representations, our framework generates structured identifiers that encode high-level semantic information, aligning better with diverse query intents. GRAM-RL's superior performance across benchmarks demonstrates the benefits of structured identifier generation. LTRGR uses static relevance labels and DGR distills preferences from teacher models, lacking dynamic feedback. In contrast, GRPO provides fine-grained rewards based on relative similarity differences, bridging identifier generation and ranking effectiveness.

Traditional GR models face limitations with constrained auto-regressive decoding and beam search, which can hinder recall performance. Our method employs vector-based similarity for flexible, non-autoregressive generation, enhancing recall and robustness to distributional variance. GRAM-RL consistently outperforms sparse and dense baselines. On NQ, it improves MRR@10 over GTR-base by +6.6%, and on TriviaQA, achieves +9.0% gain in Recall@1. On NFCorpus, it surpasses BGE in NDCG@10 by +0.7% and Recall@100 by +1.4%, demonstrating strong generalization and promises for end-to-end Retrieval-Augmented Generation (RAG) applications.

Table 4: Ablation studies on NFCorpus.

	ndcg@10	R@5	R@20	R@100
GRAM-RL	**0.373**	**0.144**	**0.220**	**0.270**
w/o GRPO	0.335	0.135	0.204	0.259
w/o LLM+GRPO	0.312	0.125	0.199	0.259
w/o keyword+GRPO	0.295	0.119	0.188	0.254
w/o query+GRPO	0.305	0.130	0.196	0.258
w/o summary+GRPO	0.312	0.121	0.197	0.258

Table 5: Ablation studies on NQ.

	R@1	R@5	R@10	MRR@10
GRAM-RL	**0.495**	**0.687**	**0.736**	**0.576**
w/o GRPO	0.486	0.683	0.735	0.571
w/o LLM+GRPO	0.462	0.667	0.726	0.548
w/o query+GRPO	0.465	0.683	0.731	0.559
w/o keyword+GRPO	0.464	0.681	0.733	0.553
w/o summary+GRPO	0.456	0.671	0.724	0.548

Table 6: Ablation studies on TivaQA.

	R@1	R@5	R@10	MRR@10
GRAM-RL	**0.600**	**0.752**	**0.792**	**0.665**
w/o GRPO	0.572	0.748	**0.792**	0.647
w/o LLM+GRPO	0.535	0.718	0.769	0.613
w/o query+GRPO	0.560	0.745	0.789	0.640
w/o keyword+GRPO	0.543	0.748	0.788	0.632
w/o summary+GRPO	0.556	0.740	0.785	0.635

4.2 The Effect of Component Design (RQ2)

We conduct detailed ablation studies to investigate the impact of different components during the training and inference stages. The ablation is performed on the training phase using our framework, with or without GRPO and with different backbone models. Specifically, we explore the following variants: (1) w/o keyword+GRPO: The keyword generation module is removed during the identifier generation stage and the GRPO module is removed. (2) w/o query+GRPO: The pseudo-query generation module is removed and the GRPO module is ommited. (3) w/o summary+GRPO: The summary generation module is removed and the GRPO module is dropped. (4) w/o GRPO: The reinforcement learning module (GRPO) is removed. (5) w/o LLM+GRPO: The model backbone is replaced with BART-large instead of a LLM, and the GRPO training stage is omitted. The results are shown in Table 4, Table 5 and Table 6.

Removing any single identifier view consistently degrades performance. On NQ, excluding summary, keyword, and pseudo-query causes Recall@1 to drop by 3.0%, 2.2%, and 2.1%, respectively. NF-Corpus shows NDCG@10 decreases of 2.3%, 4.0%, and 3.0%, and TriviaQA exhibits 1.6%, 2.9%, and 1.2% drops in Recall@1. Each view contributes unique information: summary captures document semantics, keywords highlight core concepts, and pseudo-queries simulate user queries. Combining all three provides richer, more diverse information than any two, validating the necessity of multi-view identifiers in generative retrieval.

Dataset sensitivity varies by identifier view. On QA-oriented NQ, summary contributes most by condensing answer-focused content. NF-Corpus benefits more from pseudo-queries, likely due to capturing diverse matching paths. On TriviaQA, keyword removal leads to the largest Recall@1 drop, reflecting the effectiveness of concise keyword representations for factoid questions. These patterns underline the importance of tailoring identifier strategies to dataset characteristics.

Reinforcement learning also proves effective. Removing GRPO drops Recall@1 by 0.9% and MRR@10 by 0.5% on NQ, and NDCG@10 by 3.8% on NF-Corpus. On TriviaQA, it reduces Recall@1 by 2.8% and MRR@10 by 1.8%, demonstrating GRPO's utility in enhancing semantic relevance. Backbone choice significantly impacts performance. Replacing the LLM with BART-large and removing GRPO ("w/o LLM+GRPO") yields the largest degradation: Recall@1 drops by 6.5% on TriviaQA, 3.3% on NQ, and NDCG@10 by 6.1% on NF-Corpus. Both a strong backbone and GRPO are critical for effective identifier generation and retrieval, especially in complex QA tasks.

4.3 The Effect of Negative Sample Selection Strategy (RQ3)

We conduct studies on negative sample selection strategies in the GRPO reinforcement learning phase. Specifically, we select the top-M positive samples and M negative samples from Top-k candidates for training. We evaluate the following strategies for selecting negative samples: (1) Easy: Selecting the top 10 negative samples (i.e., those ranked higher) from the Top-k candidates. (2) Mixed: Dividing the negative samples into three difficulty levels:hard, medium, and easy, selecting 3, 4, and 3 samples respectively from each level. (3) Hard: Selecting the bottom 10 negative samples (i.e., those ranked lower) from the Top-k candidates.

The results are summarized in Table 7, Table 8, and Table 9. As shown, For NFCorpus, the *Mixed* strategy achieves the best performance across all metrics, particularly in terms of NDCG@10. Specifically, the *Mixed* strategy outperforms both the *Hard* and *Easy* strategies by +3.5% and +2.8% respectively in NDCG@10. This suggests that balancing the difficulty levels of negative samples contributes to better ranking performance. The *Mixed* negative sampling strategy consistently yields higher Recall@1 scores on both the NQ and TriviaQA datasets, surpassing the *Hard* and *Easy* strategies by +0.6% and +1.0% on NQ, and by +0.6% and +1.4% on TriviaQA, respectively. Overall, the *Mixed* strategy consistently outperforms the *Hard* and *Easy* strategies across different datasets (NFCorpus, TrivaQA, and NQ) and evaluation metrics, demonstrating its robustness and effectiveness in improving retrieval performance by balancing the difficulty levels of negative samples. Further analysis indicates that the *Mixed* strategy effectively captures a diverse range of negative samples, which helps the model learn more robust representations. In contrast, the *Hard* strategy may focus too much on difficult cases, leading to suboptimal generalization, while the *Easy* strategy may not provide sufficient challenge for the model to improve its ranking capabilities.

Table 7: Performance Comparison of Negative Sample Selection Strategies in NFCorpus.

Strategy	NDCG@10	R@5	R@20	R@100
Mixed	**0.373**	**0.144**	**0.220**	**0.270**
Hard	0.338	0.140	0.212	0.265
Easy	0.345	0.131	0.201	0.264

Table 8: Performance Comparison of Negative Sample Selection Strategies in TrivaQA.

Strategy	R@1	R@5	R@10	MRR@10
Mixed	**0.600**	**0.752**	**0.792**	**0.665**
Hard	0.594	0.750	0.791	0.662
Easy	0.586	0.746	0.788	0.655

Table 9: Performance Comparison of Negative Sample Selection Strategies in NQ.

Strategy	R@1	R@5	R@10	MRR@10
Mixed	**0.495**	**0.687**	**0.736**	**0.576**
Hard	0.489	0.686	0.733	0.571
Easy	0.485	0.683	0.736	0.569

4.4 The Effect of Hierarchical Ranking Fusion (RQ4)

We perform a grid search over the weights used in the hierarchical list result merging strategy (as described in Section 2.3.1), ranging from 0.1 to 1.0 with an interval of 0.1. Each configuration is evaluated on the validation set, and the final weight is chosen based on the setting that achieves the highest average score across all evaluation metrics. We analyze the performance of the proposed Hierarchical Ranking Fusion (HRF) framework across three datasets: NQ, TriviaQA, and NFCorpus. As shown in Figure 2, on NQ, HRF achieves Recall@1 0.495 and MRR@10 0.576, improving by up to 1.0% and 0.7% over the rank method. On TriviaQA, it reaches Recall@1 0.600 and MRR@10 0.665, again outperforming others. The improvements are especially notable at Recall@1, showing HRF's strength in ranking the most relevant document first. On NFCorpus, HRF obtains NDCG@10 0.373 and Recall@5 0.144, outperforming frequency (0.352/0.141) and rank-based (0.322/0.138) baselines. These results demonstrate that the hierarchical merging strategy effectively integrates frequency and rank cues, leading to more accurate and generalizable retrieval.

4.5 Case Study (RQ5)

Query Group Retrieval Analysis. To demonstrate the semantic consistency of our retrieval algorithm, we analyze retrieval results for queries in groups A and B,see Figure 3 and Figure 4. The x-axis represents document IDs and the y-axis shows their scores.

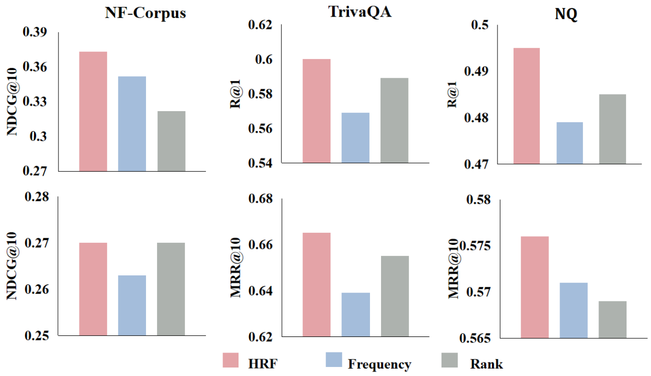

Figure 2: Performance Comparison of HRF, frequency and Rank.

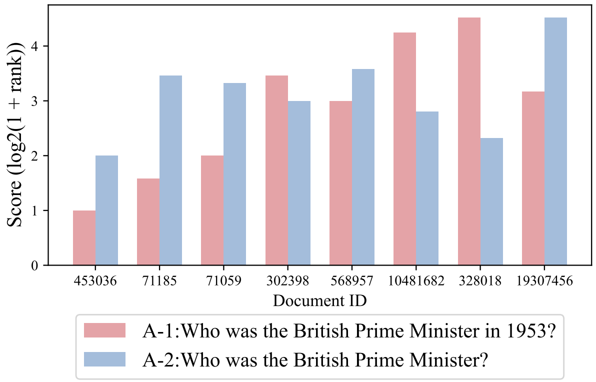

Figure 3: Scores of retrieved documents for Query Group A.

Similar queries retrieve closely distributed and partially overlapping documents, while dissimilar queries from different groups yield distinct, non-overlapping sets. For example, group A often retrieves documents 71185, 568957, and 71059, while group B retrieves 42305, 129601, and 581476. To further illustrate this, we visualize BGE-based document embeddings using t-SNE in Figure 5, where each color denotes documents linked to a specific query. The formation of two clear clusters confirms that our method effectively captures group-level semantics and improves retrieval accuracy.

Qualitative Analysis of GRPO Effectiveness. To illustrate the effect of GRPO, Table 10 compares retrieval results before and after training. Retrieval is performed using the query concatenated with generated identifiers (summary, keyword, pseudo-query). Before GRPO, identifiers only partially capture the query intent: the pseudo-query mentions Darwin but omits Wallace, and the keyword "evolution" is overly generic, leading to a result focused mainly on Darwin. After GRPO, the identifiers become more precise: the keyword explicitly includes "Darwin," the summary states his role in evolution, and the pseudo-query is broadened to "Who first published evolution theory?", covering both Darwin and Wallace. This yields a top-1 document that highlights their joint contributions. Overall, GRPO enhances the informativeness and semantic alignment of identifiers, resulting in more accurate and complete retrieval.

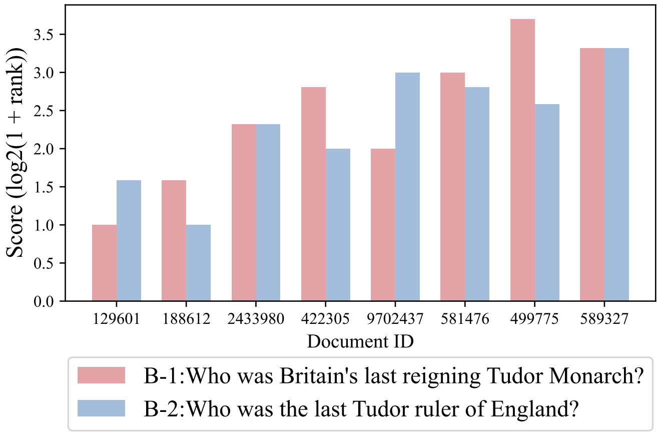

Figure 4: Scores of retrieved documents for Query Group B.

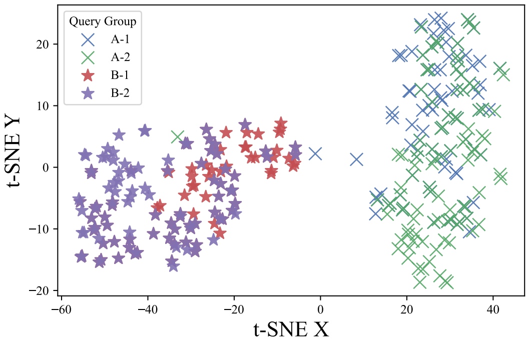

Figure 5: t-SNE visualization of BGE document embeddings.

5 Related Work

5.1 Document Retrieval

Document retrieval aims to find relevant documents from a large corpus given a user query. Traditional methods use a multi-stage pipeline of indexing and ranking, and are mainly divided into sparse and dense retrieval. Sparse retrieval, such as BM25 [23], relies on inverted indexes and exact term matching. BM25 is efficient but depends on lexical overlap, making it insensitive to semantic relationships and unable to model contextual information. Dense retrieval models, like DPR [9], represent queries and documents as dense vectors and use approximate nearest neighbor search. Thanks to pre-trained language models, dense retrieval achieves better performance, but still faces challenges such as limited end-to-end optimization, embedding bottlenecks, and insufficient fine-grained query-document interactions.

5.2 Generative Retrieval

Unlike traditional retrieval models, generative retrieval models directly utilize language models to generate identifiers for relevant

Table 10: Comparison of generated identifiers and top-ranked document identifiers before and after GRPO.

Query: who proposed evolution in 1859 as the basis of biological development?	
Answer: Charles Darwin, Alfred Russel Wallace	
Generated Identifiers before GRPO	**Generated Identifiers after GRPO**
Keyword: evolution **Summary:** Darwin’s ’On the Origin of Species’ introduced evolution. **Pseudo-Query:** What did Charles Darwin propose in On the Origin of Species?	**Keyword:** Darwin **Summary:** Darwin proposed evolution in ’On the Origin of Species’. **Pseudo-Query:** Who first published evolution theory?
Top 1 document’s identifiers (Before GRPO)	**Top 1 document’s identifiers (After GRPO)**
Keyword: Darwinism, biological evolution, natural selection, Charles Darwin, species **Summary:** Darwinism explains species evolution via natural selection. **Pseudo-Query:** – What is Darwinism and how does it explain evolution? – How does natural selection contribute to biological evolution? – Who was Charles Darwin and what did he discover?	**Keyword:** biological organization, evolution, natural selection, Charles Darwin, Alfred Russel Wallace **Summary:** Evolution by natural selection explained by Darwin and Wallace. **Pseudo-Query:** – What is evolution by natural selection? – How did Darwin and Wallace contribute to evolutionary theory? – What is Darwin’s Origin of Species about?

documents based solely on the input query. GENRE [6] is one of the early representatives in this field, employing an auto-regressive language model with prefix-constrained beam search to generate target entity titles from a candidate set. Recent research has primarily focused on improving document representation and training objectives. Based on their representational forms, document identifiers can be broadly categorized into numerical and lexical types. Numerical identifiers model document similarity through clustering methods [4, 27, 28] or product quantization [34], offering compact semantic encodings. However, due to the semantic gap between numerical representations and natural language, they struggle to fully leverage the capabilities of pre-trained language models. In contrast, lexical identifiers—being natural language sequences—are easier to generate and interpret, thus receiving increasing attention. For instance, Ultron [34] uses document URLs and titles, SEAL [1] selects n-grams as identifier candidates, MINDER [14] introduces multi-view combinations, and TSGen [31] constructs representations based on sets of salient terms.

On the training side, researchers have incorporated ranking supervision to improve generation quality. LTRGR [15] utilizes a marginal ranking loss based on generation probabilities to enhance document-level ranking performance. GLEN [11] models query-document relevance via a two-stage indexing process, while DGR [16] introduces teacher model-based ranking information through knowledge distillation. D2Gen [5] and DOGR [17] adopt contrastive learning strategies to enhance the semantic distinctiveness of document identifiers. GenRRL [32] and Re3val [25] apply reinforcement learning to generative retrieval, using KL-regularized rewards with contextual re-ranking to improve identifier generation and retrieval performance. However, most of these approaches rely on static supervision signals, limiting their ability to directly optimize retrieval performance. Despite notable progress in identifier design and training objectives, the mismatch between generation objectives and downstream retrieval performance is still remained due to insufficient semantic expressiveness of identifiers. To address these limitations, we propose a reinforcement learning-driven generative retrieval framework that incorporates semantic-aligned multi-layer identifiers to comprehensively capture document semantics. By using the similarity margin between the generated identifier and positive/negative documents as rewards, our framework generate more discriminative DocIDs for advanced retrieval performance.

6 Conclusions

This paper proposes GRAM-RL, a generative retrieval framework based on multi-layer identifiers and reinforcement learning, designed to address limited identifier quality and insufficient query-document interaction in generative retrieval. GRAM-RL generates three types of document identifiers: summaries, keywords, and pseudo-queries to capture document semantics at different levels of granularity. A prompt-driven multi-task training strategy enables joint optimization of identifier generation. Furthermore, the framework introduces a multi-view vector fusion mechanism and a GRPO-based reinforcement learning module, which together enhance retrieval performance. Experimental results demonstrate that GRAM-RL significantly outperforms existing generative retrieval approaches across multiple benchmark datasets, particularly in terms of recall and ranking quality. In future work, we plan to explore dynamically adapting identifier granularity based on query characteristics to further improve semantic alignment. Integrating real-time user feedback into the reinforcement learning loop for online optimization of retrieval strategies is another direction. Additionally, extending GRAM-RL to multi-modal retrieval scenarios and optimizing its efficiency for large-scale corpus will be key research focuses to enhance its practical applicability and scalability.

Acknowledgments

This work was supported in part by the Ministry of Education Humanities and Social Science Project (No.22YJC740110), the Liaoning Provincial Natural Science Foundation Joint Fund Program(2023-MSBA-003), and the Fundamental Research Funds for the Central Universities (DUT24MS003).

GenAI Usage Disclosure

This work involved the use of generative AI tools in limited and supervised ways. Specifically, DeepSeek was used to assist with generating multi-level identifiers, including summaries, keywords, and pseudo-queries, during the data preparation stage. In addition, OpenAI's ChatGPT was employed solely for minor language polishing, such as grammar correction and wording improvements. All uses of GenAI tools were conducted under the full supervision of the authors, who are solely responsible for the final content.

References

[1] Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers. *Advances in Neural Information Processing Systems* 35 (2022), 31668–31683.

[2] Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A full-text learning to rank dataset for medical information retrieval. In *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38*. Springer, 716–722.

[3] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Regina Barzilay and Min-Yen Kan (Eds.). Association for Computational Linguistics, Vancouver, Canada, 1870–1879. doi:10.18653/v1/P17-1171

[4] Xiaoyang Chen, Yanjiang Liu, Ben He, Le Sun, and Yingfei Sun. 2023. Understanding differential search index for text retrieval. *arXiv preprint arXiv:2305.02073* (2023).

[5] Jiehan Cheng, Zhicheng Dou, Yutao Zhu, and Xiaoxi Li. 2025. Descriptive and Discriminative Document Identifiers for Generative Retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 11518–11526.

[6] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. *arXiv preprint arXiv:2010.00904* (2020).

[7] DeepSeek-AI. 2024. DeepSeek-V3 Technical Report. arXiv:2412.19437 [cs.CL] https://arxiv.org/abs/2412.19437

[8] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551* (2017).

[9] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. arXiv:2004.04906 [cs.CL] https://arxiv.org/abs/2004.04906

[10] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur 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 Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics* 7 (2019), 452–466. doi:10.1162/tacl_a_00276

[11] Sunkyung Lee, Minjin Choi, and Jongwuk Lee. 2023. GLEN: Generative Retrieval via Lexical Index Learning. 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, 7693–7704. doi:10.18653/v1/2023.emnlp-main.477

[12] 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. doi:10.18653/v1/2020.acl-main.703

[13] 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 (2020), 9459–9474.

[14] 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. doi:10.18653/v1/2023.acl-long.366

[15] Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2024. Learning to Rank in Generative Retrieval. In *AAAI*. 8716–8723. https://doi.org/10.1609/aaai.v38i8.28717

[16] Yongqi Li, Zhen Zhang, Wenjie Wang, Liqiang Nie, Wenjie Li, and Tat-Seng Chua. 2024. Distillation Enhanced Generative Retrieval. In *Findings of the Association for Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 11119–11129. doi:10.18653/v1/2024.findings-acl.662

[17] Penghao Lu, Xin Dong, Yuansheng Zhou, Lei Cheng, Chuan Yuan, and Linjian Mo. 2025. DOGR: Leveraging Document-Oriented Contrastive Learning in Generative Retrieval. *arXiv preprint arXiv:2502.07219* (2025).

[18] Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith B Hall, Daniel Cer, and Yinfei Yang. 2021. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877* (2021).

[19] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. 2021. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899* (2021).

[20] Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage Re-ranking with BERT. arXiv:1901.04085 [cs.IR] https://arxiv.org/abs/1901.04085

[21] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375* (2019).

[22] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 Technical Report. arXiv:2412.15115 [cs.CL] https://arxiv.org/abs/2412.15115

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

[24] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv:2402.03300 [cs.CL] https://arxiv.org/abs/2402.03300

[25] EuiYul Song, Sangryul Kim, Haeju Lee, Joonkee Kim, and James Thorne. 2024. Re3val: Reinforced and Reranked Generative Retrieval. In *Findings of the Association for Computational Linguistics: EACL 2024*, Yvette Graham and Matthew Purver (Eds.). Association for Computational Linguistics, St. Julian's, Malta, 393–409. https://aclanthology.org/2024.findings-eacl.27/

[26] Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, Maarten Rijke, and Zhaochun Ren. 2023. Learning to tokenize for generative retrieval. *Advances in Neural Information Processing Systems* 36 (2023), 46345–46361.

[27] Yi Tay, Vinh 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. *Advances in Neural Information Processing Systems* 35 (2022), 21831–21843.

[28] Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, et al. 2022. A neural corpus indexer for document retrieval. *Advances in Neural Information Processing Systems* 35 (2022), 25600–25614.

[29] 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*. 2656–2665.

[30] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-Pack: Packed Resources For General Chinese Embeddings. arXiv:2309.07597 [cs.CL] https://arxiv.org/abs/2309.07597

[31] Peitian Zhang, Zheng Liu, Yujia Zhou, Zhicheng Dou, Fangchao Liu, and Zhao Cao. 2024. Generative Retrieval via Term Set Generation. 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, 458–468. doi:10.1145/3626772.3657797

[32] 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. doi:10.18653/v1/2023.emnlp-main.768

[33] 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. doi:10.1145/3603167

[34] 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. arXiv:2208.09257 [cs.IR] https://arxiv.org/abs/2208.09257

[35] Shengyao Zhuang, Houxing Ren, Linjun Shou, Jian Pei, Ming Gong, Guido Zuccon, and Daxin Jiang. 2023. Bridging the Gap Between Indexing and Retrieval for Differentiable Search Index with Query Generation. arXiv:2206.10128 [cs.IR] https://arxiv.org/abs/2206.10128