

Generalized Pseudo-Relevance Feedback

Yiteng Tu

DCST, Tsinghua University
Beijing, China
tyt24@mails.tsinghua.edu.cn

Weihang Su

DCST, Tsinghua University
Beijing, China

Yujia Zhou

DCST, Tsinghua University
Beijing, China

Yiqun Liu

DCST, Tsinghua University
Beijing, China

Fen Lin

Tencent
Beijing, China

Qin Liu

Tencent
Beijing, China

Qingyao Ai*

DCST, Tsinghua University
Beijing, China
aiqy@tsinghua.edu.cn

Abstract

Query rewriting is a fundamental technique in information retrieval (IR). It typically employs the retrieval result as relevance feedback to refine the query and thereby addresses the vocabulary mismatch between user queries and relevant documents. Traditional pseudo-relevance feedback (PRF) and its vector-based extension (VPRF) improve retrieval performance by leveraging top-retrieved documents as relevance feedback. However, they are constructed based on two major hypotheses: the relevance assumption (top documents are relevant) and the model assumption (rewriting methods need to be designed specifically for particular model architectures). While recent large language models (LLMs)-based generative relevance feedback (GRF) enables model-free query reformulation, it either suffers from severe LLM hallucination or, again, relies on the relevance assumption to guarantee the effectiveness of rewriting quality. To overcome these limitations, we introduce an assumption-relaxed framework: *Generalized Pseudo Relevance Feedback* (GPRF), which performs model-free, natural language rewriting based on retrieved documents, not only eliminating the model assumption but also reducing dependence on the relevance assumption. Specifically, we design a utility-oriented training pipeline with reinforcement learning to ensure robustness against noisy feedback. Extensive experiments across multiple benchmarks and retrievers demonstrate that GPRF consistently outperforms strong baselines, establishing it as an effective and generalizable framework for query rewriting.

CCS Concepts

- Information systems → Query reformulation; • Computing methodologies → Natural language generation.

Keywords

Query Rewriting, Pseudo-Relevance Feedback, Retrieval, Large Language Models, Assumption-Relaxed

1 Introduction

Search engines have become indispensable tools for accessing information, powering applications ranging from web search and e-commerce to open-domain question answering and knowledge-grounded dialogue [3, 16, 28, 44]. A central goal of these systems is to bridge the gap between user queries and vast document collections [21, 27, 45]. However, a long-standing challenge in search engines lies in the vocabulary mismatch problem: users often express information needs with general and ambiguous terms, while

relevant documents may employ more formal, specialized, or emergent terminology [21, 45]. To address this gap, query rewriting has emerged as a crucial technique, enriching initial queries with semantically related or contextually grounded expressions to enhance the likelihood of retrieving documents that align with the user’s intent [1, 25]. Over decades of research, it has proven to be an effective method for improving retrieval effectiveness in both classical and neural search paradigms [10].

Query rewriting approaches typically first perform an initial retrieval using the original query and then refine it with top-ranked retrieved results, regarding them as the relevance feedback. One of the most well-known paradigms is pseudo-relevance feedback (PRF) [1, 17, 25, 36]. It estimates term distributions from the top retrieved documents, assuming that they are relevant, and interpolates them with the original query term distribution, thereby improving retrieval performance and robustness of sparse retrievers like BM25 [24]. Vector-based pseudo-relevance feedback (VPRF) [19, 20], on the other hand, is a variant of PRF tailored for dense retrieval [13, 16, 42, 43] scenarios, which directly aggregates dense embeddings of the top-retrieved documents to refine the query representation. It effectively leverages the semantic richness of neural representations and has been shown to boost retrieval effectiveness across a range of tasks [20].

Despite their effectiveness, both PRF and VPRF are fundamentally constrained by two strong assumptions that severely limit their robustness and generalizability. The first one is **relevance assumption**, which assumes that all the top-ranked documents retrieved in the initial stage should be relevant and thus be beneficial for query rewriting. While this assumption may hold in carefully curated test collections, it is far from true in real-world scenarios, where retrieval systems are inherently imperfect and top results often include noisy and irrelevant information [35, 37]. Once these noisy and off-topic documents are incorporated into the rewriting process, they can introduce misleading content and even drift the reformulation away from the user’s true intent. The second assumption is **model assumption**: these methods are tightly coupled to a specific retriever’s internal representations. By operating at the level of term weights or dense embeddings, the rewritten query is inherently tied to a particular model’s feature space, making it challenging to transfer across different or evolving retrieval systems. This rigidity and coupling not only narrow their applicability but also constrain the exploration of richer, more flexible reformulation strategies [45]. The reliance on these two assumptions makes PRF

and VPRF highly vulnerable to noisy feedback and difficult to adapt across models, motivating the search for alternative approaches.

Recently, the rise of large language models (LLMs) [2, 4, 8] has led to a new type of methods named generative relevance feedback (GRF) [10, 23, 27, 39]. Given a short or ambiguous query, an LLM can synthesize pseudo-documents or detailed answer-style passages that articulate the user’s information need with richer context [10, 21, 39]. By operating at the natural language level instead of adjusting weights or embeddings, GRF mitigates the model assumption: the reformulated query is no longer bound to a specific embedding space, making the approach more interpretable and transferable across different retrieval models and domains. However, GRF methods still rest on the relevance assumption, assuming that the generated expansions faithfully reflect the user’s intent and provide useful retrieval cues. In practice, this assumption is also questionable, as LLMs are prone to hallucination, producing fluent but factually incorrect or semantically irrelevant content [30, 33].

To address the limitations above, we introduce *Generalized Pseudo-Relevance Feedback* (GPRF), a generative, evidence-guided query rewriting framework that relaxes both assumptions. From the model perspective, GPRF overcomes the limitation of PRF by leveraging LLMs to conduct natural language-based query reformulation and avoids the hallucination problem of GRF by grounding the process with top-retrieved documents (as shown in Figure 1). From the relevance perspective, GPRF relaxes the assumption on top-retrieved documents’ quality by introducing a comprehensive, utility-oriented optimization pipeline. This pipeline is specifically designed to make the generative model robust to noisy feedback through three stages: retrieval-augmented rejection sampling filters unfaithful generations and selects high-quality training samples, supervised fine-tuning equips the model with the initial ability to generate high-quality rewrites, and reinforcement learning directly aligns the model with retrieval utility. By explicitly forcing the model to learn from reliable feedback in the earlier stage and then shaping its generation behavior with task-aligned rewards, our training pipeline corrects the model’s tendency to propagate misleading and detrimental information. This process empowers the model to discern and leverage useful signals even from imperfect feedback, thus substantially mitigating the negative impact of irrelevant documents. Extensive experiments across multiple retrievers and benchmarks demonstrate that GPRF consistently outperforms strong baselines, including both classical PRF methods and recent GRF approaches. These results highlight the effectiveness of our method, positioning GPRF as a promising direction for advancing query reformulation in retrieval systems.

In summary, this paper makes three key contributions: (1) We conduct a systematic analysis of existing query rewriting methods, including PRF and GRF, highlighting two major challenges they face: the reliance on **relevance assumption** and the **model assumption**. (2) We propose Generalized Pseudo-Relevance Feedback (GPRF) and a corresponding utility-oriented training pipeline, which effectively integrates the advantages of PRF and GRF while alleviating their weaknesses. (3) Extensive experiments show that our framework consistently outperforms strong baselines, demonstrating its effectiveness and generalizability.

2 Preliminary

2.1 Sparse Retrieval and Dense Retrieval

We consider the standard ad-hoc retrieval setting, where a system takes as input a query q and ranks documents from a large collection $\mathcal{D} = \{d_1, d_2, \dots, d_N\}$. The retrieval process relies on a scoring function $s(q, d)$ that estimates the relevance between the query and a document. Traditional sparse retrieval methods, such as BM25, represent queries and documents as high-dimensional sparse vectors over the vocabulary space \mathcal{V} . Each dimension corresponds to a term, weighted by functions such as term frequency (TF) and term frequency-inverse document frequency (TF-IDF). The relevance score is computed by lexical matching:

$$s_{\text{sparse}}(q, d) = \sum_{t \in q \cap d} w_q(t) \cdot w_d(t), \quad (1)$$

where $w_q(t)$ and $w_d(t)$ denote the weights of term t in the query and document representations, respectively. While effective and interpretable, sparse retrieval is inherently limited to surface-level term overlap and often fails to capture semantic similarity.

In contrast, dense retrieval encodes queries and documents into low-dimensional dense vectors using a neural encoder $\text{Enc}(\cdot)$. Each query and document is mapped into the same semantic space (denoted as \mathbf{q} and \mathbf{d}), and their relevance is estimated via similarity measures such as inner product or cosine similarity:

$$\mathbf{q} = \text{Enc}(q), \quad \mathbf{d} = \text{Enc}(d), \quad (2)$$

$$s_{\text{dense}}(q, d) = \langle \mathbf{q}, \mathbf{d} \rangle, \quad (3)$$

where $\langle \cdot, \cdot \rangle$ denotes dot product or cosine similarity. This formulation enables retrieval beyond exact term overlap, capturing paraphrases and deeper semantic relations. However, the query and document representations are tied to the embedding space of a specific model, making adaptation and transfer across different retrievers more challenging, leading to the model assumption.

2.2 Pseudo-Relevance Feedback and Generative Relevance Feedback

Pseudo-relevance feedback expands the initial query q by leveraging the top- k documents retrieved in the first stage. Let $\mathcal{D}_q^{(k)} = \{d_1, d_2, \dots, d_k\}$ denotes the feedback set obtained from the initial retrieval. Classical PRF methods estimate a relevance model $p(t | q)$ over terms $t \in \mathcal{V}$ using statistics from $\mathcal{D}_q^{(k)}$. A common formulation, as in RM3, interpolates the original query term distribution with the feedback model [1]:

$$p(t | q') = (1 - \alpha) \cdot p(t | q) + \alpha \cdot \sum_{d \in \mathcal{D}_q^{(k)}} p(t | d) \cdot p(d | q), \quad (4)$$

where q' is the reformulated query, $\alpha \in [0, 1]$ controls the interpolation weight, and $p(t | \mathcal{D}_q^{(k)})$ is estimated from the feedback documents. This reformulation is then used for subsequent retrieval under the sparse retrieval framework.

In dense retrieval settings, PRF is performed directly in the embedding space. VPRF refines the query representation \mathbf{q} by aggregating feedback document vectors [19, 21]:

$$\mathbf{q}' = \alpha \cdot \mathbf{q} + \beta \cdot \sum_{i=1}^k \mathbf{d}_i, \quad (5)$$

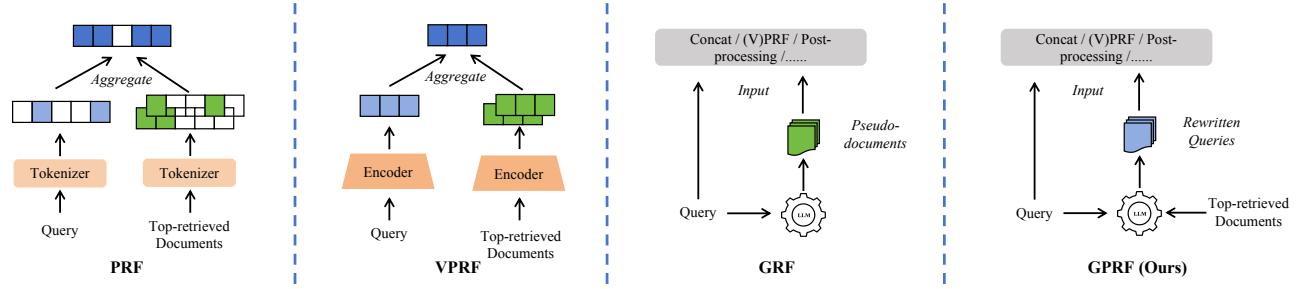


Figure 1: The comparison between PRF, VPRF, GRF, and our proposed GPRF. The pseudo-documents or rewritten queries produced by GRF and GPRF can be processed in various ways, such as directly concatenating them with the original query, integrating them into PRF or VPRF systems, or performing retrieval separately for each, then post-processing the results.

where $\mathbf{d}_i = \text{Enc}(d_i)$, $d_i \in \mathcal{D}_q^{(k)}$ is the embedding of the feedback document, while α, β control the contribution of the original query and feedback documents.

On the other hand, large language models (LLMs) have recently been employed for query rewriting in natural language. Given the initial query, GRF uses an LLM parameterized by θ to construct expansions like pseudo-document, pseudo-answer, etc. [10, 27]: $d' \sim \text{LLM}_\theta(\mathcal{I}, q)$, where \mathcal{I} denotes the instruction. It articulates the information needs in a more detailed form. The final reformulated query is obtained by concatenation:

$$q' = [q; d'_1; d'_2; \dots], \quad (6)$$

where $[\cdot ; \cdot ; \cdot]$ denotes text concatenation, and d'_i represents different sample result. The reformulated query is then used for retrieval with either sparse or dense methods.

3 Methodology

3.1 Generalized Pseudo-Relevance Feedback

Building on both PRF and GRF, we propose *Generalized Pseudo-Relevance Feedback* (GPRF), which integrates retrieval evidence with generative rewriting. A comparison of GPRF against PRF, VPRF, and GRF is shown in Figure 1. Specifically, given the initial query q and its top- k retrieved documents $\mathcal{D}_q^{(k)}$, an LLM directly generates the rewritten query:

$$q' \sim \text{LLM}_\theta(\mathcal{I}, q, \mathcal{D}_q^{(k)}). \quad (7)$$

To enhance robustness and diversity, multiple reformulations can be sampled to capture different possible user intents. These diverse reformulations can be conveniently incorporated into downstream retrieval, such as appending them directly to the original query (i.e., concatenation in Figure 1), encoding them into embeddings and then aggregating them (just like PRF and VPRF), or directly conducting retrieval on these samples and post-processing the retrieval results. This design combines the semantic grounding of PRF with the expressive generative capacity of LLMs, providing a model-agnostic mechanism that bridges sparse and dense retrieval. On the other hand, GPRF can also be seamlessly combined with methods such as few-shot learning and Chain-of-Thought (CoT), which we leave for future work.

Nevertheless, this retrieval-augmented generation-based query rewriting paradigm is not without challenges. Although grounding in feedback documents reduces hallucinations compared to GRF, their effectiveness is limited because the generative model remains sensitive to noisy or off-topic feedback documents, which may mislead reformulations and degrade retrieval performance [35, 37]. These challenges motivate the development of a dedicated training method to control generation quality better and alleviate the influence of noisy feedback.

3.2 Utility-oriented Training Pipeline

3.2.1 Overview.

As discussed above, a key challenge in generative query reformulation lies in the vulnerability of LLMs to noisy feedback documents: irrelevant or misleading evidence can easily distort the rewriting process, leading to suboptimal or even harmful expansions. To address this issue, we design a *utility-oriented training pipeline* that explicitly incorporates ultimate retrieval performance into the model training process. By optimizing query rewriting not only for fluency or faithfulness, but also for retrieval effectiveness, the pipeline strengthens the model’s robustness against noisy inputs and enhances its ability to produce reliable, utility-driven reformulations aligned with the downstream retrieval task.

As shown in Figure 2, our pipeline consists of three stages. First, we perform sampling-based evaluation to identify the rewritten queries that maximize retrieval utility. Second, the best-performing samples are used to construct high-quality supervision signals for fine-tuning. Finally, reinforcement learning (RL) with direct utility-based rewards further aligns the model toward the ultimate goal of query rewriting. Together, these stages form an iterative framework that grounds query reformulation in retrieval performance while improving both accuracy and resilience.

3.2.2 Retrieval-augmented Rejection Sampling.

The first stage of our pipeline is retrieval-augmented rejection sampling, which aims to filter out low-utility query reformulations and retain only those that improve retrieval effectiveness. Concretely, given an initial query q and its top- k feedback documents $\mathcal{D}_q^{(k)}$, the rewriting model (LLM_θ) generates a set of candidate reformulations:

$$\{q'_1, q'_2, \dots, q'_M\} \sim \text{LLM}_\theta(\mathcal{I}, q, \mathcal{D}_q^{(k)}), \quad (8)$$

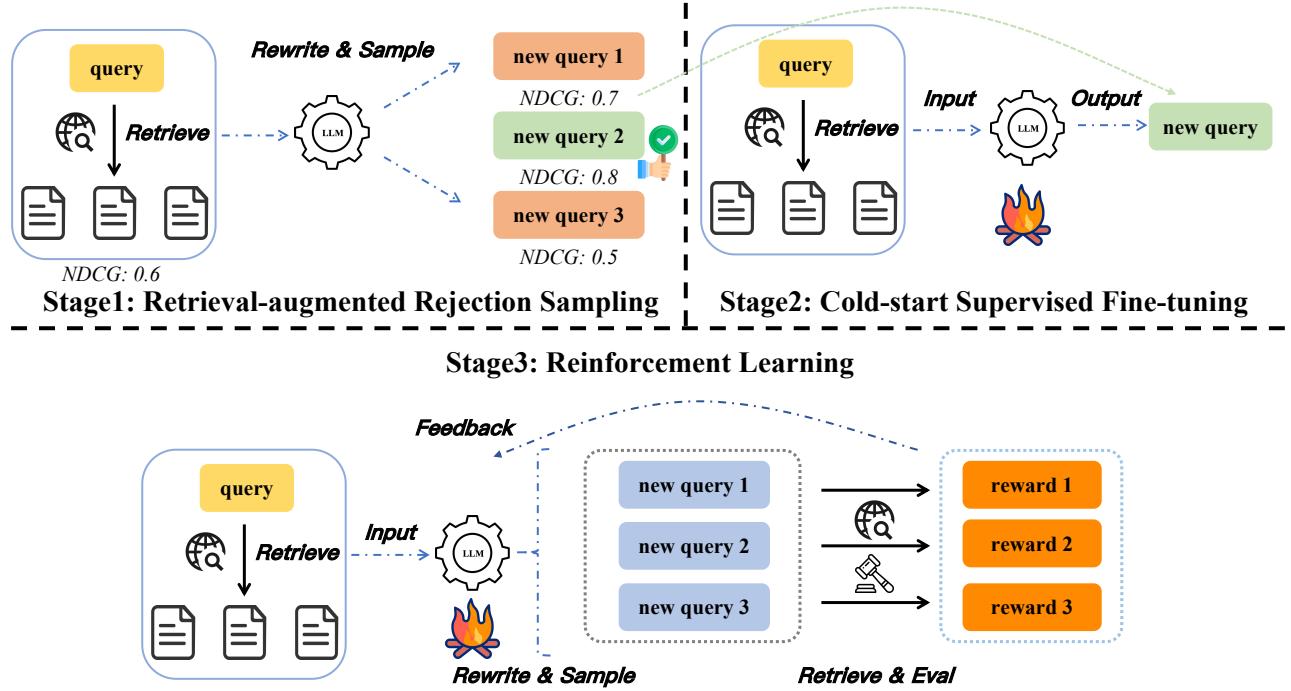


Figure 2: Overview of the Utility-oriented Training Pipeline. The high-utility reformulations obtained via rejection sampling in Stage 1 are directly utilized as training labels for supervised fine-tuning (SFT) in Stage 2. In Stage 3, we directly use the performance on downstream retrieval tasks as the reward signal of reinforcement learning (RL).

where M denotes the number of sampled rewrites (e.g., $M = 10$) and \mathcal{I} is the instruction (detailed in Table 1). Each reformulated query q'_j is then submitted back to the retrieval system, producing a ranked list of documents $\mathcal{D}_{q'_j}^{(k)}$. To evaluate its effectiveness, we measure a utility function $U(\cdot)$, defined as the improvement in retrieval quality (for example, NDCG@10 [15]) compared to the original query q :

$$U(q'_j) = \text{NDCG}@10(\mathcal{D}_{q'_j}^{(k)}) - \text{NDCG}@10(\mathcal{D}_q^{(k)}). \quad (9)$$

Finally, we select the reformulation with the highest utility score as the accepted rewrite:

$$q^* = \arg \max_{q'_j} U(q'_j). \quad (10)$$

This procedure ensures that only rewrites yielding the greatest retrieval improvement are retained, while others are discarded. In this way, retrieval-augmented rejection sampling provides high-quality pseudo-supervision signals for the subsequent supervised fine-tuning stage, effectively grounding query rewriting in retrieval performance and mitigating the influence of noisy feedback.

3.2.3 Cold-start Supervised Fine-tuning (SFT).

After obtaining high-utility reformulations from retrieval-augmented rejection sampling, we use them as pseudo-supervision signals to initialize the rewriting model. This stage provides the model with explicit guidance on how to generate reformulations that improve retrieval performance.

Table 1: The prompt template for the query rewriting module of GPRF. For the various tasks and datasets presented in this paper, we employ a uniform prompt.

Please rewrite the user's query based on several relevant passages (which may contain noise or errors). The rewritten query should preserve the original meaning while incorporating as much information as possible, so that search engines can more effectively retrieve relevant passages.
 Relevant Passages:
 Passage 1: {passage 1}
 Passage 2: {passage 2}

 User Query: {question}
 Rewritten Query:

Formally, let $D_{\text{SFT}} = \{(x, y)\}$ denotes the SFT dataset, where input $x = (\mathcal{I}; q; \mathcal{D}_q^{(k)})$ can be regarded as a combination of the instruction \mathcal{I} , the initial query q , and the original feedback $\mathcal{D}_q^{(k)}$. The output $y = q^*$ is the selected formulation with the highest utility for query q . It minimizes the negative log-likelihood (NLL)

of generating the target reformulation:

$$\mathcal{L}_{\text{SFT}}(\theta) = - \sum_{i=1}^{|y|} \log p_\theta(y_i | x, y_{<i}). \quad (11)$$

This training step encourages the model to imitate utility-driven reformulations, thereby reducing its tendency to be misled by noisy feedback. By grounding the model in supervised signals, SFT establishes a strong initialization that enhances both stability and convergence in the subsequent RL stage.

3.2.4 Reinforcement Learning (RL).

While SFT provides the model with high-utility reformulation examples, it cannot fully address the variability of real retrieval scenarios, where feedback may be noisy and ambiguous. To further align the model with retrieval-oriented objectives, we adopt reinforcement learning (RL) with the Generalized Reweighted Policy Optimization (GRPO) [11, 26] algorithm. It samples a group of outputs $G = \{y_1, \dots, y_{|G|}\}$ for each input x during training, and each y_i corresponds to a reward r_i . To jointly account for retrieval performance at top ranks and the overall recall, we adopt a multi-view reward function defined as:

$$r_i = \text{NDCG}@10(\mathcal{D}_{y_i}^{(k)}) + \lambda \cdot \text{Recall}@100(\mathcal{D}_{y_i}^{(k)}), \quad (12)$$

where λ is a trade-off hyper-parameter. The rewards are then normalized within the group to produce the advantage function:

$$\hat{A}_i = \frac{r_i - \text{mean}(\{r_1, \dots, r_{|G|}\})}{\text{std}(\{r_1, \dots, r_{|G|}\})} \quad (13)$$

Thus, the overall loss function is formulated as:

$$\begin{aligned} \mathcal{L}_{\text{GRPO}}(\theta) = & - \frac{1}{|G|} \sum_{i=1}^{|G|} \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \min \left(r_{i,t}(\theta) \hat{A}_i, \right. \\ & \left. \text{clip}(r_{i,t}(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_i \right) - \beta D_{\text{KL}} [\pi_\theta || \pi_{\text{ref}}], \end{aligned} \quad (14)$$

where $r_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t}|x, y_{i,<t})}{\pi_{\text{old}}(y_{i,t}|x, y_{i,<t})}$ is the importance ratio, and ϵ as well as β are hyper-parameters. Through this RL stage, the model is directly optimized for retrieval effectiveness rather than imitation alone. Combined with rejection sampling and SFT, GRPO equips the rewriting model with greater robustness to noisy feedback and stronger utility-driven reformulation capabilities.

4 Experimental Setup

4.1 Datasets & Evaluation Metrics

To train the GPRF model, we use the MS-MARCO Passage Retrieval dataset [5], which provides large-scale query–document pairs for supervised retrieval. In the cold-start SFT stage, we sample 200k instances from the dataset and apply rejection sampling (following §3.2.2) based on a commonly used sparse retriever, BM25 [24], and a dense retriever, e5-base-v2 [38], selecting the top 30k instances with the greatest improvement from both retrievers to construct the training set D_{SFT} . In the RL stage, we similarly sample 200k instances directly from the dataset, while randomly assigning BM25 or e5-base-v2 as the retriever to construct the reward function.

We evaluate the performance of the retrieval model before and after query rewriting on both in-domain and out-of-domain retrieval benchmarks to assess the effectiveness and generalizability comprehensively. For in-domain evaluation, we report results on the MS-MARCO Passage Retrieval dev set (MS dev) [5] as well as the TREC Deep Learning track 2019 (DL19) and 2020 (DL20) [6]. To evaluate out-of-domain robustness, following Gao et al. [10], we test on six publicly available low-resource datasets from the BEIR benchmark [34], namely ArguAna, DBpedia, FiQA-2018, SCIDOCs, SciFact, and TREC-COVID. Following previous works [10, 20], we adopt two standard retrieval metrics: NDCG@10 and Recall@100 (R@100). NDCG@10 emphasizes effectiveness for highly relevant documents at top ranks, while R@100 reflects the system’s ability to cover a broader set of relevant results.

4.2 Baselines

We mainly compare GPRF with three categories of methods. The first is the direct retrieval baseline, where no rewriting is applied and the system relies solely on the original query. The second category includes PRF-based approaches. For sparse retrieval, we use RM3 [1], a classical lexical feedback method that expands the query distribution with terms from top-ranked documents. For dense retrieval, we adopt VPRF [19], which refines the query embedding by aggregating representations of feedback documents. The third category consists of zero-shot GRF methods that employ LLMs for query rewriting. Specifically, we consider three methods: HyDE [10], which generates hypothetical answer passages as pseudo-documents to enrich queries; CoT [14], which leverages the Chain-of-Thought to provide a rationale for the pseudo-answer; and LameR [27], which follows a retrieve–answer–retrieve pipeline where pseudo-answers are generated to improve retrieval performance. These approaches provide a comprehensive comparison, allowing us to evaluate GPRF not only against traditional lexical and dense feedback methods but also against recent LLM-based generative rewriting approaches under both in-domain and out-of-domain retrieval settings.

4.3 Implementation Details

4.3.1 Model Selection. We experiment with various retrieval models as well as backbone LLMs. For retrievers, we consider BM25 [24] as the classical sparse approach, along with two dense retrievers: e5-base-v2 (E5) [38], which serves as our primary in-domain dense retriever, and bge-base-en-v1.5 (BGE) [41], which is not employed during the training stage and therefore functions as an out-of-domain model to test generalizability. For the query rewriting model, to balance the performance and efficiency, we select two LLMs of moderate size: Llama-3.2-3B-Instruct [8] (Llama) and Qwen2.5-3B-Instruct [4] (Qwen).

4.3.2 Training Settings. We use four NVIDIA A100-SXM4-40GB GPUs for training GPRF models. In the SFT stage, the model is trained for 2 epochs with a learning rate of 1e-6. We set both the *per-device training batch size* and the *gradient accumulation steps* to 8. In the RL stage, we train the model for 1 epoch with the same learning rate. Here, we increase the *per-device batch size* and *gradient accumulation steps* to 16, set the group size $|G| = 8$,

Table 2: Evaluation results of different rewriting methods on in-domain datasets. The best and second-best methods of each retriever are marked in bold and underlined, respectively. "L" and "Q" denote using Llama-3-3.2B-Instruct and Qwen2.5-3B-Instruct as the backbone model, while " \dagger " and " \ddagger " indicate significantly worse than the best and second-best method at the $p < 0.05$ level using the two-tailed pairwise t-test, respectively.

Method	MS dev		DL 19		DL 20	
	NDCG@10	R@100	NDCG@10	R@100	NDCG@10	R@100
BM25	0.2284 \ddagger	0.6578 \ddagger	0.5058 \ddagger	0.4531 \ddagger	0.4796 \ddagger	0.4834 \ddagger
+RM3	0.2023 \ddagger	0.6538 \ddagger	0.5216 \ddagger	0.4821 \dagger	0.4896 \ddagger	0.5316 \dagger
+HyDE _L	0.2023 \ddagger	0.6425 \ddagger	0.6001 \dagger	0.4795 \dagger	0.5733 \ddagger	0.5542
+HyDE _Q	0.2224 \ddagger	0.6829 \ddagger	0.6030 \dagger	0.4890 \dagger	0.5845 \ddagger	0.5539
+CoT _L	0.2233 \ddagger	0.6786 \ddagger	0.6215 \dagger	0.4923	0.5973 \ddagger	0.5738
+CoT _Q	0.2339 \ddagger	0.6914 \ddagger	0.5480 \ddagger	0.4625 \dagger	0.5468 \ddagger	0.5535
+Lamer _L	0.2367 \ddagger	0.6773 \ddagger	0.6361 \dagger	0.4849 \dagger	0.5975 \dagger	0.5718
+Lamer _Q	0.2593 \ddagger	0.6830 \ddagger	0.6589	0.5091	0.6219	0.5594
+GPRF _L	0.3208	0.7486	0.6917	0.5401	0.6707	0.5849
+GPRF _Q	0.3016 \dagger	0.7179 \dagger	0.6461 \dagger	0.4952	0.6332	0.5343
E5	0.4179 \ddagger	0.8878\dagger	0.7048	0.5375	0.7039 \ddagger	0.6019 \ddagger
+VPRF	0.3262 \ddagger	0.8555 \ddagger	0.6765 \dagger	0.5671	0.7027 \ddagger	0.5943 \ddagger
+HyDE _L	0.3291 \ddagger	0.8124 \ddagger	0.7096	0.5273 \dagger	0.6895 \ddagger	0.5871 \ddagger
+HyDE _Q	0.3579 \ddagger	0.8467 \ddagger	0.6781	0.5344	0.7006 \ddagger	0.6005 \ddagger
+CoT _L	0.3036 \ddagger	0.7755 \ddagger	0.5992 \ddagger	0.4532 \ddagger	0.6001 \ddagger	0.5218 \ddagger
+CoT _Q	0.2983 \ddagger	0.7767 \ddagger	0.5941 \ddagger	0.4507 \ddagger	0.5988 \ddagger	0.5088 \ddagger
+Lamer _L	0.3459 \ddagger	0.8046 \ddagger	0.6723 \dagger	0.4968 \ddagger	0.7096 \ddagger	0.5955 \ddagger
+Lamer _Q	0.3594 \ddagger	0.8011 \ddagger	0.6873 \dagger	0.4881 \ddagger	0.7297 \dagger	0.6094 \dagger
+GPRF _L	0.4283	0.8852 \dagger	0.7228	0.5405	0.7585	0.6205
+GPRF _Q	0.4231 \dagger	0.8904	0.7382	<u>0.5541</u>	0.7524	0.6257
BGE	0.4134 \ddagger	0.8856\dagger	0.7245	0.5174 \dagger	0.7052 \ddagger	0.5797
+VPRF	0.3200 \ddagger	0.8500 \ddagger	0.7096 \dagger	0.5436	0.6921 \ddagger	0.5765 \dagger
+HyDE _L	0.3348 \ddagger	0.8197 \ddagger	0.7263	0.5390 \dagger	0.7231	0.5769
+HyDE _Q	0.3639 \ddagger	0.8527 \ddagger	0.7002 \dagger	0.5388	0.7197 \dagger	0.5720 \dagger
+CoT _L	0.3531 \ddagger	0.8288 \ddagger	0.6898 \ddagger	0.4935 \ddagger	0.6822 \ddagger	0.5572 \dagger
+CoT _Q	0.3407 \ddagger	0.8224 \ddagger	0.6738 \ddagger	0.5036 \ddagger	0.6363 \ddagger	0.5101 \ddagger
+Lamer _L	0.3676 \ddagger	0.8368 \ddagger	0.7495	<u>0.5587</u>	0.7210 \dagger	0.5929
+Lamer _Q	0.3754 \ddagger	0.8407 \ddagger	0.7581	0.5341 \dagger	0.7263	0.5441 \ddagger
+GPRF _L	0.4262	0.8846 \dagger	0.7555	0.5560	0.7384	0.5778
+GPRF _Q	0.4234	0.8897	0.7612	0.5711	0.7613	0.6025

use a sampling temperature of 1.0, and apply the KL-divergence regularization term β with 1e-3.

4.3.3 Evaluation Settings. For evaluation, we set the temperature to 0 to ensure deterministic decoding, feed $k = 10$ retrieved documents into the LLMs, and sample $M = 10$ reformulated queries for each input. To combine these reformulations with retrieval, we follow different strategies for sparse and dense retrievers. For BM25, we concatenate all reformulated queries with the original query to form the final input. For dense retrievers, we apply the VPRF strategy instead, aggregating the embeddings of the reformulated queries to construct the refined query representation.

5 Results and Analysis

In this section, we mainly aim to explore the following three research questions thoroughly:

- **RQ1:** Can GPRF perform effectively on in-domain data and generalize to out-of-domain data at the same time?
- **RQ2:** Can GPRF relax the relevance assumption and tolerate noisy data in feedback documents?
- **RQ3:** Can GPRF mitigate the model assumption and perform effectively for retrievers not presented in the training process, with top documents retrieved by or not by themselves?

5.1 Main Results (RQ1)

Table 2 presents the evaluation results on three in-domain datasets. We observe that our GPRF consistently outperforms all baselines across different retrievers and evaluation metrics. For the sparse retriever BM25, both RM3 and recent GRF methods (i.e., HyDE, CoT, LameR) bring moderate gains, but their improvements are unstable and often limited by the relevance assumption. In contrast, GPRF achieves substantial improvements, with up to 40.5% NDCG@10 improvement on MS dev and 39.8% on DL 20 compared to the vanilla BM25 retriever, demonstrating its strong ability to leverage retrieval evidence and generate effective reformulations. For dense retrievers, E5 and BGE, similar trends can be observed. While VPRF provides some benefits by aggregating document embeddings, its performance lags behind GRF methods that operate at the natural language level and break the constraints of model assumptions. Among GRF baselines, LameR and HyDE are competitive, but GPRF consistently delivers the best or second-best results across nearly all settings. Notably, on DL20 with BGE, GPRF boosts NDCG@10 from 0.7052 to 0.7613, significantly surpassing the best performance of other baselines. When comparing different backbone LLMs, both Llama and Qwen yield robust results, with Qwen generally better suited to BGE and Llama more aligned with BM25. For E5, Qwen demonstrates stronger recall performance, while Llama delivers slightly better precision-oriented gains (NDCG@10). This suggests that GPRF is not only effective but also adaptable across different model backbones. Overall, it demonstrates that GPRF is particularly advantageous in in-domain settings, consistently mitigating the weaknesses of both traditional PRF and GRF, delivering substantial and statistically significant improvements across varying sparse and dense retrievers.

Table 3 further shows the results on six out-of-domain datasets when using Llama as the backbone rewriting model. It can be observed that GPRF still achieves the best overall performance across all retrievers and datasets, consistently outperforming both classical PRF and recent GRF approaches. For the sparse retriever BM25, traditional PRF (i.e., RM3) often fails to generalize and even underperforms the vanilla BM25 algorithm on several datasets, confirming its vulnerability to noisy feedback in distribution-shifted settings. GRF methods such as HyDE, CoT, and LameR yield moderate improvements on specific datasets (e.g., CoT on SciFact, LameR on DBpedia), but their performance is highly inconsistent and fails to dominate across tasks. In contrast, GPRF not only achieves the best overall performance but also consistently delivers the highest NDCG@10 across all datasets. On the other hand, for the two dense retrievers, similar trends hold. GPRF not only surpasses VPRF but

Table 3: Evaluation results of different rewriting methods on out-of-domain datasets with Llama as the backbone model. The best and second-best methods of each retriever are marked in bold and underlined, respectively. "†" and "‡" indicate significantly worse than the best and second-best method at the $p < 0.05$ level using the two-tailed pairwise t-test, respectively.

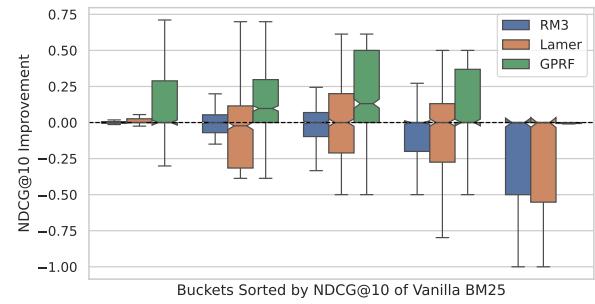
Method	ArguAna		DBpedia		FiQA-2018		SCIDOCs		SciFact		TREC-COVID		Avg.	
	NDCG@10	R@100	NDCG@10	R@100	NDCG@10	R@100	NDCG@10	R100	NDCG@10	R@100	NDCG@10	R@100	NDCG@10	R@100
BM25	0.2999 [†]	0.9324 ^{†‡}	0.3180 ^{†‡}	0.4682 ^{†‡}	<u>0.2361[†]</u>	<u>0.5395[†]</u>	0.1490 [†]	0.3477 ^{†‡}	0.6789 [†]	0.9253	0.5947 ^{†‡}	0.1091 ^{†‡}	0.3794	0.5537
+RM3	0.2865 ^{†‡}	0.9552	0.3080 ^{†‡}	0.4594 ^{†‡}	0.1916 ^{†‡}	0.4967 ^{†‡}	<u>0.1491[†]</u>	<u>0.3621</u>	0.6457 ^{†‡}	0.9147 [†]	0.5927 ^{†‡}	0.1168 [†]	0.3623	0.5508
+HyDE	0.2794 ^{†‡}	0.9260 ^{†‡}	0.3303 ^{†‡}	0.4719 ^{†‡}	0.1803 ^{†‡}	0.4931 ^{†‡}	0.1140 ^{†‡}	0.3283 ^{†‡}	0.6557 ^{†‡}	0.9377	0.6458 [†]	0.1264 [†]	0.3676	0.5472
+CoT	<u>0.3013[†]</u>	0.9403 [†]	0.3510 ^{†‡}	0.5054	0.2081 ^{†‡}	0.5187 [†]	0.1372 ^{†‡}	0.3532 [†]	0.6971	0.9437	0.6815 [†]	0.1264 [†]	0.3960	0.5646
+Lamer	0.2547 ^{†‡}	0.9189 ^{†‡}	0.3909	<u>0.5164</u>	0.1970 ^{†‡}	0.4703 ^{†‡}	0.1267 ^{†‡}	0.3120 ^{†‡}	<u>0.7047</u>	0.9397	0.6824 [†]	0.1298 [†]	0.3927	0.5479
+GPRF	<u>0.3139</u>	0.9452	0.4009	<u>0.5200</u>	<u>0.2912</u>	<u>0.5935</u>	<u>0.1579</u>	<u>0.3665</u>	<u>0.7127</u>	0.9367	<u>0.7738</u>	0.1441	<u>0.4417</u>	0.5843
E5	0.3258 ^{†‡}	0.9467 ^{†‡}	<u>0.4226[†]</u>	0.5420	<u>0.3991[†]</u>	0.7324 [†]	<u>0.1862[†]</u>	<u>0.4211</u>	0.7200 ^{†‡}	0.9627	0.6961 ^{†‡}	0.1287 ^{†‡}	0.4583	0.6223
+VPRF	0.3519	0.9630	0.3866 ^{†‡}	0.5060 ^{†‡}	0.2974 ^{†‡}	0.6839 ^{†‡}	0.1749 ^{†‡}	0.4181	0.5798 ^{†‡}	0.9467	0.6923 ^{†‡}	0.1203 ^{†‡}	0.4138	0.6063
+HyDE	0.3251 [†]	<u>0.9573[†]</u>	0.4020 ^{†‡}	0.5259	0.3818 [†]	<u>0.7336</u>	0.1833 [†]	0.4181	0.7465	<u>0.9627</u>	<u>0.7614</u>	<u>0.1392</u>	<u>0.4667</u>	<u>0.6228</u>
+CoT	0.3236 ^{†‡}	0.9467 ^{†‡}	0.3785 ^{†‡}	0.4755 ^{†‡}	0.3728 ^{†‡}	0.7081 ^{†‡}	0.1675 ^{†‡}	0.3927 ^{†‡}	0.7012 ^{†‡}	<u>0.9593</u>	0.5710 ^{†‡}	0.0982 ^{†‡}	0.4191	0.5968
+Lamer	0.3086 ^{†‡}	0.9410 ^{†‡}	0.4032 ^{†‡}	0.4682 ^{†‡}	0.3868 [†]	0.7230 [†]	0.1739 ^{†‡}	0.3987 ^{†‡}	0.7046 ^{†‡}	0.9523	0.6708 ^{†‡}	0.1117 ^{†‡}	0.4413	0.5992
+GPRF	<u>0.3285[†]</u>	0.9481 [†]	0.4442	<u>0.5355</u>	<u>0.4323</u>	0.7469	<u>0.1893</u>	<u>0.4239</u>	<u>0.7404</u>	<u>0.9593</u>	0.7642	<u>0.1396</u>	0.4832	0.6256
BGE	<u>0.4534</u>	<u>0.9915</u>	0.4078 [†]	0.5301	<u>0.4064</u>	0.7415	<u>0.2168</u>	0.4957	0.7394	0.9633 [†]	0.7802 [†]	0.1407	<u>0.5007</u>	0.6438
+VPRF	0.4383 ^{†‡}	0.9908	0.3805 ^{†‡}	0.5009 ^{†‡}	0.2898 ^{†‡}	0.6601 ^{†‡}	0.2037 ^{†‡}	0.4905 [†]	0.6080 ^{†‡}	0.9433 ^{†‡}	0.8204	0.1489	0.4568	0.6224
+HyDE	0.4156 ^{†‡}	0.9872 [†]	0.4052 ^{†‡}	0.5284	0.3919 [†]	0.7338	0.2148	0.5003	0.7488	0.9767	<u>0.8056</u>	<u>0.1481</u>	0.4970	0.6457
+CoT	0.4520	0.9900	0.3953 ^{†‡}	0.5065 ^{†‡}	0.3841 ^{†‡}	0.7123 ^{†‡}	0.2001 ^{†‡}	0.4656 ^{†‡}	0.7395	0.9600 [†]	0.7530 ^{†‡}	0.1347 ^{†‡}	0.4873	0.6282
+Lamer	0.4426 ^{†‡}	0.9900	<u>0.4101[†]</u>	0.4922 ^{†‡}	0.3751 ^{†‡}	0.7221 [†]	0.2158	0.4778 ^{†‡}	0.7428	0.9617	0.7856	0.1394 ^{†‡}	0.4953	0.6305
+GPRF	0.4542	0.9922	<u>0.4285</u>	<u>0.5393</u>	0.4119	<u>0.7395</u>	<u>0.2198</u>	0.4960	<u>0.7482</u>	0.9700	0.7909	0.1430	0.5089	0.6467

Table 4: Ablation study on the impact of different training stages with Llama. The best and second-best methods of each retriever are marked in bold and underlined, respectively. "Vanilla" denotes that the model is not trained.

Retriever	Method	MS dev		DL 19		DL 20	
		NDCG@10	R@100	NDCG@10	R@100	NDCG@10	R@100
BM25	Vanilla	0.2360	0.6651	0.6182	0.4964	0.5751	0.5624
	SFT-only	0.2511	0.6726	0.6280	0.4765	0.5890	0.5542
	RL-only	<u>0.3061</u>	<u>0.7382</u>	<u>0.6598</u>	<u>0.5389</u>	<u>0.6480</u>	<u>0.5689</u>
	GPRF	<u>0.3208</u>	0.7486	<u>0.6917</u>	0.5401	0.6707	0.5849
E5	Vanilla	0.3361	0.7979	0.6631	0.4780	0.6384	0.5316
	SFT-only	0.3677	0.8530	0.7183	<u>0.5380</u>	0.6978	0.5948
	RL-only	<u>0.4219</u>	<u>0.8806</u>	<u>0.7209</u>	0.5373	<u>0.7432</u>	<u>0.6145</u>
	GPRF	<u>0.4283</u>	0.8852	<u>0.7228</u>	0.5405	0.7585	0.6205
BGE	Vanilla	0.3693	0.8496	0.7513	0.5488	0.7293	0.5719
	SFT-only	0.3842	0.8676	<u>0.7523</u>	0.5619	<u>0.7375</u>	0.5948
	RL-only	<u>0.4183</u>	<u>0.8763</u>	0.7418	0.5328	0.7222	0.5635
	GPRF	<u>0.4262</u>	0.8846	<u>0.7555</u>	<u>0.5560</u>	0.7384	<u>0.5778</u>

also outperforms strong GRF baselines such as HyDE and Lamer, with gains most evident on domain-shifted datasets like DBpedia and FiQA-2018, still achieving the highest overall scores in both NDCG@10 and R@100. An additional advantage of GPRF lies in its domain-agnostic prompt design. While GRF methods like HyDE and Lamer require carefully crafted prompts tailored to different tasks or domains, GPRF employs a single unified prompt across all datasets. This not only simplifies deployment but also demonstrates the robustness and generalizability of our approach in cross-domain retrieval scenarios.

Figure 3: Bucket-based analysis on MS dev. Queries are grouped into buckets based on their baseline BM25 performance, and the NDCG@10 improvement of three feedback-based methods, RM3, Lamer, and GPRF, is evaluated within each group. From left to right, the relevance of top-retrieved feedback documents in each group increases.



In summary, our GPRF framework demonstrates consistent and substantial improvements over classical PRF methods and recent GRF baselines across both in-domain and out-of-domain settings. In in-domain settings, GPRF leverages retrieval grounding to achieve significant gains in precision-oriented metrics (i.e., NDCG@10), while maintaining strong recall. In out-of-domain scenarios, it further shows superior robustness and adaptability, outperforming other GRF baselines without the need for task- or domain-specific prompt engineering. It not only highlights its strong capabilities and superior performance but also underscores its generalizability,

Figure 4: A case study shows that our training framework can effectively alleviate the noisy feedback problem.

Question: definition of dignity for kids	Feedback Passage: The author errors in only looking at one definition of respect, that which is earned. Other types of respect are due to position or existence. Respect for a person as merely being a human being is dignity: Dignity is our inherent value and worth as human beings; everyone is born with it.	Vanilla: Definition of dignity, particularly for kids, and exploring different types of respect and their relevance to being a human being as an inherent value and worth.
Ground Truth (not retrieved in the first stage): Full Definition of DIGNITY. 1.: the quality or state of being worthy, honored, or esteemed. 2. a: high rank, office, or position b: a legal title of nobility or honor.....		GPRF: Definition of dignity for kids: Dignity is the quality or state of being worthy, honored, or esteemed

demonstrating the ability to fully leverage the powerful natural language understanding and generation capacities of LLMs to achieve robust adaptation across different tasks and models, thereby effectively addressing the model assumption problem. Besides, compared with other GRF methods, GPRF does not generate additional intermediate results or introduce extra modules, and thus maintains comparable efficiency. Taken together, these results establish GPRF as a practical and effective solution for query rewriting, capable of delivering reliable performance across diverse retrieval settings.

5.2 Ablation Study

To demonstrate the effectiveness of our utility-oriented training pipeline in tackling noisy feedback, we conduct ablation experiments by training the LLM with each stage individually, and the results are reported in Table 4. Compared to the vanilla setting without training, incorporating either SFT or RL alone yields substantial improvements, confirming the necessity of training for robust query reformulation: SFT provides a strong initialization, while RL contributes more significantly to performance gains. Combining both within the full GPRF pipeline achieves the best overall performance, generally outperforming the single-stage and non-training variants. By filtering low-utility generations through rejection sampling, grounding reformulations with SFT, and reinforcing utility-driven objectives via RL, GPRF reduces the negative impact of irrelevant or misleading feedback (we will further discuss later in §5.3), thereby alleviating the fragile relevance assumption. This allows the system to generate reformulations faithful to user intent.

5.3 Relevance Assumption Analysis (RQ2)

To examine whether GPRF can effectively alleviate the relevance assumption, we conduct a bucket-based analysis on the MS dev dataset. Queries are grouped into buckets according to their baseline BM25 performance ordered from low to high, and we compare three feedback-based rewriting methods, RM3, Lamer, and GPRF, on each bucket, as shown in Figure 3. The results reveal a clear trend: while other methods exhibit limited or even negative gains on queries that already perform well (rightmost buckets), GPRF consistently yields substantial improvements, especially in the more challenging buckets where the baseline retriever performs poorly (left regions, indicating lower-quality feedback). In particular, the median NDCG@10 improvement of GPRF is significantly higher than RM3 and Lamer, indicating its stronger resilience to noisy or unreliable feedback.

A case study between the vanilla model without training and GPRF further (as shown in Figure 4) illustrates how GPRF alleviates

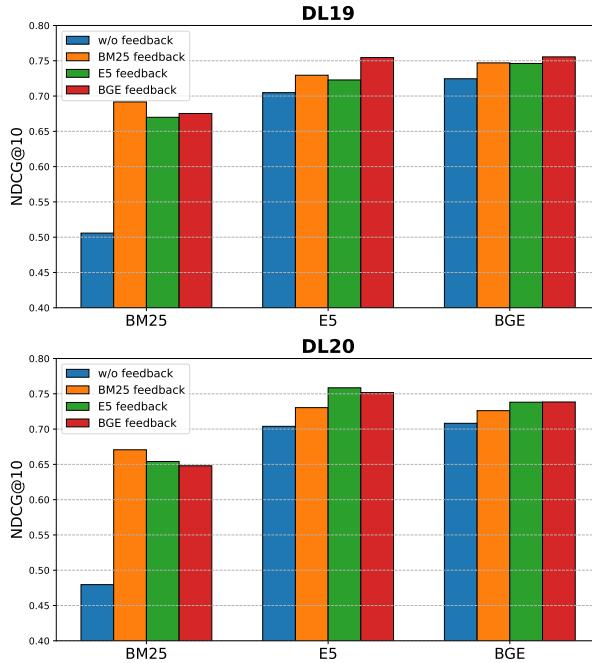
the relevance assumption. For the given query "definition of dignity for kids", the truly relevant definition of "dignity" is not retrieved, and the top-ranked feedback document contains only partial or noisy signals. It can be observed that the vanilla model is distracted by the feedback and produces a query expansion dominated by the notion of "respect", which drifts away from the canonical definition. In contrast, GPRF can utilize the feedback context more effectively, filtering out spurious associations and grounding the reformulation in the core semantic meaning of "dignity". This shows that our utility-oriented training pipeline enables the model to not only extract useful signals from noisy feedback documents but also to integrate them with its internal knowledge, producing a precise and faithful reformulation aligned with the user's intent. It demonstrates how GPRF alleviates the fragile relevance assumption and ensures robustness in realistic retrieval scenarios where top-ranked documents may not be fully reliable.

To sum up, GPRF is capable of generating robust and semantically grounded reformulations even when the initial retrieval results contain substantial noise. In these realistic situations, traditional PRF and GRF methods tend to fail. By integrating retrieval grounding with our utility-oriented training pipeline, GPRF learns to selectively leverage relevant evidence while suppressing misleading information. Consequently, it effectively relaxes the dependence on the relevance assumption, maintaining stable gains across varying retrieval quality levels and demonstrating its robustness in real-world noisy retrieval environments.

5.4 Cross-model Experiment (RQ3)

On the other hand, to further verify that GPRF can effectively address the model assumption problem, we conduct cross-model experiments, where different retrievers are used to provide feedback documents and employed for the final retrieval. As shown in Figure 5, the cross-model results on DL19 and DL20 demonstrate that GPRF consistently achieves strong performance even when the retriever used for providing feedback differs from the one used for final retrieval, showing only minor or no performance drops compared to the in-model setting. For instance, queries rewritten with BM25 feedback remain highly effective when evaluated with E5 or BGE, and using E5 as the feedback retriever still yields competitive results with BGE as the ultimate retriever. Compared to the capability of a retriever itself, the impact of feedback retrievers is relatively insignificant. Notably, even though BGE's retrieval results or reward signals are never used during training, GPRF still achieves competitive performance when evaluated with BGE (also shown in Table 2 and Table 3) or using it as the first-stage retriever.

Figure 5: Cross-model experimental results on DL19 and DL20 with Llama. The results of providing different retrievers with various feedback are reported. It can be observed that using varying feedback consistently improves the performance of different retrievers.



All these indicate that GPRF-generated reformulations are not tied to the embedding space of any particular retriever, enabling robust transferability across heterogeneous retrieval models, and thereby validating the effectiveness in overcoming the model assumption and the generalizability of GPRF.

6 Related Work

6.1 Ad-hoc Retrieval and Relevance Feedback

Ad-hoc retrieval, the task of selecting documents according to their relevance to a given query, has been a central problem in information retrieval [12, 28]. Traditional sparse retrieval methods, such as BM25 [24], often rely on exact lexical overlap between queries and documents, with relevance scores determined by term frequency (TF) and inverse document frequency (IDF) statistics. While these approaches are computationally efficient and interpretable, they inherently lack deeper semantic understanding, often failing when queries and documents use different surface forms to express the same concept. With the advent of pre-trained language models (PLMs) such as BERT and RoBERTa [7, 22], dense retrieval has emerged as a powerful alternative. Dense retrievers [9, 13, 16, 42, 43] map queries and documents into a shared semantic space, where relevance is measured by vector similarity. This paradigm enables retrieval beyond exact term matching, capturing semantic relations and paraphrases that sparse methods typically miss. Despite their differences, both sparse and dense retrieval fundamentally

operate within their respective vector spaces: sparse retrieval in high-dimensional lexical space, and dense retrieval in dense semantic space. Their reliance on model-specific representations constrains traditional feedback methods such as PRF and VPRF [1, 19], which remain tightly coupled to a specific retrieval model (i.e., the model assumption). In contrast, our proposed GPRF framework reformulates queries directly in natural language, providing a model-agnostic bridge that connects both sparse and dense retrieval, enabling more generalizable and transferable improvements across heterogeneous retrieval paradigms.

6.2 Large Language Models (LLMs) for Query Rewriting

Recent advances in large language models (LLMs) [2, 4, 8] have opened new opportunities for information retrieval (IR) modules, including rewriter, retriever, reranker, and reader [45]. In this paper, we mainly focus on the query rewriter module. Representative work includes HyDE [10], which generates hypothetical documents as supplements for retrieval, and Query2Doc [39], which leverages few-shot prompting to produce answer-like passages as query expansions. Similarly, chain-of-thought (CoT) [40] prompting has been applied to query rewriting [14], encouraging models to provide intermediate reasoning steps that lead to more interpretable reformulations. These methods demonstrate that generation-based pseudo-documents can significantly improve retrieval coverage and recall. Meanwhile, drawing from retrieval-augmented generation (RAG) [18, 29, 31, 32], other studies integrate retrieved documents into LLM prompting to mitigate generation bias and enhance factual accuracy. For instance, LameR [27] adopts a retrieve–answer–retrieve framework to guide LLM generation with pseudo-answers. Although such methods can mitigate the hallucination problem of LLM outputs to some extent, their effectiveness is quite limited, especially when the retrieved results contain noise, as LLMs are vulnerable to noisy inputs [35, 37]. Building on this line of research, by reformulating queries with retrieval augmentation and leveraging a tailored training pipeline, our GPRF achieves greater robustness and generalizability than existing approaches while better relaxing the fragile relevance assumption.

7 Conclusions

In this paper, we revisit the limitations of classical query rewriting approaches, including PRF and VPRF, and recent LLM-based generative methods. We identify two core challenges, the relevance assumption and the model assumption, that hinder their robustness and generalizability. To address these issues, we propose Generalized Pseudo-Relevance Feedback (GPRF), unifying the strengths of retrieval-based grounding and LLM-driven generation, thus resolving the model assumption. We also introduce a utility-oriented training pipeline to equip LLMs with the defense against noisy feedback and strengthen reformulation quality, thus alleviating the relevance assumption. In future work, we plan to explore more effective and efficient training strategies and extend GPRF to multimodal and interactive retrieval scenarios, further enhancing its applicability in real-world search systems.

References

- [1] Nasreen Abdul-Jaleel, James Allan, W Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D Smucker, and Courtney Wade. 2004. UMass at TREC 2004: Novelty and HARD. (2004).
- [2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).
- [3] Qingyao Ai, Yongfeng Zhang, Keping Bi, Xu Chen, and W Bruce Croft. 2017. Learning a hierarchical embedding model for personalized product search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 645–654.
- [4] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609* (2023).
- [5] 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* (2016).
- [6] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Jimmy Lin, Ellen M Voorhees, and Ian Soboroff. 2025. Overview of the TREC 2022 deep learning track. *arXiv preprint arXiv:2507.10865* (2025).
- [7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 4171–4186.
- [8] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv e-prints* (2024), arXiv-2407.
- [9] Yan Fang, Jingtao Zhan, Qingyao Ai, Jiaxin Mao, Weihang Su, Jia Chen, and Yiqun Liu. 2024. Scaling laws for dense retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1339–1349.
- [10] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1762–1777.
- [11] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-rl: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025).
- [12] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM international conference on information and knowledge management*. 55–64.
- [13] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118* (2021).
- [14] Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Benderky. 2023. Query expansion by prompting large language models. *arXiv preprint arXiv:2305.03653* (2023).
- [15] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.
- [16] Vladimir Karpukhin, Barlas Onguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering.. In *EMNLP* (1). 6769–6781.
- [17] Victor Lavrenko and W Bruce Croft. 2017. Relevance-based language models. In *ACM SIGIR Forum*, Vol. 51. ACM New York, NY, USA, 260–267.
- [18] 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.
- [19] Hang Li, Ahmed Mourad, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. 2023. Pseudo relevance feedback with deep language models and dense retrievers: Successes and pitfalls. *ACM Transactions on Information Systems* 41, 3 (2023), 1–40.
- [20] Hang Li, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. 2025. LLM-VPRF: Large Language Model Based Vector Pseudo Relevance Feedback. *arXiv preprint arXiv:2504.01448* (2025).
- [21] Minghan Li, Xinxuan Lv, Junjie Zou, Tongna Chen, Chao Zhang, Suchao An, Ercong Nie, and Guodong Zhou. 2025. Query Expansion in the Age of Pre-trained and Large Language Models: A Comprehensive Survey. *arXiv preprint arXiv:2509.07794* (2025).
- [22] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
- [23] Iain Mackie, Shubham Chatterjee, and Jeffrey Dalton. 2023. Generative relevance feedback with large language models. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*. 2026–2031.
- [24] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.
- [25] Joseph John Rocchio Jr. 1971. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing* (1971).
- [26] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxian Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300* (2024).
- [27] Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Tianyi Zhou, and Daxin Jiang. 2023. Large language models are strong zero-shot retriever. *arXiv preprint arXiv:2304.14233* (2023).
- [28] Weihang Su, Qingyao Ai, Xiangsheng Li, Jia Chen, Yiqun Liu, Xiaolong Wu, and Shengluan Hou. 2024. Wikiformer: Pre-training with structured information of wikipedia for ad-hoc retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 19026–19034.
- [29] Weihang Su, Qingyao Ai, Jingtao Zhan, Qian Dong, and Yiqun Liu. 2025. Dynamic and parametric retrieval-augmented generation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 4118–4121.
- [30] Weihang Su, Yichen Tang, Qingyao Ai, Changyue Wang, Zhijing Wu, and Yiqun Liu. 2024. Mitigating entity-level hallucination in large language models. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*. 23–31.
- [31] Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. DRAGIN: dynamic retrieval augmented generation based on the information needs of large language models. *arXiv preprint arXiv:2403.10081* (2024).
- [32] Weihang Su, Yichen Tang, Qingyao Ai, Junxi Yan, Changyue Wang, Hongning Wang, Ziyi Ye, Yujia Zhou, and Yiqun Liu. 2025. Parametric retrieval augmented generation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1240–1250.
- [33] Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. 2024. Unsupervised real-time hallucination detection based on the internal states of large language models. *arXiv preprint arXiv:2403.06448* (2024).
- [34] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663* (2021).
- [35] Yiteng Tu, Weihang Su, Yujia Zhou, Yiqun Liu, and Qingyao Ai. 2025. Robust Fine-tuning for Retrieval Augmented Generation against Retrieval Defects. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1272–1282.
- [36] Ellen M Voorhees. 1994. Query expansion using lexical-semantic relations. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*. Springer, 61–69.
- [37] Fei Wang, Xingchen Wan, Ruoxi Sun, Jiefeng Chen, and Sercan Ö Arik. 2024. Asstute rag: Overcoming imperfect retrieval augmentation and knowledge conflicts for large language models. *arXiv preprint arXiv:2410.07176* (2024).
- [38] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Dixin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533* (2022).
- [39] Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. *arXiv preprint arXiv:2303.07678* (2023).
- [40] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.
- [41] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*. 641–649.
- [42] 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. *arXiv preprint arXiv:2007.00808* (2020).
- [43] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. Report: Contextualized text embeddings for first-stage retrieval. *arXiv preprint arXiv:2006.15498* (2020).
- [44] Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020. Knowledge-grounded dialogue generation with pre-trained language models. *arXiv preprint arXiv:2010.08824* (2020).

- [45] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107* (2023).