ReSuMe: Retriever-Summarizer Mutual Enhancement via Reinforcement Learning

Owais Makroo[1,2,†], Nikhil Pattisapu[2], Karan Gupta[2], Ankit Gandhi[2], Vijay Huddar[2], Atul Saroop[2]
{omakroo, npattisa, karaniis, ganankit, vhhuddar, asaroop}@amazon.com
[1]IIT Kharagpur, Kharagpur, WB, India
[2]Amazon Inc., Bangalore, KA, India

Abstract

We present ReSuMe, a general framework for mutual enhancement of dense retrieval systems and document summarizers through reinforcement learning. The framework jointly optimizes a language model for generating retrieval-oriented summaries and adapts the retrieval model to these summaries through alternating fine-tuning phases. We employ Group Relative Policy Optimization (GRPO) to fine-tune the language model based on retrieval relevance rather than linguistic quality alone, while the retrieval model is iteratively updated using contrastive learning on the generated summaries. This co-optimization process addresses the fundamental distribution shift problem that arises when retrieval models trained on full documents must operate on synthetic summaries during inference. By progressively reducing this distribution gap, our framework yields two key benefits: improved retrieval performance and a high-quality document summarizer optimized for retrieval tasks. We demonstrate our framework using Contriever on the MS-MARCO dataset, achieving consistent improvements of **13.2% in MRR@10** and **6.7% in Recall@100** over the baseline. The framework is model-agnostic and can be applied to enhance any dense retrieval system while simultaneously producing an effective document summarization model.

Keywords

Information Retrieval, Document Summarization, Language Models, Dense Retrieval, Policy Optimization

ACM Reference Format:
Owais Makroo[1,2,†], Nikhil Pattisapu[2], Karan Gupta[2], Ankit Gandhi[2], Vijay Huddar[2], Atul Saroop[2] . 2026. ReSuMe: Retriever-Summarizer Mutual Enhancement via Reinforcement Learning. In *Proceedings of the ACM Web Conference 2026 (WWW '26), April 13–17, 2026, Dubai, United Arab Emirates.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3774904.3792959

1 Introduction

Dense retrieval systems are fundamental components of modern information retrieval and retrieval-augmented generation (RAG) systems [12, 14]. By encoding queries and documents into dense vector representations, these models enable semantic similarity matching that goes beyond traditional keyword-based approaches. However, documents in most corpora are inherently sparse and often contain substantial amounts of non-informative or redundant content, making efficient retrieval even more challenging.

Document summarization has emerged as a promising approach to address this efficiency challenge. By replacing full documents with concise summaries during retrieval, systems can reduce memory requirements and accelerate inference while preserving essential semantic information. However, this approach introduces a fundamental **distribution shift** problem: retrieval models trained on full documents must operate on LLM-generated summaries at inference time, creating a mismatch between training distribution $P(d)$ and inference distribution $P_\theta(s|d)$ that often degrades retrieval performance [1, 10].

Existing approaches to retrieval-augmented summarization typically treat summarization and retrieval as independent components. Static summarization methods using models like PEGASUS [5] or BART [9] generate summaries without considering their impact on downstream retrieval performance [18].

An alternative line of work has explored asymmetric retrieval architectures, where queries and documents are encoded differently to optimize for retrieval efficiency. However, training asymmetric models presents significant challenges. Methods like KALE [7] demonstrate that achieving effective asymmetric encoding requires careful design to maintain semantic alignment between query and document representations. The asymmetric training process often suffers from representational drift, where the query and document encoders learn divergent embedding spaces, leading to suboptimal retrieval performance [6]. Furthermore, maintaining semantic consistency across different encoding pathways requires sophisticated training procedures and architectural constraints that can limit model flexibility [17].

Recent work has increasingly explored reinforcement learning (RL) approaches for retrieval-oriented summarization and retrieval-augmented generation (RAG). LightRetriever [3] employs Group Relative Policy Optimization (GRPO) [16] to train summarization models that enhance retrieval effectiveness, yet it treats the summarizer and retriever as fixed components, leaving unresolved the distribution shift that emerges between training and inference when the two are not jointly adapted. Building on this direction, DeepRetrieval [11] directly optimizes retrieval performance via reinforcement learning by aligning large language models with real search engines and retrievers. Instead of relying on supervised query–document pairs, it trains LLMs to generate retrieval-optimized queries using reward signals derived from retrieval metrics, achieving notable gains even with relatively small models. While both of these advances highlight the potential of reinforcement learning for retrieval alignment, they primarily focus on optimizing one

[†]Work done during an internship at Amazon Inc. Correspondence: makroo.owais@kgpian.iitkgp.ac.in.

WWW '26, Dubai, United Arab Emirates

ACM ISBN 979-8-4007-2307-0/2026/04
https://doi.org/10.1145/3774904.3792959

component (either query generation or summarization) in isolation, leaving open the challenge of jointly adapting both the retriever and the generator to the evolving distribution of document representations.

Complementary to these directions, Kim et al. [8] propose a dual-augmentation approach for retrieval-augmented generation, where large language models enhance both the query and the retrieved passages to improve open-domain question answering. Their method decomposes complex user queries into sub-questions to facilitate more accurate retrieval and supplements retrieved content with self-generated passages from the LLM's internal knowledge. While this approach improves retrieval coverage and robustness, it assumes a static retrieval model and does not explicitly address the distributional mismatch that arises when the underlying document representations or summarization models evolve over time.

To bridge this gap, MMOA-RAG [15] formulates RAG as a multi-agent reinforcement learning problem, jointly optimizing query rewriting, retrieval, filtering, and answer generation under a shared reward based on answer quality. RAG-RL [4] further employs curriculum based RL to progressively align retrieval and generation, enhancing factual consistency and precision. Collectively, these approaches mark a shift toward fully adaptive RAG pipelines where retriever and generator co-evolve under shared reinforcement signals, effectively mitigating distribution shift and improving robustness in retrieval-oriented summarization.

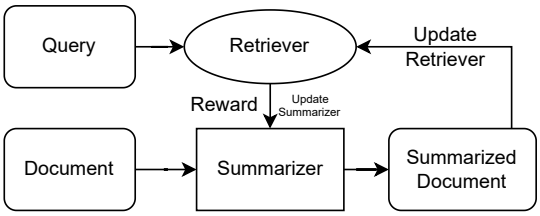

Figure 1: Overview of the ReSuMe framework.

We propose **ReSuMe** (Retriever-Summarizer Mutual Enhancement), a novel framework (Figure 1) that addresses distribution shift through joint optimization of both retrieval and summarization models. Unlike previous approaches that optimize these components in isolation, ReSuMe alternates between optimizing the summarizer and retriever under a shared retrieval-based signal. This mutual enhancement process progressively aligns both models, reducing the distribution gap while yielding two valuable outputs: an improved retrieval system and a specialized document summarizer optimized for retrieval tasks.

ReSuMe is designed as a general framework that can enhance any dense retrieval architecture, making it broadly applicable across different retrieval systems and domains. The framework's model-agnostic design allows practitioners to apply it to existing retrieval pipelines without architectural modifications, while the iterative optimization ensures that both the summarizer and retriever co-evolve toward optimal performance.

Our main contributions are (1) **ReSuMe Framework**: A general iterative training framework for mutual enhancement of dense retrieval systems and document summarizers that can be applied to any retrieval architecture. (2) **Distribution Shift Mitigation**: A principled approach to addressing the fundamental mismatch between training on full documents and inference on summaries through joint optimization. (3) **Empirical Validation**: Comprehensive evaluation showing 13.2% improvement in MRR@10 and 6.7% in Recall@100 on MS-MARCO when applied to Contriever, validating the framework's effectiveness.

Through systematic mutual enhancement, ReSuMe bridges the gap between retrieval and summarization, providing a unified approach to jointly optimizing both tasks while addressing the practical challenges of deploying efficient retrieval systems at scale.

2 Methodology

Given a collection of documents $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$ and queries $\mathcal{Q} = \{q_1, q_2, \ldots, q_m\}$, ReSuMe's objective is to enhance any dense retrieval model by jointly learning a summarization function $f_\theta : d \rightarrow s$, where s denotes a summary, and adapting the retrieval function $g_\phi : (q, s) \rightarrow \mathbb{R}$ to operate effectively on these summaries. The framework alternates between (i) optimizing the LLM summarizer via Group Relative Policy Optimization (GRPO) to generate retrieval-oriented summaries, and (ii) adapting the retrieval model to operate effectively on the newly generated summaries. Starting with any pretrained retrieval model g_{ϕ_0} and LLM f_{θ_0}, the procedure proceeds iteratively: at each iteration t, the LLM is fine-tuned using GRPO where $g_{\phi_{t-1}}$ serves as the reward model. The updated LLM f_{θ_t} produces a new set of corpus summaries $S_t = \{f_{\theta_t}(d_i) : d_i \in \mathcal{D}\}$, which are then used to fine-tune the retrieval model using its native training procedure (e.g., contrastive learning for dense retrievers). The updated retriever parameters ϕ_t are obtained as $\phi_t = \text{RetrieverUpdate}(g_{\phi_{t-1}}, S_t, \mathcal{Q})$. This iterative process continues until convergence as formalized in Algorithm 1.

Algorithm 1 ReSuMe: Retriever-Summarizer Mutual Enhancement

```
Initialize LLM f_{θ_0} and retrieval model g_{φ_0}
for iteration t = 1 to T do
    Fine-tune LLM using GRPO with g_{φ_{t-1}} as reward model
    θ_t = GRPO(f_{θ_{t-1}}, g_{φ_{t-1}})
    Summarize full corpus: S_t = {f_{θ_t}(d_i) : d_i ∈ D}
    Fine-tune retriever using Contriever pipeline on S_t
    φ_t = RetrieverUpdate(g_{φ_{t-1}}, S_t, Q)
    if reward improvement < ε then break
end for
```

Group Relative Policy Optimization (GRPO). To optimize summaries for retrieval, we employ GRPO, which fine-tunes the LLM based on relative preferences across a group of candidate summaries rather than absolute scores. For a group $G = \{s_1, \ldots, s_k\}$ corresponding to a document d, the GRPO objective is defined as:

$$\mathcal{L}_{\text{GRPO}} = -\log \frac{\exp(r(s_i)/\beta)}{\sum_{j=1}^{k} \exp(r(s_j)/\beta)}, \tag{1}$$

where $r(s)$ denotes the scalar reward and β is a temperature parameter that smooths preference differences.

The total reward comprises two complementary components - retrieval effectiveness and structural quality:

$$r(s,d) = \text{StructureReward}(s) + r_\phi(q^+, q^-, s, d), \quad (2)$$

where q^+ and q^- represent sets of positive and negative queries for d. Negative queries are sampled uniformly from $Q \setminus q^+$. The StructureReward(s) term enforces fluency, brevity, and proper format, while $r_\phi(q^+, q^-, s, d)$ quantifies the improvement in retrieval relevance induced by summarization.

Retrieval-based Reward. Δ_ϕ measures the difference in retrieval quality between a summary and its corresponding original document:

$$\Delta_\phi(q^+, q^-, s, d) = \text{RetScore}_\phi(q^+, q^-, s) - \text{RetScore}_\phi(q^+, q^-, d), \quad (3)$$

with

$$\text{RetScore}_\phi(q^+, q^-, x) = \frac{\sum_{q\in q^+} \text{S}_\phi(q,x)}{\sum_{q\in q^+} \text{S}_\phi(q,x) + \sum_{q\in q^-} \text{S}_\phi(q,x)}, \quad (4)$$

where $\text{S}_\phi(q,x)$ denotes the cosine similarity between query and text embeddings.

Discrete Reward Mapping. Following prior work [19], we discretize the continuous retrieval signal Δ_ϕ into a piecewise linear reward scale, r_ϕ, to improve GRPO stability. Negative Δ_ϕ values are assigned penalizing rewards of -1.0, -0.8, -0.6, and -0.3 for the ranges $[-1, -0.5)$, $[-0.5, -0.2)$, $[-0.2, -0.1)$, and $[-0.1, 0)$, respectively, while positive Δ_ϕ values yield proportional gains of 0.3, 0.6, 0.8, and 1.0 for $[0, 0.1)$, $[0.1, 0.2)$, $[0.2, 0.5)$, and $[0.5, 1]$, respectively.

Structural Reward. The StructureReward(s) encourages summaries that are concise, well-formed, and adhere to the reasoning-to-summary format. It is computed using the retrieval model's tokenizer to evaluate the portion of text generated after the `</think>` token. The `</think>` token serves as an explicit separator between the model's internal reasoning and the final summary, ensuring that only the intended summary portion is evaluated and preventing leakage of chain-of-thought into the retrieved representation. Summaries whose tokenized length lies within 64–128 tokens receive a reward of 1, slightly overlength summaries receive 0.9, and underlength ones receive 0.3. If the `</think>` token is absent, both StructureReward(s) and Δ_ϕ are assigned -1, penalizing invalid or unstructured outputs.

Retriever Adaptation. After each LLM update, the retriever is fine-tuned on the updated summaries S_t using a contrastive loss:

$$\mathcal{L}_{\text{retrieval}} = -\log \frac{\exp(\text{S}(q, s^+))}{\exp(\text{S}(q, s^+)) + \sum_{s^-\in\mathcal{N}} \exp(\text{S}(q, s^-))}, \quad (5)$$

where s^+ is a relevant summary, $\mathcal{N}$ is the set of negatives, and $\text{S}(\cdot,\cdot)$ denotes cosine similarity. This stage aligns g_ϕ to the summary distribution, thereby reducing the domain shift.

3 Experimental Setup

To validate our framework's effectiveness, we demonstrate its application using Contriever[1] [2] as the base retrieval model on the MS-MARCO passage ranking dataset [13]. This dataset contains 8.8M passages and 500K+ queries for training, with separate development and evaluation sets, making it suitable for evaluating retrieval performance improvements.

Implementation. We initialize the summarization component with Qwen3-4B-Thinking-2507 [20] and apply our iterative GRPO framework to fine-tune both the LLM and Contriever model. The framework is implemented using PyTorch with GRPO using batch size 128 over 200 steps, learning rate 3e-7, and temperature $\beta = 0.5$. The retrieval model adaptation uses AdamW optimizer with learning rate 1e-5. The iterative process runs for a maximum of ten iterations with early stopping based on validation convergence. We use greedy decoding during inference to ensure deterministic outputs. All document summaries are pre-computed and cached for efficient retrieval.

Baselines. We compare our iterative framework against: (1) **Baseline**: Original Contriever without summarization, (2) **Fixed Summary**: Contriever with static Qwen-generated summaries, (3) **Non-iterative**: Single-step LLM fine-tuning without retriever adaptation, and (4) **Iterative (Ours)**: Full iterative framework with GRPO. We evaluate using standard IR metrics: Mean Reciprocal Rank at 10 (MRR@10), Recall at 100/1000 (R@100/R@1000), and Normalized Discounted Cumulative Gain at 10 (NDCG@10). While contrastive training with hard negative examples can further enhance retrieval performance, we exclude such techniques from both the baseline and our proposed method to maintain experimental simplicity and prevent confounding effects.

Hardware. LLM alignment (GRPO) is performed on 40 NVIDIA A100 GPUs (40GB) using Ray, while standard LLM fine-tuning uses Fully Sharded Data Parallel (FSDP). Retrieval model adaptation is conducted with Distributed Data Parallel (DDP) across 8 A100 GPUs. Although GRPO alignment incurs a substantial training cost, this overhead does not affect deployment: all documents can be summarized offline, so inference requires no additional alignment computation.

4 Results and Discussion

Table 1 demonstrates the effectiveness of our iterative framework applied to Contriever on MS-MARCO. The iterative approach achieves substantial improvements across all metrics, with 13.2% relative improvement in MRR@10, 13.6% in NDCG@10, 6.7% in R@100, and 2.3% in R@1000 compared to the baseline. Notably, fixed summarization without adaptation performs worse than the original model (0.234 vs 0.266 MRR@10), highlighting the critical importance of addressing distribution shift through joint optimization.

Table 1: Performance comparison on MS-MARCO dataset.

Method	MRR@10	NDCG@10	R@100	R@1000
Baseline	0.266	0.324	0.819	0.954
Fixed Summary	0.234	0.290	0.789	0.943
Non-iterative	0.235	0.291	0.792	0.942
Iterative (Ours)	**0.301**	**0.368**	**0.874**	**0.976**

[1] https://huggingface.co/facebook/contriever

Out-of-Distribution Generalization. Our model also improves out-of-distribution performance when evaluated on NFCorpus, as shown in Table 2. While the gains of the Iterative method over the Baseline and Fixed Summary settings are smaller than those observed in-distribution, this is expected due to the domain shift between MS-MARCO and NFCorpus.

Table 2: Out-of-distribution performance on NFCorpus.

Method	MRR@10	NDCG@10	R@100
Baseline	0.501	0.305	0.290
Fixed Summary	0.470	0.280	0.283
Iterative (Ours)	**0.535**	**0.329**	**0.303**

Ablation Studies. Table 3 summarizes the ablation results. The full model attains the best NDCG@10. Removing the retrieval adaptation module causes the largest drop, confirming its importance for aligning the retriever with summarized inputs. Excluding negative query generation or the discrete reward also reduces performance, indicating that each component, and the iterative optimization they support, contributes meaningfully to overall effectiveness.

Table 3: Ablation study results (NDCG@10)

Configuration	NDCG@10
Full model	**0.368**
w/o retrieval adaptation	0.291
w/o negative queries	0.362
w/o discrete reward	0.340

Convergence Analysis. The iterative framework demonstrates stable convergence within 3-5 iterations. Performance improvements are most significant in the first two iterations, with diminishing returns afterward. The validation MRR@10 stabilizes around third iteration, indicating effective convergence of the joint optimization process.

Summary Quality. We present a representative example of a document and its corresponding summary generated by the best model on **MS-MARCO**. The full evolution of summaries across iterations is available on the accompanying GitHub page [2]. As shown in Table 4, the generated summary is concise and focused, effectively capturing the core information relevant to retrieval while omitting peripheral details.

5 Conclusion

We introduced an iterative fine-tuning framework that jointly optimizes a GRPO-trained summarizer and a Contriever-based retriever, enabling effective adaptation to LLM-generated summaries and yielding strong performance gains on MS-MARCO, including a 13.2% relative improvement in MRR@10 and a 6.7% gain in Recall@100. These results demonstrate that iterative co-training is highly effective, even before considering the downstream implications of distribution shift, and that GRPO provides stable, reward-driven optimization suited for retrieval-oriented summarization.

[2]https://makrooowais.github.io/resume-paper/

Table 4: Example of original passage and generated summary (MS-MARCO).

	Content
Original Passage	Primary biliary cirrhosis, or PBC, is a chronic, or long-term, disease of the liver that slowly destroys the medium-sized bile ducts within the liver. Bile is a digestive liquid that is made in the liver. It travels through the bile ducts to the small intestine, where it helps digest fats and fatty vitamins.In patients with PBC, the bile ducts are destroyed by inflammation. This causes bile to remain in the liver, where gradual injury damages liver cells and causes cirrhosis, or scarring of the liver.As cirrhosis progresses and the amount of scar tissue in the liver increases, the liver loses its ability to function.t travels through the bile ducts to the small intestine, where it helps digest fats and fatty vitamins. In patients with PBC, the bile ducts are destroyed by inflammation. This causes bile to remain in the liver, where gradual injury damages liver cells and causes cirrhosis, or scarring of the liver.
Generated Summary	Primary biliary cirrhosis (PBC) is a chronic liver disease that slowly destroys medium-sized bile ducts within the liver. In PBC, bile ducts are destroyed by inflammation, causing bile to accumulate in the liver. This bile accumulation leads to gradual liver cell injury and cirrhosis, or scarring of the liver. Bile is a digestive liquid produced by the liver that normally travels through bile ducts to the small intestine to aid fat and fatty vitamin digestion. PBC causes progressive liver scarring as cirrhosis advances. Primary biliary cirrhosis is a chronic liver disease.x

The overall performance improvements highlight the importance of aligning the retriever to the evolving summary distribution, pointing to distribution shift as a key challenge in retrieval systems using synthetic or compressed inputs.

Looking ahead, we aim to extend this framework to multi-modal retrieval, larger-scale datasets, and alternative policy-optimization methods. We also plan to study the impact of summarizer and retriever model sizes, incorporate continual-learning strategies to prevent retriever drift, and investigate the theoretical convergence properties of iterative co-training.

References

[1] A. Wang et al. 2023. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. *arXiv:2310.11511* (2023).
[2] G. Izacard et al. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv:2112.09118* (2021).
[3] G. Ma et al. 2025. LightRetriever: A LLM-based Text Retrieval Architecture with Extremely Faster Query Inference. *arXiv:2505.12260* (2025).
[4] J. Huang et al. 2025. RAG-RL: Advancing Retrieval-Augmented Generation via Reinforcement Learning and Curriculum Learning. *arXiv:2503.12759* (2025).
[5] J. Zhang et al. 2020. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. *arXiv:1912.08777* (2020).
[6] K. Santhanam et al. 2022. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. *arXiv:2112.01488* (2022).
[7] L. Xiong et al. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.
[8] M. Kim et al. 2024. QPaug: Question and Passage Augmentation for Open-Domain Question Answering of LLMs. *arXiv:2406.14277* (2024).
[9] M. Lewis et al. 2019. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *arXiv:1910.13461* (2019).
[10] N. Thakur et al. 2021. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv:2104.08663* (2021).
[11] P. Jiang et al. 2025. DeepRetrieval: Hacking Real Search Engines and Retrievers with Large Language Models via Reinforcement Learning. *arXiv:2503.00223* (2025).
[12] P. Lewis et al. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems* (2020).
[13] T. Nguyen et al. 2016. MS MARCO: A human generated machine reading comprehension dataset. *arXiv:1611.09268* (2016).
[14] V. Karpukhin et al. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP*.
[15] Y. Chen et al. 2024. MMOA-RAG: Improving Retrieval-Augmented Generation through Multi-Agent Reinforcement Learning. *arXiv:2501.15228* (2024).
[16] Z. Shao et al. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv:2402.03300* (2024).
[17] O. Khattab and M. Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of SIGIR*.
[18] Y. Liu and M. Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of EMNLP-IJCNLP*.
[19] Y. Mroueh. 2025. Reinforcement Learning with Verifiable Rewards: GRPO's Effective Loss, Dynamics, and Success Amplification. *arXiv:2503.06639* (2025).
[20] Qwen Team. 2025. Qwen3 Technical Report. *arXiv:2505.09388* (2025).