LLM and User Feedback Based Contarstive Learning Improves Retrieval-Augmented Generation When Question and Answer Domains Shift

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) attempts to mitigate the issue of outdated knowledge and hallucinations in large language models (LLMs) by retrieving real-time information for LLMs. Nevertheless, we observe that the domain of user questions undergoes rapid changes over time, resulting in a significant decrease in RAG performance. Meanwhile, existing methods either overlook the feedback present in the workflow or fail to fully utilize them to improve the RAG system. To this end, we propose a method that utilizes both LLM and User Feedback (LUF) to improve RAG performance with shifts in question domains and answer domains. With the framework designed to automatically extract diverse feedback signals from both LLM and user within the existing workflow, LUF can adjust to variations in questions and user preferences through updates to the retriever and document database, guided by three complementary training objectives derived from feedback-all without explicit annotations. Experiments on two tasks demonstrate that LUF significantly improves the accuracy of the retriever and the responses of the LLM. Compared to baselines, LUF provides more accurate responses aligned with different user preferences.

1 Introduction

Retrieval-Augmented Generation (RAG) (Lewis et al., 2021; Gao et al., 2024) tackles hallucinations resulting from the inability to access real-time information by employing an additional retriever to obtain the latest data. Existing RAG frameworks typically involve the following steps: first, an additional retriever is used to fetch relevant documents from a document database; then, an LLM is utilized to rerank and filter irrelevant documents; finally, the question is combined with the filtered documents and fed into an LLM to generate a response (Sachan et al., 2023).

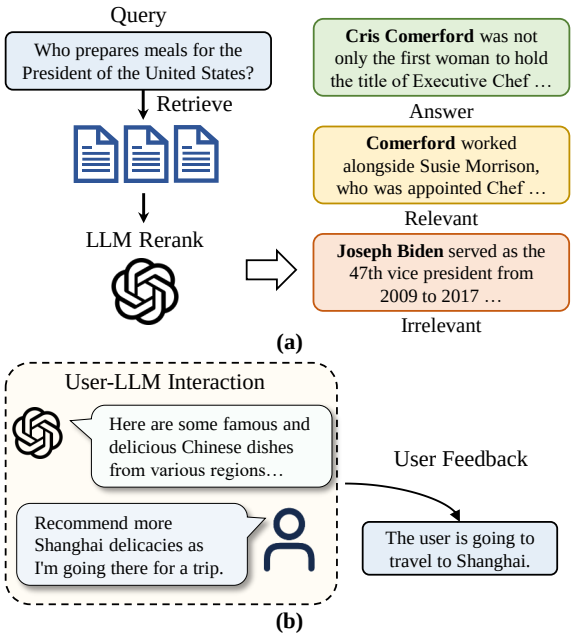


Figure 1: Illustration of the feedback signals already present in the existing workflow..

Previous research (Yoran et al., 2024) has shown that LLMs exhibit a notable decrease in answer accuracy when irrelevant documents are retrieved. Thus, the output of the LLMs is highly dependent on the precision of the retriever and the quality of the document database. Several studies have attempted to improve the performance of RAG retrieval, including constructing better training data for retriever training (Yu et al., 2023), using data augmentation techniques such as query rewriting or expansion (Wang et al., 2023; Ma et al., 2023), and employing multiple retrievals or active retrieval methods (Borgeaud et al., 2021; Jiang et al., 2023). However, these studies failed to recognize RAG's real-time potential as their retrievers and document databases were frozen during testing.

However, our research shows that domain shifts in questions and answers can significantly degrade retriever performance, even when the document

database remains unchanged. In our experiments, widely adopted dense retrievers like Contriever and DPR experience up to an 8% drop in retrieval accuracy on unseen datasets, resulting in a 6% decrease in LLM answer accuracy. This suggests that static retrievers struggle to adapt as user queries evolve over time. While some recent RAG methods incorporate feedback to mitigate this issue, most rely on explicit user annotations and overlook the fine-grained feedback already present in the existing workflow, such as those reranking results from LLM. Moreover, current research utilizes the same database to answer all questions, providing indistinguishable responses to different users, who often have varying expectations for the domain of the answers. For instance, when recommending courses, a computer science student would benefit from an answer of studying "Computer Systems", yet a biology student would find "General Biology" more appropriate.

Recognizing these limitations, we propose the LLM and User Feedback-based RAG (LUF), an RAG framework that autonomously adapts to domain shifts in questions and answers based on feedback. LUF initially follows the standard procedure of retrieving, reranking, and generating outputs. It then self-updates based on LLM and user feedback during user interaction. LUF initially follows the standard RAG pipeline of retrieving, reranking, and generating outputs. It then self-improves by incorporating both LLM feedback and user feedback during interaction. First, LUF prompts the LLM to label retrieved documents with fine-grained relevance signals, which are used to construct relevance-aware contrastive training samples. Second, we design three complementary contrastive objectives—strict supervision from answers, soft supervision from related content, and hard negative penalization—to guide retriever updates, enabling it to retrieve more useful evidence for unseen questions. Finally, LUF processes information emerging from user-LLM interactions through an agent-based pipeline that classifies feedback into factual updates and user-specific preferences. Notably, LUF operates entirely without explicit user annotations, relying instead on implicit supervision, which allows it to continuously adapt and enhance LLM outputs in evolving application scenarios.

Our key motivation is to improve the adaptability of RAG systems under domain shifts by leveraging fine-grained and reliable supervision signals already available in the RAG pipeline. Although LLMs may not always know the correct answers, they possess strong semantic understanding and can accurately assess which documents are helpful for generation. These reranking signals offer valuable supervision for retriever training. To fully exploit them, we consider three types of feedback as shown in Fig. 1 (a): the presence of answer-containing documents, graded relevance of partially helpful documents, and the rejection of irrelevant yet high-scoring ones. Each provides a distinct learning signal—ranging from precise supervision to softer or contrastive cues—that collectively enhance the retriever's ability to identify documents beneficial for generation. Beyond LLM feedback, user-LLM interactions often yield implicit signals about factual knowledge and user preferences, as shown in Fig. 1 (b). Incorporating such feedback helps the system adapt not only to new knowledge but also to individual user expectations.

LUF is compatible with any learning-based retriever and can be combined with other methods. Experiments across multiple datasets demonstrate that LUF improves both retriever accuracy and LLM generation in tasks such as question answering and multi-turn dialogues, achieving superior results compared to baselines. In summary, the contributions of this paper are:

- We propose LUF, a self-improving RAG framework that adapts to domain shifts without requiring labeled data, enabling the system to continuously refine its retriever and generator based on feedback signals observed during deployment.

- We design three complementary contrastive objectives based on LLM-provided pseudo-supervision, which serve as a central part of our feedback-driven training pipeline. Each objective is tailored to a specific type of LLM-derived feedback, enabling the retriever to learn from both precise and approximate supervision.

- We validate our approach on two tasks using two backbone LLMs, demonstrating consistent improvements in retrieval and generation performance. We also provide empirical analysis showing how feedback-driven updates help the system acquire new knowledge and personalize to user intent.

2 Method

In this section, we present a contrastive training framework that leverages feedback signals to improve RAG system. Specifically, Section 2.1 introduces how we utilize the capabilities of LLMs to construct relevance-labeled training samples. To help the retriever identify documents helpful for generation, we propose a set of learning objectives described in Section 2.2, built upon the relevance-labeled samples. In addition, Section 2.3 details how we incorporate user feedback through an agent-based pipeline for personalized knowledge updates.

2.1 Contrastive Learning Sample Construction from LLM Feedback

Every time a user poses a question q, the system first retrieves a set of candidate documents $\{d_i\}$ from the document database. To construct informative samples for contrastive learning, we prompt the LLM with a designed instruction to classify each retrieved document into one of three categories: documents that contain the correct answer ($\mathcal{D}^+(q)$), documents relevant but do not contain the answer ($\mathcal{D}^r(q)$), and irrelevant documents ($\mathcal{D}^-(q)$). For each relevant document $d \in \mathcal{D}^r(q)$, we further elicit from the LLM a scalar relevance score $sim'(q,d) \in [0,1]$, reflecting its estimated usefulness in answering q. This LLM-guided labeling and scoring provides efficient pseudo supervision that captures both direct answers and supporting context without requiring manual annotation.

After obtaining relevance-labeled samples, we further construct training batches by clustering queries based on their similarity. For each query q_i, we identify its batch assignment by selecting the cluster that maximizes relevant document overlap with other queries:

$$\text{Batch}(q_i) = \arg\max_{B} \sum_{q_j \in B} \left| \mathcal{D}^{+\cup r}(q_i) \cap \mathcal{D}^{+\cup r}(q_j) \right| \tag{1}$$

This relevance-guided batching groups semantically related questions together, with their associated documents naturally serving as in-batch negatives, which in turn enhances contrastive supervision by promoting fine-grained semantic discrimination within each batch.

In practice, we also employ a multi-level reranking mechanism to better select and prioritize documents before sample construction. Detailed descriptions of the multi-level reranking are provided in Appendix A.2.

2.2 LLM Feedback-Based Contrastive Training

Grounded in the relevance-labeled samples and clustered training batches, we design three complementary objectives that leverage this feedback structure to update the retriever:

Strict Label Contrastive Learning. We treat answer-containing documents as strong positive examples for the query. Given a query q and its associated answer-containing documents $\mathcal{D}^a(q)$, we minimize a standard contrastive loss that pulls these documents closer in the embedding space:

$$\mathcal{L}_{\text{s}}(q) = -\log \frac{\sum_{d^+ \in \mathcal{D}^+(q)} \exp\left(\text{sim}\left(q, d^+\right)/\tau\right)}{\sum_{d \in \mathcal{B}} \exp\left(\text{sim}\left(q, d\right)/\tau\right)} \tag{2}$$

Here, $\mathcal{B}$ denotes all documents in the current batch (including in-batch negatives), and τ is a temperature scaling parameter.

Flaccid Label Distillation. While strict label contrastive learning focuses on answer-containing documents, we aim to further utilize the LLM-provided graded relevance signals for documents in $\mathcal{D}^r(q)$ that are relevant but not fully answer-bearing. Instead of treating these examples as uniform positives, we adopt a soft label distillation approach based on the scalar relevance scores $\sigma(q,d) \in [0,1]$ introduced in Section 2.1.

We train the retriever to align its predicted similarity distribution with these soft targets via a KL divergence objective:

$$\begin{aligned} \mathcal{L}_{\text{f}}(q) &= \text{KL}(P_{LLM} \,\|\, P_{Retriever}), \\ P(d) &= \frac{\exp(\text{sim}(q,d)/\tau)}{\sum_{d'} \exp(\text{sim}(q,d')/\tau)}. \end{aligned} \tag{3}$$

Hard Negative Contrastive Learning. In addition to utilizing relevant documents for positive supervision, we incorporate hard negatives to penalize high-scoring yet irrelevant documents. Specifically, we select documents from $\mathcal{D}^n(q)$ with high initial retrieval scores and apply a rank-weighted margin loss:

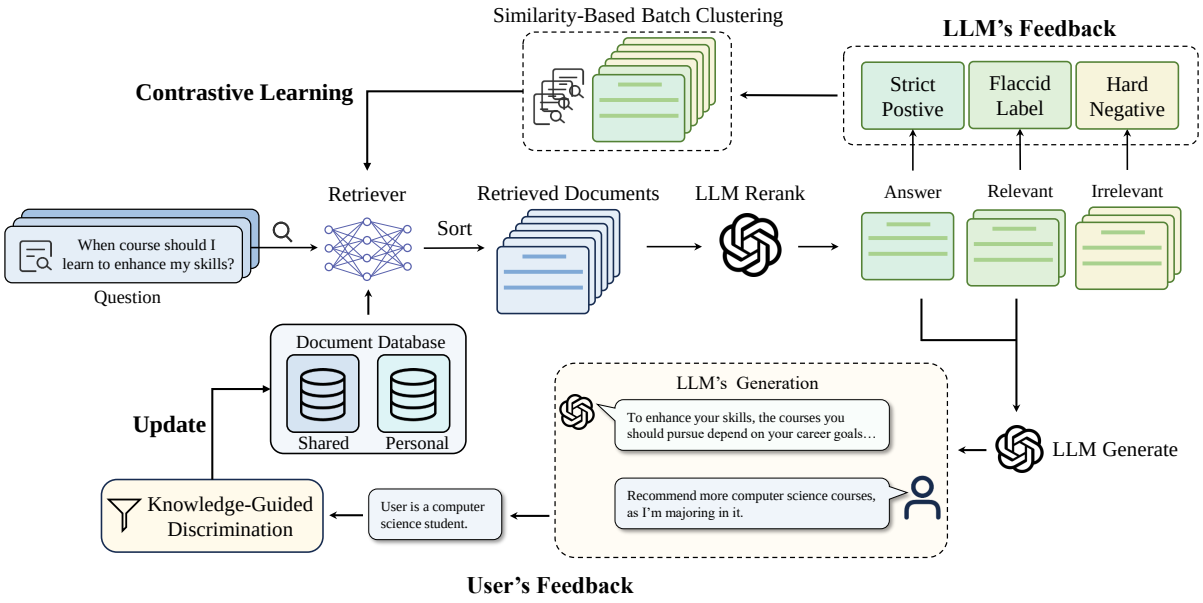


Figure 2: Framework of LUF, including the widely used RAG workflow and processors for LLM and user feedback.

$$\mathcal{L}_{\mathrm{h}}(q) = \sum_{d^- \in \mathcal{D}_q^-} w(d^-) \times \max\left(0,\, \gamma - \mathrm{sim}(q, d^+) + \mathrm{sim}(q, d^-)\right) \tag{4}$$

where $d^+ \in \mathcal{D}^a(q)$ is a strict positive, γ is a margin hyperparameter, and $w(q, d^-)$ is a weight based on the retrieval rank of d^-.

Overall Objective. We combine the three objectives into a weighted training loss: $\mathcal{L}_{\text{total}} = \lambda_1 \cdot \mathcal{L}_{\mathrm{s}} + \lambda_2 \cdot \mathcal{L}_{\mathrm{f}} + \lambda_3 \cdot \mathcal{L}_{\mathrm{h}}$, where λ_1, λ_2, and λ_3 balance the contributions of strict positives, soft relevance, and hard negatives.

2.3 User Feedback Handling

Beyond LLM-generated feedback, we also leverage information from user-LLM interactions to better align the system with individual preferences. To manage such feedback, we design an agent-based processing pipeline that categorizes feedbacks into two types: *factual updates*, which describe objective knowledge that applies to all users, and *user preferences*, which reflect user-specific preferences. Factual updates are verified and stored in a shared document database, ensuring the consistency and reliability of responses across users. In contrast, user preferences are stored in a separate user-specific database, enabling personalized responses. This separation helps the system maintain factual accuracy while supporting flexible customization for different users.

Since factual updates may be noisy or unreliable, we introduce a verification mechanism termed *Knowledge-Guided Discrimination*. It is based on the hypothesis that LLMs are less prone to error when evaluating new information in the context of existing knowledge, rather than in isolation. To implement this, our approach uses two sources of existing information: documents retrieved from the database and the LLM's own response generated without retrieval. Given these references, the agent is prompted to assess the correctness of the user feedback—reasoning based on the retrieved evidence and its own prior answer. Based on this judgment, and taking user feedback into account, the system then updates the document database through additions, deletions, or modifications as needed.

3 Experiment

3.1 Implementation Details

Base LLMs. We employed two language models representing different deployment scenarios: GPT-4o (OpenAI et al., 2024), a strong proprietary model, and Qwen3-32B (Yang et al., 2025), a recent open-source model with a smaller parameter scale. Table 1 reports the answer accuracy of both models without retrieval, where Qwen3-32B performs significantly worse than GPT-4o. We conduct our experiments on both models with their respective knowledge bases to demonstrate the generality and adaptability of LUF across different

	NQ	TQA	SQuAD
Qwen3	33.99	57.60	16.06
GPT-4o	59.42	67.33	27.66

Table 1: Accuracy without retrieval of GPT-4o and Qwen3.

model capacities and accessibility levels.

Datasets. We evaluate the effectiveness of LUF on two tasks that LLMs frequently encounter in real-world applications:

Question Answering is a knowledge-intensive task requiring the LLM to provide accurate answers to specific questions. We used four English datasets: Natural Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (TQA) (Joshi et al., 2017), SQuAD (Rajpurkar et al., 2016), and Web Questions (WebQ) (Berant et al., 2013), with articles extracted from Wikipedia as the initial document database as in previous work (Izacard et al., 2022a). Another Chinese dataset WebQA (Li et al., 2016), with original document database is used. For this task, we evaluate both the precision of the retriever and the answers generated by the LLM.

Multi-turn Dialogue State Tracking requires to extract user intents during interactions and provide corresponding responses. We use MultiMOZ 2.2 (Zang et al., 2020) and SGD (Rastogi et al., 2020) as our testbed, where the inputs are real dialogues between users and assistants which contains real user feedback. We follow the same evaluation protocol as in (Feng et al., 2023).

Our method. To simulate domain shift in questions, we use the contriever (Izacard et al., 2022a), which is pretrained only on Wikipedia and CCNET, without training on the training set, and is tested directly on the test set. The hyperparameter settings of LUF can be found in the Appendix A.

3.2 Baselines

We compare LUF with several methods that also utilize LLMs' feedback:

Retrieve and Rerank (R&R) (Zhuang et al., 2023): The widely-used approach which involves retrieving documents based on the user-provided question and then using a LLM to rerank the results.

Query Rewrite (Ma et al., 2023): Query rewriting modifies or reformulates the initial user query to improve retrieval accuracy. We use the same LLM employed as a reranker to perform query rewriting.

Query2Doc (Wang et al., 2023): Query2Doc first transforms a user query into a pseudo-document, then concatenates the original query with the pseudo-document to serve as a new query. We use the same LLM serving as a reranker to generate pseudo-documents.

RaFe (Mao et al., 2024): RaFe is a RAG method that also utilizes ranking feedback. Unlike LUF, RaFe additionally employs a query rewriter to improve retrieval accuracy and uses ranking feedback to update the query rewriter. To ensure alignment with LUF, we adopt the "Online Feedback" settings described in the original paper.

3.3 Improvement of Retriever Accuracy

The **R@5** in Table 2 shows the retrieval accuracy of different methods. **LUF w/o UF** denotes the setting where no user feedback is utilized, under which our method has access to exactly the same information as all baselines. Despite this, LUF consistently outperforms all baselines across both backbone LLMs, and maintains strong retrieval performance on datasets and document databases in different languages, highlights its robustness and effectiveness across diverse application scenarios. LUF yielded more substantial improvements on larger datasets, as they provide richer LLM feedback for updating. Nevertheless, even on smaller datasets like Web Questions—where only about 1,016 feedback instances were used to update the retriever—yet performance gains were still observed, demonstrating the system's ability to benefit from sparse feedback.

Notably, even weaker LLMs can provide valuable supervision within our framework. For example, Qwen3-32B, without access to retrieved documents, correctly answers fewer than 20% of questions on SQuAD. However, as shown in Table 2, it still provides effective feedback by assessing semantic relevance between questions and documents, and helps the retriever perform more accurate retrieval. This highlights that LUF leverages the LLM's general reasoning capabilities to generate feedback, rather than relying on its stored factual knowledge, making LUF applicable to open-source or smaller-scale LLMs with limited parametric knowledge. When using stronger LLMs like GPT-4o, which offer higher-quality reranking signals, the retriever benefits from richer supervision and achieves greater improvements.

Method	NQ		TriviaQA		SQuaD		WebQ		WebQA	
	QA	R@5	QA	R@5	QA	R@5	QA	R@5	QA	R@5
Qwen3-32B +	33.99	-	57.60	-	16.06	-	28.04	-	51.29	-
R&R	44.29	54.82	59.47	64.02	32.14	55.03	38.39	43.86	47.59	16.24
Query2Doc	45.83	58.38	59.75	64.20	32.85	55.47	39.63	46.43	49.21	18.02
Query Rewrite	43.47	53.5	58.41	62.86	32.66	55.6	38.94	44.8	51.46	19.64
RaFe	45.09	56.46	59.97	64.66	32.40	54.25	39.48	46.28	48.02	21.99
LUF w/o UF	**47.06**	**58.8**	**61.52**	**66.52**	**34.2**	**59.59**	**39.77**	**46.43**	**52.98**	**22.62**
LUF w/ UF	47.29	59.16	61.97	67.29	34.56	60.13	39.97	46.62	53.41	23.31
GPT-4o +	59.42	-	67.33	-	27.66	-	52.37	-	63.14	-
R&R	63.38	59.86	64.27	67.05	34.95	58.28	54.82	51.67	61.89	20.57
Query2Doc	64.46	61.00	63.89	67.75	37.28	59.73	55.76	52.07	62.97	21.30
Query Rewrite	64.18	59.81	64.56	67.68	37.09	60.33	56.12	52.21	63.55	22.19
RaFe	64.57	61.66	68.03	69.12	37.36	59.67	55.09	51.72	63.22	21.73
LUF w/o UF	**67.92**	**64.74**	**68.98**	**70.08**	**40.76**	**63.70**	**56.74**	**52.81**	**66.04**	**27.31**
LUF w/ UF	68.17	64.99	69.42	70.60	41.13	64.36	56.84	52.90	66.60	27.41

Table 2: The accuracy of the answers provided by the LLM, with "QA" representing responses from the LLM that contain the correct answer. "UF" means incorporating user feedback.

Method	NQ	TQA	SQuAD
Direct	92.02	89.14	87.79
Ours	98.73	98.34	97.71

Table 3: Accuracy of judging the correctness of user feedback. "Direct" means directly asking the LLM to make the judgment.

3.4 Improvement of LLMs' Response in Question Answering

We further evaluate how LUF affects LLMs' outputs on all datasets. Specifically, we simulated real-world scenarios where users provide feedback. Based on the questions and corresponding answers in each dataset, we used GPT-4o to generate a dialogue containing relevant information about the questions to mimic user feedback. For each dataset, 40% of the questions were paired with correct feedback, 40% with incorrect feedback, and 20% with feedback unrelated to the dataset questions, to evaluate the robustness against incorrect user feedback. Details of the feedback and question simulation are provided in Appendix A.1.

Table 2 presents the performance of all methods, with LUF delivering the most accurate responses across all datasets. Without incorporating user feedback, LUF 's superior retrieval accuracy provided the LLMs with more precise evidence, resulting in more accurate answers. Notably, under comparable retrieval accuracy—such as on Natural Questions, where LUF and Query2Doc perform similarly—LUF still produces significantly more accurate responses. This may be attributed to the flaccid label contrastive learning in LUF, which encourages the retriever to retrieve not only answer-containing documents but also additional relevant context. After incorporating both correct and incorrect user feedback, LUF effectively extracts and filters useful signals from the correct feedback to update the document database, leading to improvements in both retrieval and answering accuracy.

Our proposed Knowledge-Guided Discrimination effectively prevents incorrect user feedback from polluting the overall system. Table 3 compares the accuracy of letting LLM directly judge correctness of user feedback and using LUF 's discriminate-and-classify modules. On all three datasets, LUF correctly judged over 97% of the feedback, significantly outperforming the direct judgment. In our experiments, direct judging feedback correctness introduced erroneous updates, leading to degradation in generation quality.

3.5 Improvement of LLMs' Response in Multi-turn Dialogues

To evaluate the impact of LUF on LLM outputs in dialogue scenarios, we conducted tests on two dialog state tracking datasets. The test samples are conversations between the user and assistant, containing real user feedback. Each test sample is paired with an annotated user intents, the LLM was required to comprehend all user's intents and provide corresponding responses. Fol-

Method	MultiWOZ 2.2		SGD	
	JGA	AGA	JGA	AGA
Qwen3+	54.30	86.53	73.20	84.71
R&R	55.90	87.89	73.11	84.06
Query2Doc	57.73	88.85	74.43	86.07
Query Rewrite	59.68	91.40	74.59	85.33
RaFe	59.60	91.68	74.98	86.84
LUF	**60.67**	**92.81**	**77.27**	**88.95**
GPT4o+	60.82	92.73	79.43	90.87
R&R	61.39	93.30	80.23	91.67
Query2Doc	61.73	92.71	80.67	91.19
Query Rewrite	61.85	93.52	80.85	91.63
RaFe	61.45	93.90	79.95	91.29
LUF	**62.83**	**94.67**	**81.51**	**92.55**

Table 4: The accuracy of the LLM responses in multi-turn dialogues.

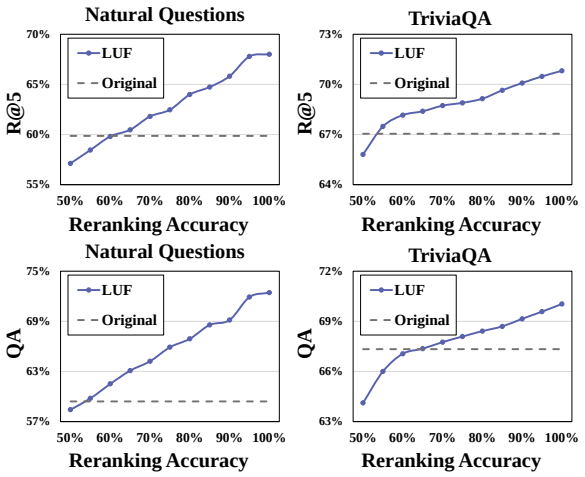


Figure 3: R@5 of LUF with different reranking accuracy.

lowing the evaluation setup in (Feng et al., 2023), we adopted the same prompting strategy and assessment method to evaluate whether the LLM correctly understood and retained the user's intents. From Table 4, we observed that while other methods provided minimal improvements, LUF-generated responses better satisfied user requirements. This improvement stems from LUF's ability to distill user feedback into structured preferences, persist them in the RAG document database, and incorporate them as auxiliary context during response generation—thereby enabling the LLM to better capture user intent from dialogue.

Each test sample consisted of a full multi-turn dialogue, allowing the LLM to infer user intents directly from the conversational context. However, we observed that the LLM occasionally failed to retain specific user requests across turns. Experimental results showed that LUF effectively reinforced user preferences through feedback integration, enabling the LLM to generate responses that remained consistent with user expectations throughout the conversation.

3.6 Further Investigations

Ablation Study

To understand the specific roles of different components in LUF, we conducted a series of ablation experiments. We tested the improvements brought by different strategies within LUF across three datasets, as shown in Table 5.

Adding LLM feedback (strict positive in Table 5) supervision leads to a clear improvement over the R&R standard procedure. This confirms that LLM-labeled answer documents provide valuable training signals for grounding the retriever. Incorporating flaccid label supervision and hard negative loss brings further gains, validating the effectiveness of objectives in LUF. Moreover, integrating user feedback yields additional improvements in both QA and R@5, demonstrating that the agent-based feedback processing module can extract useful information from noisy user interactions.

How Reliable Reranking Feedback is

LUF is updated based on reranking results from LLMs, so relies on the accuracy of LLM feedback. We simulated the impact of LLMs' feedback with different accuracy on the retriever on the NQ and TQA datasets. In Fig. 3, the accuracy refers judging single relevant documents, as there is no significant difference between different LLMs in judging irrelevant documents.

Fig. 3 illustrates that even when the LLM correctly identifies only 60% of relevant samples, its feedback still improves the retriever's performance, demonstrating LUF is highly robust to feedback accuracy. This robustness allows smaller-scale LLMs to provide useful feedback to improve LUF, suggesting that our approach is broadly applicable across models with varying capacities.

How LUF Identifies Feedback

The results in Table 3 demonstrate how LUF identifies user feedback and safeguards the document database from contamination by erroneous information.

Compare with existing information: In the example shown in Fig. 4 (a), the evidences from document database and LLM provide a foundational

Method	Natural Questions		TriviaQA		SQuaD	
	QA	R@5	QA	R@5	QA	R@5
R&R	44.18	57.01	57.21	58.45	31.79	52.26
+Strict Postive	64.51	61.20	63.16	64.37	38.52	62.38
+Flaccid Label	65.19	61.68	63.98	65.44	39.01	63.85
+Hard Negative	67.92	64.74	68.98	70.08	40.76	63.70
+User Feedback	68.17	64.99	69.42	70.60	41.13	64.36

Table 5: Ablation study results.

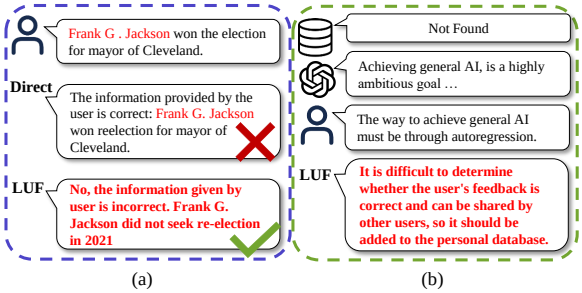


Figure 4: Examples of LUF's responses to different kinds of incorrect feedback.

reference. If judged directly, the LLM may be misled and add incorrect information to the database. However, by comparing with existing knowledge, LLM can identify misinformation and reject it.

Cautious handle unknown information: When faced with feedback that neither the LLM nor the document database can evaluate, LUF tends to prioritize adding such feedback to the user's personal database rather than the shared one, as shown in Fig. 4 (b).

4 Related Work

4.1 Retrieval-Augmented Generation

Research on RAG has advanced rapidly in recent years. Sparse retrieval techniques, such as BM25 (Sparck Jones, 1988), are simple and effective (Chen et al., 2024; Jiang et al., 2023; Ram et al., 2023). Dense retrieval methods like Dense Passage Retriever (DPR) (Karpukhin et al., 2020) demonstrate greater flexibility and adaptability (Izacard et al., 2022b; Shi et al., 2024; Sachan et al., 2024; Siriwardhana et al., 2023).

Recent studies have proposed various pre-retrieval and post-retrieval enhancement strategies. Pre-retrieval enhancement strategies, such as Query2doc (Wang et al., 2023), Hypothetical Document Embedding (HyDE) (Gao et al., 2023), and Query Rewriting (Ma et al., 2023) improve the relevance of retrieval results by reorganizing or expanding queries. Post-retrieval enhancement strategies, such as R2G (Sachan et al., 2023), filter irrelevant information by reranking retrieval results. These methods use static retrievers and overlook the role of feedback.

4.2 Feedback for Language Models

Feedback has been widely used in NLP and applied to many traditional tasks, such as question answering (Li et al., 2022; Harabagiu et al., 2001), text summarization (Nguyen et al., 2022; Liu et al., 2023), and machine translation (Saluja et al., 2012).

We note that several recent studies have also attempted to improve RAG using feedback too: Pistis-RAG (Bai et al., 2024) focuses on utilizing explicitly provided feedback from users in the form of labels (e.g., copying or disliking certain content) to improve retrieval performance. RaFe (Mao et al., 2024) utilizes feedback to improve query rewriting, while InstructRAG (Wei et al., 2024) employs feedback to train LLMs. Both approaches only utilize feedback from LLMs, whereas LUF also considers user feedback. RaFe focuses more on enhancing query rewriter performance through feedback, while LUF's motivation is to adapt the entire RAG system to changes in questions and answers, so we adopt the most widely used RAG process without query rewriter. InstructRAG requires training the LLM, which is computationally costly and does not fit our use case.

5 Conclusion

To address the performance degradation of RAG on unseen questions, we propose an framework called LUF based on user and LLM feedback. By updating the retriever and the document database, both the retriever and the LLM adapt to the shifts in question and answer domains, thereby improving their ability to provide responses that align with user preferences. Experiments conducted on two tasks demonstrate the effectiveness of our method.

6 Limitations

In this work, we primarily conducted evaluations on the question-answering and multi-turn dialogue state tracking, the performance of LUF on other tasks such as commonsense reasoning and open-domain summarization remains unknown. Besides the LLMs mentioned in the paper, we also tested smaller LLMs like Qwen3-7b-chat. For models with such smaller parameters, their reranking accuracy was below the minimum threshold required to improve the retriever, and LUF did not provide any meaningful enhancement.

7 Ethics Statement

In this study, we utilized publicly available datasets that do not contain any personal or private information, ensuring full compliance with ethical guidelines. The prompts used and the outputs generated by the LLMs were selected to exclude any content that might be discriminatory, violent, or otherwise inappropriate. No personal data was collected throughout the experimental process, and the design and execution of the experiments pose no negative societal impact. Therefore, this research adheres to ethical standards, with no risks of privacy infringement or harmful social consequences.

References

Yu Bai, Yukai Miao, Li Chen, Dawei Wang, Dan Li, Yanyu Ren, Hongtao Xie, Ce Yang, and Xuhui Cai. 2024. Pistis-rag: Enhancing retrieval-augmented generation with human feedback.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, and et al. 2021. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning*.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17754–17762.

Yujie Feng, Zexin Lu, Bo Liu, Liming Zhan, and Xiao-Ming Wu. 2023. Towards LLM-driven dialogue state tracking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 739–755, Singapore. Association for Computational Linguistics.

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)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Sanda Harabagiu, Dan Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan Bunsecu, Roxana Girju, Vasile Rus, and Paul Morarescu. 2001. The role of lexico-semantic feedback in open-domain textual question-answering. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 282–289, Toulouse, France. Association for Computational Linguistics.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022a. Unsupervised dense information retrieval with contrastive learning.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022b. Atlas: Few-shot learning with retrieval augmented language models.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, and et al. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Peng Li, Wei Li, Zhengyan He, Xuguang Wang, Ying Cao, Jie Zhou, and Wei Xu. 2016. Dataset and neural recurrent sequence labeling model for open-domain factoid question answering.

Zichao Li, Prakhar Sharma, Xing Han Lu, Jackie Cheung, and Siva Reddy. 2022. Using interactive feedback to improve the accuracy and explainability of question answering systems post-deployment. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 926–937, Dublin, Ireland. Association for Computational Linguistics.

Yixin Liu, Budhaditya Deb, Milagro Teruel, Aaron Halfaker, Dragomir Radev, and Ahmed Hassan Awadallah. 2023. On improving summarization factual consistency from natural language feedback. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15144–15161, Toronto, Canada. Association for Computational Linguistics.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

Shengyu Mao, Yong Jiang, Boli Chen, Xiao Li, Peng Wang, Xinyu Wang, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. 2024. RaFe: Ranking feedback improves query rewriting for RAG. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 884–901, Miami, Florida, USA. Association for Computational Linguistics.

Duy-Hung Nguyen, Nguyen Viet Dung Nghiem, Bao-Sinh Nguyen, Dung Tien Tien Le, Shahab Sabahi, Minh-Tien Nguyen, and Hung Le. 2022. Make the most of prior data: A solution for interactive text summarization with preference feedback. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1919–1930, Seattle, United States. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, and et al. 2024. Gpt-4 technical report.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8689–8696.

Devendra Singh Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2023. Improving passage retrieval with zero-shot question generation.

Devendra Singh Sachan, Siva Reddy, William Hamilton, Chris Dyer, and Dani Yogatama. 2024. End-to-end training of multi-document reader and retriever for open-domain question answering. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA. Curran Associates Inc.

Avneesh Saluja, Ian Lane, and Ying Zhang. 2012. Machine translation with binary feedback: a large-margin approach. In *Proceedings of the 10th Conference of the Association for Machine Translation in the Americas: Research Papers*, San Diego, California, USA. Association for Machine Translation in the Americas.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384, Mexico City, Mexico. Association for Computational Linguistics.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

Karen Sparck Jones. 1988. *A statistical interpretation of term specificity and its application in retrieval*, page 132–142. Taylor Graham Publishing, GBR.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *Conference on Empirical Methods in Natural Language Processing*.

Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2024. Instructrag: Instructing retrieval-augmented generation with explicit denoising.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai

Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context.

Zichun Yu, Chenyan Xiong, Shih Yuan Yu, and Zhiyuan Liu. 2023. Augmentation-adapted retriever improves generalization of language models as generic plug-in. In *Annual Meeting of the Association for Computational Linguistics*.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.

Shengyao Zhuang, Bing Liu, Bevan Koopman, and Guido Zuccon. 2023. Open-source large language models are strong zero-shot query likelihood models for document ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8807–8817, Singapore. Association for Computational Linguistics.

A Appendix

A.1 Prompt

Frequent LLM calls were required to generate responses in our experiment. In this section, we present the prompts used to accomplish different tasks.

Prompt A.1.1: Single Document Reranking

Please analyze whether this document can help answer this question, and provide your answer using single "yes" or "no" in the end.
Question:
$\langle Question \rangle$
Document:
$\langle Document \rangle$

Prompt A.1.2: Multiple Documents Reranking

Below is a question, along with $\langle Num\,of\,Docs \rangle$ documents that might be related to this question. Please judge if which document(s) are relevant to the question, and finally provide your answer using the document number + yes like "answer:1,yes" or "no".
Question:
$\langle Question \rangle$
Document 1:
$\langle First\,document \rangle$
Document 2:
$\langle Second\,ocument \rangle$
...

When performing rerank for a single document and multiple documents using LLM, to ensure that the LLM's response adheres to a fixed format suitable for code analysis, we use prompts as shown in Prompt A.1.1 and A.1.2.

Prompt A.1.3: User Feedback Stimulation

Here is a question and the correct answer is $\langle Answer \rangle$. Please simulate a user's statement in a conversation, and include the correct/incorrect information in this statement. Here is an example for your reference:
Question: Who invented the microscope?
Answer: Zacharias Janssen
Stimulated: I was reading about the history of scientific inventions, please tell me how Zacharias Janssen invented the microscope.
Question:
$\langle Question \rangle$
Answer:
$\langle Answer \rangle$
Stimulated:

In the experiment in Table 2, we used LLMs to simulate user feedback. Prompt A.1.3 asks the LLM to simulate correct/incorrect user feedback.

Prompt A.1.4: User Feedback Summarization

The following is a conversation between a user and an LLM. Did the user provide any meaningful information? If so, please summarize the information given by the user.

Prompt A.1.5: User Feedback Discriminate and Classify

The following are the document from Wikipedia, the LLM's answer, and the feedback provided by the user in the conversation with the LLM. Please use the first two as references to determine whether the information provided by the user is correct. There are two databases, one shared by all users and the other exclusive to the user. Please decide which database this information should be added to. At the end, give your answer like "correct, shared" or "correct, personal".
Document:
$\langle Document \rangle$
LLM answer:
$\langle LLM\ Answer \rangle$
User feedback:
$\langle User\ Feedback \rangle$

LUF first summarizes the user feedback to ex-

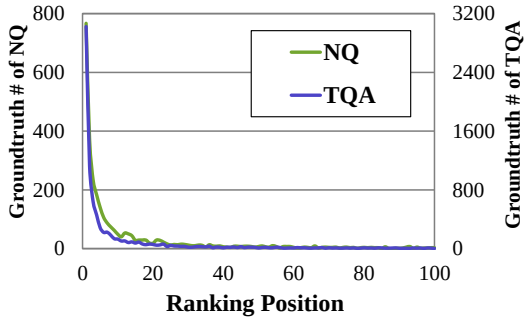


Figure 5: Retrieval results on Natural Questions and Trivia QA, as the rank position increases, the number of relevant documents decreases rapidly.

tract valid information, then performs discrimination and classification. Prompt A.1.4 is used to extract user feedback from the conversation, while Prompt A.1.5 simultaneously handles both the discrimination and classification of the feedback.

Prompt A.1.6: Get LLM Knowledge

The following is a piece of material that may be correct or incorrect. Please generate a paragraph on the same topic based on your knowledge. Material:$\langle Summarized\,User\,Feedback \rangle$

To accurately assess the correctness of user feedback, LUF need to compare it with the knowledge from the LLM. We use prompt A.1.6 to retrieve knowledge from the LLM.

Prompt A.1.7: Query Rewrite

Provide a better search query for retriever to search the given question.
Original question: What 2000 movie does the song "All Star" appear in?
New question: 2000 movie "All Star" song
Original question:$\langle Question \rangle$
New question:

Query rewrite uses the original prompt from previous work (Ma et al., 2023) where the LLM was used as a rewriter. as shown in Prompt A.1.7. Query2doc follows the original prompt in previous work (Wang et al., 2023).

A.2 Multi-level Reranking

Previous methods primarily adopt two forms of reranking: single reranking, where the LLM evaluates each retrieved document individually, and total reranking, which evaluates all documents together. Single reranking offers higher accuracy but incurs additional token computation costs, primarily due to the prompt. Conversely, total reranking yields the opposite results.

We observe that the results of retrieval typically follow a long-tail distribution, where higher-ranked positions have a much higher likelihood of providing relevant documents and thus have greater value compared to lower-ranked positions, as illustrated in Fig. 5. Therefore, we propose a multi-level reranking strategy, employing different reranking approaches for different positions. For higher-ranked positions, we use smaller reranking steps to enhance precision. As the process goes on, we gradually increase the step to reduce token consumption until at least one relevant document is found. It is a cost-effective solution that balances accuracy and computational overhead, allowing for accurate judgment without incurring excessive computational expenses.

A.3 Additional Experiment Results

A.3.1 Multi-level Reranking

For retrieval results from 500 randomly sampled questions, we applied the different reranking methods and calculated the average token consumption (both input and output) to achieve a similar R@5. "Single" means each document is individually assessed by the LLM, while "Every 5" and "Every 20" refer to a reranking step of 5 or 20 documents, respectively. Table 7 shows the result, the multi-level reranking consumes the fewest tokens while achieving similar R@5, can save the resources consumed by reranking.

A.3.2 Time Consumption

Table 6 shows the detailed time required by different methods to answer 3,000 questions, with all results tested on an dual RTX 4090 and Intel Xeon Gold 6430 machine. "Retrieve" refers to the time spent on dense retrieval, "LLM" indicates the time taken for LLM generation, and "Train" represents the time required for model training. For Query rewrite, we adopted the approach from (Mao et al., 2024), where the results of two rewritten queries are fused during retrieval.

Compared to the simplest R&R process, the additional time introduced by LUF is within an acceptable range. Moreover, aside from reranking, LUF does not require any preprocessing before retrieval, meaning the time from when the user asks

Method	Time to Retrieve 3,000 Questions (s)					
	TriviaQA			SQuAD		
	Retrieve	LLM	Train	Retrieve	LLM	Train
R&R	4,096.4	6,696.8	-	4,077.1	9,706.9	-
Query Rewrite	8,157.2	8,110.0	-	8,134.8	11,321.4	-
Query2Doc	4,092.9	8,931.2	-	4,069.6	11,635.4	-
TENT	8,162.4	6,709.2	66.0	8,138.4	9,704.8	66.0
LUF	4,091.4	7,639.5	65.8	4,069.5	12,261.8	65.6

Table 6: The detailed time required to retrieve 3,000 questions of different methods.

Method	Natural Questions			TriviaQA			SQuAD		
	R@5	Docs	Tokens	R@5	Docs	Tokens	R@5	Docs	Tokens
Single	55.75	39	3,257.6	57.64	38	3,313.7	55.09	38	3,174.1
Every 5	54.11	75	5,899.2	57.51	75	5,316.9	54.35	80	5,448.8
Every 20	55.44	80	4,920.1	57.95	80	5,004.3	55.02	80	4,851.9
Multi-level	55.78	40	**2,816.4**	57.9	40	**2,880.9**	55.08	40	**2,777**

Table 7: The number of tokens consumed by different reranking methods.

Method	Web Questions			WebQA		
	R@5	Docs	Tokens	R@5	Docs	Tokens
Single	49.16	35	2,853.3	18.68	35	2,069.6
Every 5	47.64	70	5,129.8	18.29	75	2,406.5
Every 20	49.02	80	4,933.5	18.72	60	1,623.9
Multi-level	49.26	40	2,791.6	18.72	40	1,183.0

Table 8: The number of tokens consumed by different reranking methods on Web Questions and WebQA.

Dataset	Test Set Size		Number of Documents
	Qwen3	GPT-4o	
Natural Questions	3,604	3,610	21,015,324
TriviaQA	11,311	11,313	21,015,324
SQuAD	10,570	10,570	21,015,324
Web Questions	2,032	2,032	21,015,324
WebQA	3,019	3,024	3,024

Table 9: The number of test questions and documents of different dataset.

Dataset	Update Interval	Epochs	Learning Rate
Natural Questions	2,000	4	5e-06
TriviaQA	4,000	4	1e-05
SQuAD	4,000	4	1e-05
Web Questions	1,000	1	5e-06
WebQA	2,000	3	5e-06

Table 10: Training details on different datasets.

a question to receiving an answer is the same as in the R&R. The additional time is attributed to summarizing user feedback with the LLM and training the retriever , both of which can be performed in parallel with retrieval, thus not adding extra time.

A.3.3 Additional Examples

Fig. 6 presents two examples of responses provided by LUF.

Fig. 6 (a) illustrates how user feedback can assist the LLM in correcting outdated information. For recent events, the LLM's knowledge may not be promptly updated; however, LUF can prevent giving outdated and incorrect answers by leveraging user feedback.

In Fig. 6 (b), the user's question has multiple correct answers, but the answer directly provided by the LLM was not the one the user desired. Through the first conversation, LUF learned the user's preferences and then provided the answer that met the user's expectations.

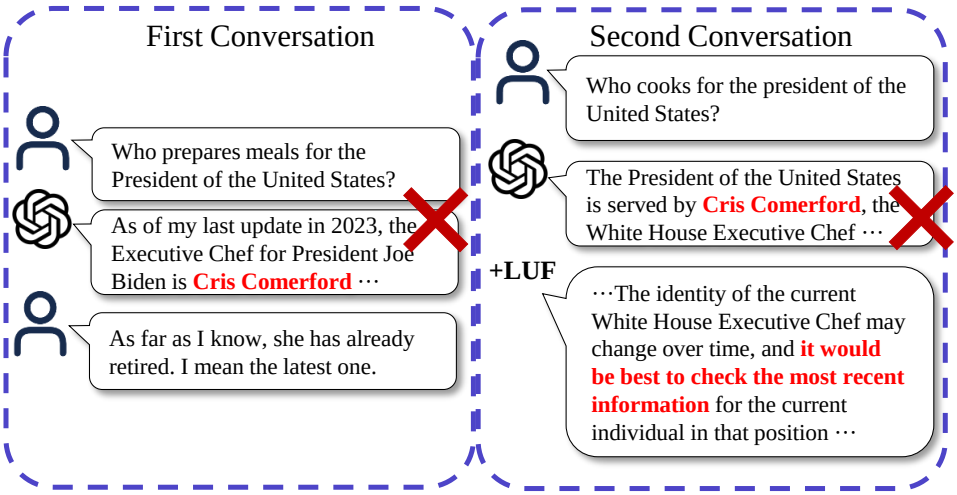
B Implementation Details

B.0.1 Dataset

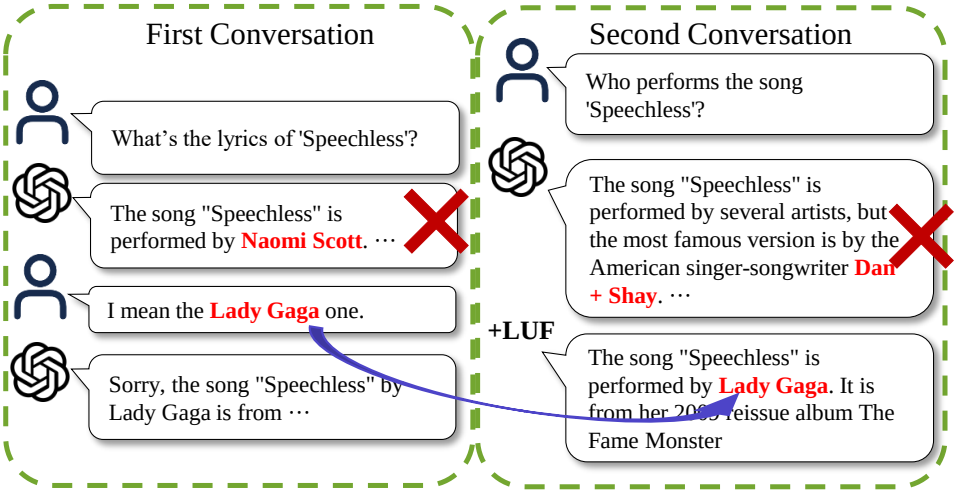
Table 9 shows the number of questions used for testing and the number of documents in the database across different datasets. For the four English datasets, we used the splited Wikipedia from previous studies (Karpukhin et al., 2020) as the document database, while for the Chinese dataset WebQA, we used the evidence provided within the dataset. Since Kimi was unable to answer a few questions, the number of questions tested by Kimi is smaller than that tested by GPT-4o.

B.0.2 Training

Table 10 presents the training details of different datasets. Updating the retriever with more than 2,000 questions each time results in more significant improvements. The similarity threshold λ for retrieving personal user information is set to 0.5.



(a)



(b)

Figure 6: Examples of other LUF reponses.