# Leveraging LLM and User Feedback to Improve 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Retrieval-Augmented Generation (RAG) attempts to mitigate the issue of outdated knowledge and hallucinations in large language models (LLMs) by retrieving real-time information for LLMs.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. Additionally, users have varying expectations for replies. However, current RAG approaches generally overlook the variations in preferred answer domains across different users. 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 extract, identify, and leverage LLM and user feedback from classic RAG process, LUF can adjust to variations in questions and user preferences through updates to the retriever and document database 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).

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 significantly reduce retriever performance, even with the same document database. By splitting the questions and answers from different datasets into word sets and calculating the TF-IDF similarity between them, Table 1 shows that the retriever's accuracy drops more

| Trained on | R@5 of Retriever/TF-IDF Similarity Tested on | | |
|---|---|---|---|
| | NQ | TQA | SQuAD |
| NQ | **56.43/100** | 51.91/42.76 | 55.68/64.64 |
| TQA | 60.55/42.76 | **66.96/100** | 61.67/51.83 |
| SQuAD | 58.39/64.64 | 50.05/51.83 | **64.23/100** |

Table 1: Accuracy of retriever trained and tested on different datasets and TF-IDF similarity of three datasets.

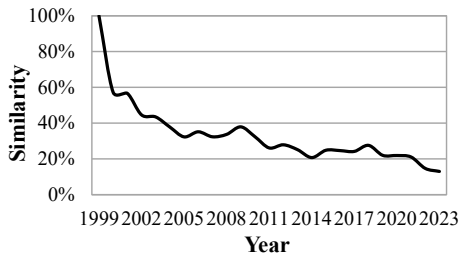



Figure 1: Similarity of most searched questions on Google across years.

when answering questions less similar to those in the training set. In reality, as Fig. 1 shows, we analyzed the top 100 most searched keywords on Google each year and found that people's search interests change rapidly over time, indicating that RAG solutions may perform poorly. 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 asking about courses that would enhance one's skills, a computer science student would benefit from an answer of studying "Computer Systems", yet a biology student would find "General Biology" more appropriate.

Our insight is that feedback from both the LLM and the user in the existing RAG process can help address the performance decline caused by domain shifts in questions and answers. On one hand, reranking results from the LLM better align with the generation goal, as LLMs—despite lacking direct knowledge of the answer—can semantically analyze and identify which documents are more beneficial for generation. Documents retrieved by the retriever but filtered out by the LLM can serve as hard negative samples. Training with these samples can rapidly improve the retriever's accuracy in new question domains (Robinson et al., 2021). On the other hand, during interactions between the LLM and the user, the user naturally provides meaningful feedback regarding both personal preferences and general facts, assisting the LLM in understanding user preferences and learning new knowledge. By incorporating this information into the LLM's "memory", LLM can generate future responses better aligned with user expectations, thereby adapting to shifts in the answer domain.

With this in mind, 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. First, LUF leverages reranking feedback from the LLM to guide retriever updates, improving the retriever's accuracy for unseen questions and enabling it to find more relevant evidence for the LLM. Second, LUF identifies whether user-LLM interactions yield meaningful information. This information is processed by a specifically designed user feedback module, which classifies and integrates it into different document databases. User preferences and feedback from the conversations are materialized as documents, enabling the LLM to provide responses in future interactions that align with user expectations. Notably, LUF does not require explicit annotations from the user but relies entirely on implicit feedback within the existing process, allowing it to quickly adapt to rapidly evolving questions and improve the accuracy of LLM outputs.

LUF is compatible with any learning-based retriever and can be combined with other strategies to enhance RAG performance. 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 or competitive results compared to other LLM-enhanced baselines. In summary, the contributions of this paper are:

- We propose LUF, an RAG framework that self-updates based on feedback, adapts the retriever and LLM output to question and answer domain shifts occurs in real scenarios.
- We designed a module to analyze user feedback provided to help the LLM learn user preferences. To the best of our knowledge, we are the first to integrate user feedback into RAG.
- We conducted experiments using two language models across two tasks to validate the effectiveness of our approach. Additionally, we explored how LLM and user feedback enable RAG to acquire new knowledge and align with user preferences.

## 2 Method

In this section, we introduce the framework and technical details of our method.

### 2.1 Overview of the Proposed Method

The overall architecture of LUF is illustrated in Fig. 2. Compared to other methods where all users share a single document database, we divide documents into two types of databases: a shared database for all users and a personalized database storing information for individual users. Given a user's question $q$, LUF follows a widely adopted workflow to generate responses: it retrieves relevant documents from both databases, reranks them to filter out irrelevant ones, and passes the results to the LLM for response generation. During this

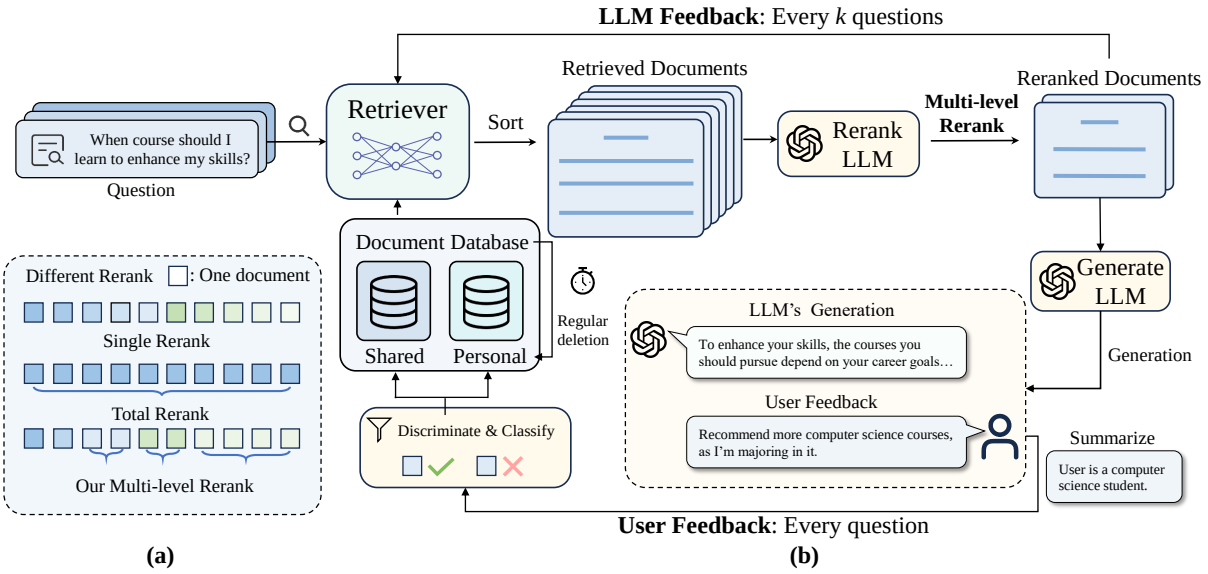



Figure 2: Framework of LUF, including the widely used RAG workflow and processors for LLM and user feedback.

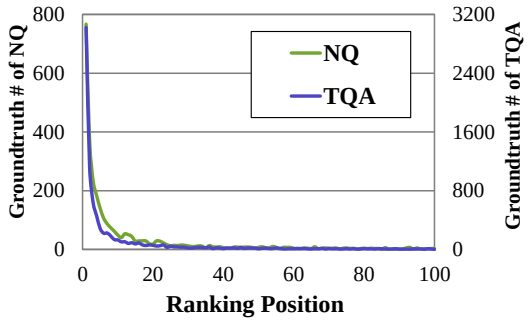



Figure 3: Retrieval results on Natural Questions and Trivia QA, as the rank position increases, the number of relevant documents decreases rapidly.

process, ranking feedback collected from the LLM and textual feedback from users are used to update LUF, aligning RAG with the question and answer domain. The entire process does not require explicit annotations and can be integrated with any LLM or learning-based retriever as plug-in.

## 2.2 Updates Based on LLM Feedback

The accuracy of the retriever tends to decline when faced with unseen questions, whereas the LLM can still assess the relevance between the question and the document based on semantics, even without knowing the exact answer. LLM reranking exploits this characteristic by categorizing the retrieved documents into relevant ones $D^+$ and irrelevant ones $D^-$. The relevant documents filtered by the LLM not only provide the answer but are also very likely to contain relevant information that might help the LLM generate more diverse responses.

LUF advances by utilizing the reranked results provided by LLM as pseudo ground truth to update the retriever during testing. To quickly improve the retriever's accuracy on new questions, we update the model every $k$ questions. Specifically, for every $k$ questions $\{q\}$ posed by users, we conduct offline training of the retriever based on the feedback from the LLM by the following loss function:

$$L(q_i, D_i^+, D_i^-) = -\log \frac{\sum_{d\in D_i^+} e^{\text{sim}(q_i,d)}}{\sum_{d\in D_i^+} e^{\text{sim}(q_i,d)} + \sum_{d\in D_i^-} e^{\text{sim}(q_i,d)}}, \quad (1)$$

where $\text{sim}(\cdot,\cdot)$ measures the similarity between the question and the document given by the retriever.

We emphasize that the ranking feedback used to update the retriever comes from the widely adopted LLM reranking process. LUF does not introduce any additional steps during retrieval and does not delay the time for users to receive a response.

## 2.3 Updates Based on User Feedback

In real scenarios, the feedback provided by users is more complex and may even be entirely incorrect. Therefore, we designed a discriminate-and-classify module to effectively utilize user-provided information. First, information extracted from the dialogue is categorized into two types: common knowledge and user-specific information. Common knowledge is appropriate for all users, while user-specific information contains only individual preferences.

These two types of feedback are stored in separate databases: one shared database for common knowledge, accessible to all users, and a personal database for each user to store individual information, catering to different user preferences for answers.

However, user-provided factual feedback is not always reliable, and incorporating incorrect feedback into the database can harm the generation of LLMs. Additionally, user feedback can sometimes be highly ambiguous, making it difficult to discern its correctness. We propose that instead of directly determining the correctness of user feedback, comparing user feedback with existing knowledge in the document database and LLM is a more reliable approach. Specifically, we designed the following process: first retrieve existing knowledge: relevant documents from the database and LLM's answer without retrieval. Then, we use them as a reference to evaluate the correctness of user feedback. User feedback is added to the database only if the LLM determines it to be correct based on existing knowledge. Since the knowledge in the LLM and the document database provides a foundational reference, this process enables more reliable verification of user feedback. For feedback about user-specific information, we always assume it to be correct and add it to the user's personal database. During response generation, all user-specific information with similarity above a threshold $\lambda$ is included as auxiliary evidence to ensure the LLM sufficiently understands the user's preferences.

LUF allows modifications and deletions in the document database based on user feedback. If the database contains incorrect or outdated information (e.g., Obama being the current U.S. president), LUF will update it accordingly. We also set an expiration period for time-sensitive information (e.g., the user is traveling somewhere this week and might need related recommendations), and remove it after the expiration period to ensure the timeliness of the documents.

### 2.4 Multi-level Reranking

As shown in Fig. 2, 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.

| | NQ | TQA | SQuAD |
|---|---|---|---|
| Kimi | 37.34 | 60.16 | 18.93 |
| GPT-4o | 59.42 | 67.33 | 27.66 |

Table 2: Accuracy without retrieval of GPT-4o and Kimi.

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. 3. Therefore, we propose a multi-level reranking strategy, employing different reranking approaches for different positions, as shown in Fig. 2 (a). 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.

## 3 Experiment

### 3.1 Implementation Details

**Retriever.** We used the pre-trained Contriever (Izacard et al., 2022a) as the retrieval model, which was trained on Wikipedia and CCNet and then tested on the test sets of all five datasets.

**Base LLMs.** We employed two language models: Kimi (Qin et al., 2024), primarily trained on Chinese corpora, and GPT-4o (OpenAI et al., 2024), primarily trained on English corpora. Table 2 shows the accuracy of both models' answers without any retrieval, Kimi's accuracy was significantly lower than that of GPT-4o. We conducted our experiments based on these two models with different knowledge bases, in order to demonstrate the generality of LUF.

**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. The LLM is expected to generate responses that cover all user requirements. Inspired by previous work (Bai et al., 2024), we utilize another LLM to evaluate whether the generated responses satisfy the user intent slots annotated in the datasets.

**Our method.** To simulate domain shift in questions, the retriever used in LUF 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** (Wu 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

Table 3 shows the retrieval accuracy of different methods before and after LLM reranking. Across all baselines, the performance of GPT-4o surpassed that of Kimi, so we will only show the results achieved with GPT-4o. It is evident that LUF consistently achieved superior or comparable performance on all datasets, both before and after reranking. Notably, there was a significant improvement in R@5, which is particularly advantageous because it matches the standard procedure in RAG, where just a select number of very relevant papers are often presented as evidence to LLM.

In general, LUF exhibited more substantial enhancements on larger datasets due to the fact that smaller datasets, such as Web Questions, offer less feedback information from the LLM. In the case of Web Questions, the retriever was only updated once out of every 1000 questions. However, this update still led to an improvement.

LUF also demonstrated consistent improvements across different LLMs. Despite Kimi's ability to answer less than 20% of the questions correctly on SQuAD, the accuracy increase shown in Table 3 implies that when Kimi is used, it can still effectively assess the relevance between questions and documents and provide valuable feedback to the retriever through LUF. This suggests that LUF primarily use the LLM for analyzing semantic relevance, rather than relying on the LLM's knowledge for updates, hence showing LUF's generality. Compared to Kimi, GPT-4o has more knowledge and higher reranking accuracy, leading to greater improvements for the retriever.

### 3.4 Improvement of LLMs' Response in Question Answering

To evaluate how LUF affects LLMs' outputs, we evaluated the responses generated by LLMs on the NQ, TQA, and SQuAD datasets. Specifically, we simulated real-world scenarios where users provide meaningful feedback. Based on the questions in each dataset, we used GPT-4o to generate 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 varying types of user feedback. All generated user feedback was processed by LUF. Then, we created two semantically

| Method | LLM | Natural Questions | | Trivia QA | | SQuaD | | Web Questions | | WebQA | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | R@5 | R@20 | R@5 | R@20 | R@5 | R@20 | R@5 | R@20 | R@5 | R@20 |
| R&R | Kimi | 58.92 | 70.78 | 65.88 | 69.53 | 57.53 | 66.11 | 50.94 | 59.50 | 20.04 | 24.14 |
| R&R | GPT-4o | 59.86 | 71.00 | 67.05 | 69.84 | 58.28 | 66.26 | 51.67 | 59.65 | 20.57 | 24.37 |
| Query Rewrite | GPT-4o | 61.00 | 70.55 | 67.75 | 70.52 | 59.73 | 67.31 | 52.07 | 60.29 | 21.30 | 26.36 |
| Query2Doc | GPT-4o | 59.81 | 71.61 | 67.68 | 70.68 | 60.33 | 67.78 | 52.21 | 60.24 | 22.19 | 26.92 |
| RaFe | GPT-4o | 61.66 | 72.49 | 69.12 | 70.33 | 59.67 | 67.11 | 51.72 | 58.89 | 21.73 | 26.39 |
| **LUF** | Kimi | 62.22 | 75.29 | 68.15 | 71.64 | 62.82 | **70.98** | 52.21 | **60.33** | 25.26 | 30.22 |
| **LUF** | GPT-4o | **64.27** | **76.34** | **69.35** | **71.91** | **63.23** | 70.91 | **52.41** | 59.99 | **26.65** | **31.55** |

Table 3: Retrieval results before and after reranking. The highest results are bolded.

| Method | Natural Questions | | | | TriviaQA | | | | SQuAD | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Original | | Re-Phrased | | Original | | Re-Phrased | | Original | | Re-Phrased | |
| | QA | R@5 | QA | R@5 | QA | R@5 | QA | R@5 | QA | R@5 | QA | R@5 |
| Kimi+ | 37.34 | - | 37.29 | - | 60.16 | - | 57.83 | - | 18.93 | - | 17.47 | - |
| R&R | 48.37 | 58.92 | 44.60 | 57.81 | 61.63 | 65.88 | 58.01 | 59.13 | 36.40 | 57.53 | 32.03 | 52.70 |
| Query2Doc | 49.34 | 61.61 | 45.15 | 59.17 | 61.81 | 66.37 | 58.37 | 60.24 | 35.84 | 58.19 | 31.50 | 54.76 |
| Query Rewrite | 48.39 | 58.86 | 44.65 | 57.34 | 61.76 | 66.23 | 58.54 | 60.39 | 36.13 | 59.35 | 32.72 | 56.16 |
| RaFe | 49.22 | 60.39 | 45.10 | 59.42 | 63.01 | 67.69 | 59.19 | 60.95 | 36.31 | 58.12 | 31.69 | 52.11 |
| **LUF w/o UF** | **50.50** | **62.22** | **46.93** | **60.53** | **63.58** | **68.15** | **59.41** | **62.10** | **37.85** | **62.82** | **33.47** | **58.27** |
| **LUF w/ UF** | - | - | 53.27 | 66.04 | - | - | 64.63 | 65.75 | - | - | 42.33 | 64.53 |
| GPT-4o + | 59.42 | - | 57.42 | - | 67.33 | - | 65.35 | - | 27.66 | - | 24.75 | - |
| R&R | 63.38 | 59.86 | 59.42 | 58.84 | 64.27 | 67.05 | 68.85 | 59.86 | 34.95 | 58.28 | 35.67 | 53.02 |
| Query2Doc | 64.46 | 61.00 | 58.75 | 59.45 | 63.89 | 67.75 | 68.31 | 60.91 | 37.28 | 59.73 | 37.02 | 55.51 |
| Query Rewrite | 64.18 | 59.81 | 59.47 | 58.64 | 64.56 | 67.68 | 69.42 | 60.97 | 37.09 | 60.33 | 37.52 | 56.09 |
| RaFe | 64.57 | 61.66 | 59.92 | 60.28 | 68.03 | 69.12 | 70.21 | 60.79 | 37.36 | 59.67 | 35.94 | 54.09 |
| **LUF w/o UF** | **67.31** | **64.27** | **62.66** | **63.80** | **68.32** | **69.35** | **71.61** | **61.62** | **40.04** | **63.23** | **38.74** | **56.98** |
| **LUF w/ UF** | - | - | 68.98 | 68.37 | - | - | 73.19 | 65.78 | - | - | 43.82 | 64.70 |

Table 4: 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 5: Accuracy of judging the correctness of user feedback. "Direct" means directly asking the LLM to make the judgment.

equivalent questions for each original question in the datasets to construct re-phrased QA datasets, aiming to verify whether LUF accurately extracted information from the user feedback. Details of the feedback and question simulation are provided in Appendix A.1.

Table 4 presents the performance of all methods on both the original and re-phrased datasets, with LUF delivering the most accurate responses across all datasets. By incorporating user feedback, LUF significantly improved retrieval and answering accuracy on the re-phrased datasets. This improvement is attributed to LUF's ability to extract correct information from user feedback, and add it to the document database, thereby enhancing the accuracy of responses to future questions from other users. Notably, even without incorporating user feedback, LUF 's superior retrieval accuracy supplied LLMs with more precise evidence, resulting in better responses on both the original and re-phrased datasets compared to the baselines.

Table 5 compares the accuracy of letting LLM directly judge correctness of user feedback and using LUF 's discriminate-and-classify modules. On all

| Method | MultiWOZ 2.2 | | SGD | |
|---|---|---|---|---|
| | JGA | AGA | JGA | AGA |
| R&R | 61.39 | 93.30 | 80.23 | 91.67 |
| Query2Doc | 61.73 | 92.71 | 80.67 | 92.19 |
| Query Rewrite | 61.85 | 93.52 | 80.85 | 91.63 |
| RaFe | 61.45 | 93.90 | 79.95 | 92.29 |
| LUF | **62.83** | **94.67** | **81.51** | **92.55** |

Table 6: The accuracy of the LLM responses in multi-turn dialogues.

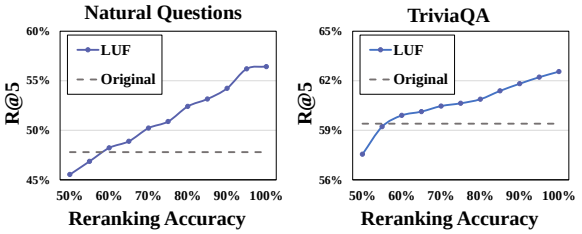



Figure 4: R@5 of LUF with different reranking accuracy.

three datasets, LUF correctly judged over 97% of the feedback, demonstrating its robustness in leveraging useful user feedback while preventing incorrect information from influencing the databases and LLM outputs.

### 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. We let different methods directly generate responses and employed another LLM to evaluate whether these responses met the annotated user intents. From Table 6, we observed that while other methods provided minimal improvements, LUF-generated responses better satisfied user requirements. This was attributed to LUF's ability to summarize user feedback, store user preferences in the RAG document database, and use these preferences as auxiliary evidence during response generation. LUF enabled the LLM to better understand and retain user preferences.

The test samples used included complete dialogues, meaning the LLM could directly infer user intents from the context. However, the LLM occasionally forgot certain user requests over turns. Experimental results demonstrated that, even within the same conversation, LUF could utilize user feedback to concretize user preferences, thereby assisting the LLM in providing responses aligned with user expectations.

### 3.6 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.

### 3.7 Further Investigations

**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. 4, the accuracy refers judging single relevant documents, as there is no significant difference between different LLMs in judging irrelevant documents.

Fig. 4 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 comes from LLMs' high accuracy in filtering out irrelevant samples, the eliminated hard negative samples help the retriever improve. We tested three widely used LLMs: GPT-4o, Kimi, and DeepSeek-v2 (DeepSeek-AI, 2024), all of which achieved reranking accuracy above 70%, suggesting that LUF can be applied to them.

**How LUF Identifies Feedback**

The results in Table 5 demonstrate that the designed discriminate-and-classify module improves the accuracy of identifying user feedback, ensuring that the document database is not contaminated by erroneous feedback. This section provides two examples of how discriminate-and-classify module works to prevent incorrect feedback from corrupting the whole system's output.

**Compare with existing information**: In the ex-

| 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.

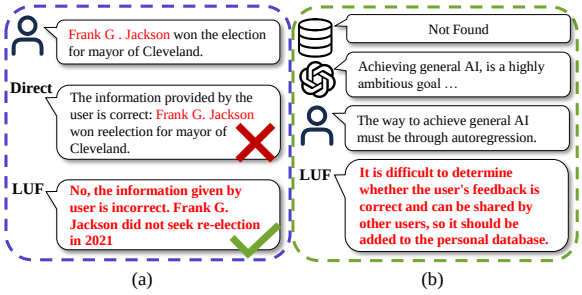



Figure 5: Examples of LUF's responses to different kinds of incorrect feedback.

ample shown in Fig. 5 (a), the evidences from document database and LLM provide a foundational 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. 5 (b). Although the correctness of such feedback cannot be determined, LUF prefers to add it to personal databases to prevent contamination of the whole system and other users.

# 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 two recent studies have also attempted to improve RAG using feedback too: RaFe (Wu 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 task, 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 Qwen1.5-32b-chat. For these models with 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


Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, et al. 2024. Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. *arXiv preprint arXiv:2402.14762*.

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.

DeepSeek-AI. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model.

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.

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.

Ruoyu Qin, Zheming Li, Weiran He, Mingxing Zhang, Yongwei Wu, Weimin Zheng, and Xinran Xu. 2024. Mooncake: A kvcache-centric disaggregated architecture for llm serving.

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.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive learning with hard negative samples.

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.

Zhongkai Wu, Ziyu Wan, Jing Zhang, Jing Liao, and Dong Xu. 2024. Rafe: Generative radiance fields restoration.

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 4, 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-

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 Additional Experiment Results

### A.2.1 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 reask datasets, as shown in Table 8.

After applying different strategies, both retrieval and response accuracy improved. Replacing the single reranking with multi-level reranking that consumes a similar number of tokens, there has been a certain improvement retrieval accuracy. LLM Feedback mainly helps the retriever adapt to the shift in the domain of questions, while user feedback significantly improves the results obtained by the retriever and LLM by adding new information to the database.

### A.2.2 Time Consumption

Table 9 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 (Wu 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 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.2.3 Token Consumption

Table 10 shows the token consumption of different reranking strategies on Web Questions and WebQA. Compared to other methods, Multi-level reranking used the fewest tokens on the both English and Chinese datasets.

### A.2.4 Additional Examples

Fig. 7 presents two examples of responses provided by LUF.

Fig. 7 (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. 7 (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 11 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

| 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 |
| +Multi-level Reranking | 44.60 | 57.81 | 58.01 | 59.13 | 32.03 | 52.70 |
| +LLM Feedback | 47.23 | 61.69 | 60.67 | 60.81 | 34.32 | 57.79 |
| +User Feedback | **66.04** | **71.29** | **72.54** | **72.48** | **58.97** | **77.84** |

Table 8: Ablation study results.

| 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 9: The detailed time required to retrieve 3,000 questions of different 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 10: The number of tokens consumed by different reranking methods on Web Questions and WebQA.

| Dataset | Test Set Size | | Number of Documents |
|---|---|---|---|
| | Kimi | GPT-4o | |
| Natural Questions | 3,602 | 3,610 | 21,015,324 |
| TriviaQA | 11,297 | 11,313 | 21,015,324 |
| SQuAD | 10,554 | 10,570 | 21,015,324 |
| Web Questions | 2,029 | 2,032 | 21,015,324 |
| WebQA | 3,024 | 3,024 | 3,024 |

Table 11: The number of test questions and documents of different dataset.

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 12 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 $\lambda$ for retrieving personal user information is set to 0.5.

| 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 12: Training details on different datasets.

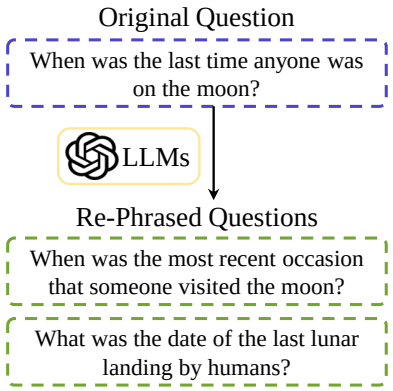
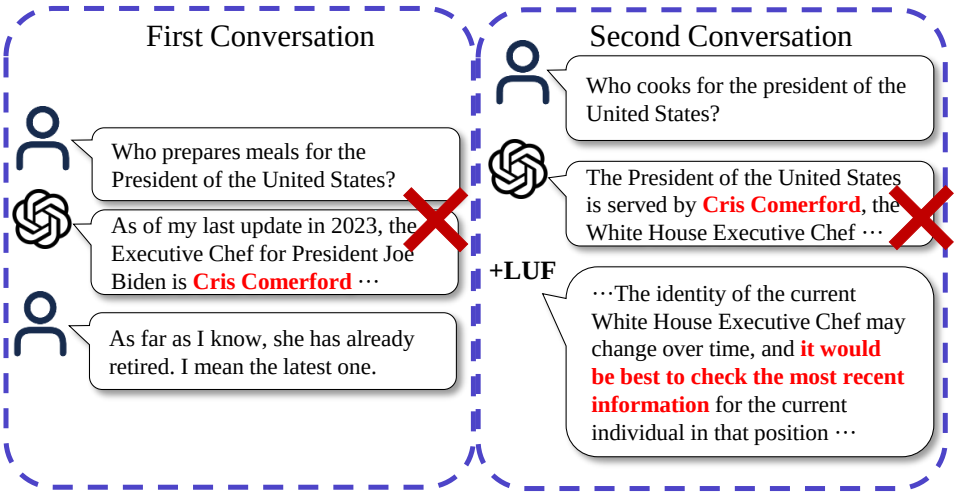


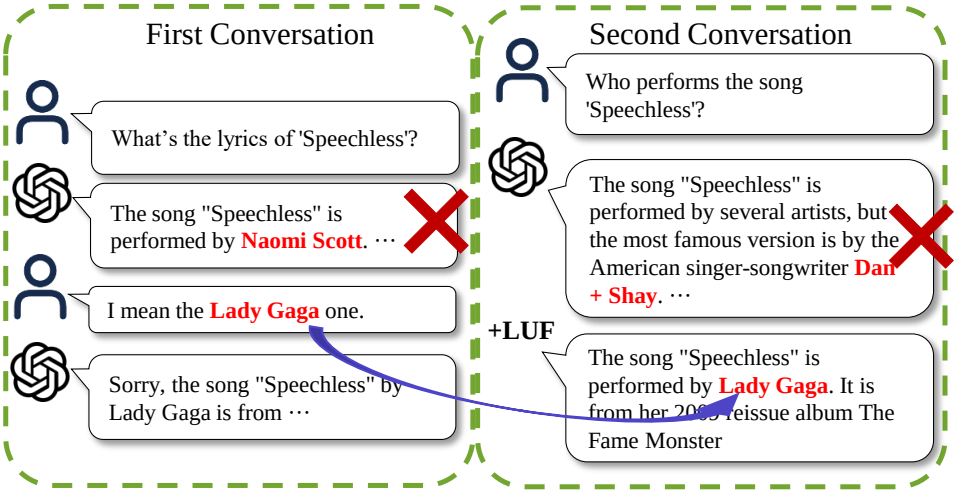

Figure 6: Examples of constructed re-ask dataset.

### B.0.3 Construction of Re-phrased Datasets

Fig. 6 shows an example of how we constructed the re-ask datasets. We used an LLM to rephrase the original question, generating two new questions with the same meaning but different structures, simulating the scenario where different users ask similar questions.




(a)




(b)

Figure 7: Examples of other LUF reponses.