`RAG-EF`: Train-free RAG Enhancement from Expert Feedbacks

Anonymous ACL submission

Abstract

As large language models (LLMs) become increasingly integral to digital interactions, their susceptibility to generating inaccurate or nonsensical content, called *hallucination*, poses significant challenges. Retrieval-Augmented Generation (RAG) has emerged as a promising technique to curb hallucinations by leveraging external databases to inform response generation. However, the RAG framework is not without limitations, often requiring computationally expensive methods like domain-specific retrieval-augmented finetuning. We introduce a novel and efficient enhancement RAG framework, `RAG-EF` (**RAG** with **E**xpert **F**eedback), which incorporates expert-provided feedback composed of problematic Q&A and context pairs. Also, we present a new retrieval strategy that utilizes contexts alongside Q&A pairs to optimize information selection and prevent incorrect responses. To show the effectiveness of `RAG-EF`, we establish three new benchmarks with three datasets, and demonstrate adding relevant feedback into the database greatly improves the performance.

1 Introduction

The growth of data on the internet has spurred the advancement of large language models (LLMs) (Dubey et al., 2024; Achiam et al., 2023; Team et al., 2024). These models are transforming our interaction with digital content, proving invaluable in various domains, such as personalization (Bang et al., 2024), text summarization (Basyal and Sanghvi, 2023), etc. Despite their advanced capabilities, LLMs are prone to producing content that are inaccurate or nonsensical, an issue referred to as *hallucination* (Ji et al., 2023).

To alleviate this problem, Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) has been introduced, which is a technique that improves LLMs by incorporating information from external databases (*e.g.,* Wikipedia) during the inference.

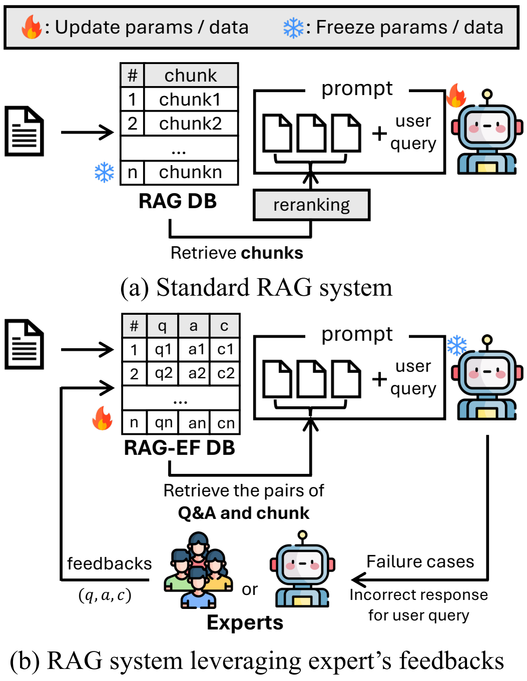

Figure 1: **Overall system of `RAG-EF`.** For enhancing the model for failure cases, standard RAG system trains the LLM with training set augmented with feedbacks for failure cases, yet it is inefficient. `RAG-EF` is the new RAG system for enhancing the feedback (q, a, c) from the experts without model training.

This approach has been effective in reducing hallucinations by providing a factual basis for response generation. However, it is inevitable for a retriever to select the irrelevant (*i.e.,* noisy) context to the user query. Additionally, the prior work (Liu et al., 2024a) found that ordering retrieved contexts can severely affect the RAG performance (*i.e., lost-in-the-middle*). As such, many works (Yan et al., 2024; Yu et al., 2024c; Dong et al., 2024) proposed the methods for reranking retrieved contexts using LLM, but it still has difficulties for specific domain that the LLM struggles with.

To adapt the model for the specific domain, fine-tuning methods have been revealed. RAFT (Zhang et al., 2024) was proposed to make the robust generator in irrelevant contexts for a specific domain. Additionally, the paper (Siriwardhana et al., 2023) was introduced to adapt both a retriever and a generator into the domain-specific document. Although they improve the performance for the specific domain, those kind of methods require to train the model whenever updating the database, and it is impractical framework for maintaining the model.

This paper aims to address the efficient RAG enhancement framework without additional extensive training when the database is updated. Inspired by the paper that argues LLM cannot correct reasoning itself and need external feedback (Huang et al., 2024), we firstly propose the novel RAG framework based on feedback from the experts, as illustrated in Fig. 1. More specifically, when the LLM generates the incorrect answer for the user query, the experts (*e.g.,* human or LLM) generate the feedback to correct the answer for the user query, and this feedback is inserted into the database. In this context, we also introduce a new retrieval strategy that utilizes both the contexts and the Q&A pairs (*i.e.,* feedback) in the database to refine the RAG process. The contributions are three folds:

- We firstly propose **RAG-EF**, a novel efficient RAG enhancement framework by updating the database (*i.e.,* feedback) without training.
- We develop a unique retrieval method considering both context and Q&A that are stored pairly in the database for effective sampling information in our framework.
- We demonstrated that as the relevant feedback set in **RAG-EF** becomes more structured, it boosts performance more significantly compared to the standard RAG, and it also outperforms the baselines using the larger models.

2 RAG-EF: RAG with Expert Feedback

2.1 Problem Formulation of RAG

Typically, RAG utilizes two models; Retriever($\mathcal{R}$) and Generator($\mathcal{G}$). At first, given a document d, we need to split it into several chunks($C = \{c_1, c_2, ..., c_n\}$) due to max token constraint of LLM. Then, retriever $\mathcal{R}$ generates the embeddings for each chunk and the user query, and selects the top-K chunks from C that are most relevant chunks to the given user query q by calculating the similarity. The similarity $\mathbf{s}_i$ between each chunk c_i and user query q as below:

$$\mathbf{s}_i = e_{\mathrm{q}}^{\top} e_{c_i}, \;\; e_{\mathrm{q}} = \mathcal{R}(\mathrm{q}), \;\; e_{c_i} = \mathcal{R}(c_i) \quad (1)$$

Then, both a query and selected chunks are fed into the prompt as the input of the generator $\mathcal{G}$.

2.2 Configuration of RAG-EF Database

Different from conventional RAG, **RAG-EF** database (DB) consists of both Q&A and chunk that contains the answer the question. There are two approaches to collect the Q&A-Chunk pairs.

Feedback from Expert. In some cases, RAG may generate the wrong responses by several reasons. For failure cases, the experts can annotate the answer a and golden chunk c for the user query q, where c is part of the document d but is not necessary to be same as one of the c_i. After constructing the (q, a, c) pair, it is inserted into the database.

Generation from LLM. When the chunk c_i is given, we can ask the LLM to generate the Q&A pairs to validate the understanding of c_i. By doing so, we can collect the set of Q&A and chunk c_i pairs (*i.e.,* $\{(q_j, a_j, c_i) | j \in [1, m_i]\}$, and add them into the database (see Appendix A for details). Generally, we utilize this approach when building the database at the initial stage.

2.3 Retrieval Method of RAG-EF

Different from standard RAG solutions, we need to consider Q&A and chunk pair $p_i = (q_i, a_i, c_i)$ that is stored in the database. To do so, we propose a new retrieving method by reformulating the similarity from Eq. 1.

$$\mathbf{s}_i = (e_{\mathrm{q}}^{\top} e_{q_i})^{\gamma} (e_{\mathrm{q}}^{\top} e_{c_i})^{1-\gamma} \quad (2)$$

where $e_{\mathrm{q}} = \mathcal{R}(\mathrm{q})$, $e_{q_i} = \mathcal{R}(q_i)$, $e_{c_i} = \mathcal{R}(c_i)$ and hyperparameter $\gamma \in [0, 1]$. Given similarities $\mathbf{s}_i$ for all the pairs p_i from Eq. 2, we select top-K pairs.

After sampling the Q&A-Chunk pairs p_i, both a user query q and selected pairs p_i are added into the input of the generator $\mathcal{G}$. Since the chunks can be duplicated, the number of chunks can be less than the number of pairs (See Appendix B for details).

3 Experiment

3.1 RAG-EF Database Benchmark

Database. This work is the first to suggest the use of feedback (*i.e.,* Q&A-Chunk pairs p_i) to enhance the RAG system. Since there is no standard for evaluating the efficacy of **RAG-EF**, we introduce three benchmarks to assess its performance. Based

Method	Model	NQ	TriviaQA	HotpotQA	Average
Standard RAG	Llama-3 8B	28.7	67.1	28.5	42.1
Self-RAG (Asai et al., 2024)	Llama-3 8B	36.4	38.2	29.6	34.7
Auto-RAG (Yu et al., 2024a)	Llama-3 8B	37.9	60.9	44.9	47.9
ChatQA-1.5 (Liu et al., 2024b)	Llama-3 8B	42.4	61.0	44.6	49.3
RankRAG (Yu et al., 2024c)	Llama-3 8B	50.6	82.9	46.7	60.1
Standard RAG	Llama-3 70B	42.7	82.4	43.3	56.1
ChatQA-1.5 (Liu et al., 2024b)	Llama-3 70B	47.0	85.6	54.4	62.3
RankRAG (Yu et al., 2024c)	Llama-3 70B	54.2	86.5	55.4	65.4
RAG-EF (Ours)	Llama-3 8B	**76.6**	**96.5**	**84.5**	**85.9**

Table 1: **Performance comparison of RAG-EF and baseline RAG methods.** We compare **RAG-EF** with various RAG-based approaches using Llama-3 8B and Llama-3 70B models on Natural Question (NQ), TriviaQA, and HopotQA benchmarks. Note that **RAG-EF** utilizes bge-m3 retriever with $\mathcal{S} \cup \mathcal{E} \cup \mathcal{O}$ feedback databases.

$\mathcal{R}$	Database	NQ	TriviaQA	HotpotQA	Average
BM25	$\mathcal{S}$	26.0	75.6	30.1	43.9
	$\mathcal{S} \cup \mathcal{E}$	27.0	77.8	30.8	45.2
	$\mathcal{S} \cup \mathcal{E} \cup \mathcal{O}$	27.1	90.1	31.2	49.5
DPR	$\mathcal{S}$	36.9	74.7	43.8	51.8
	$\mathcal{S} \cup \mathcal{E}$	50.0	81.6	52.7	61.4
	$\mathcal{S} \cup \mathcal{E} \cup \mathcal{O}$	77.9	93.2	81.3	84.1
bge-m3	$\mathcal{S}$	36.2	76.8	44.8	52.6
	$\mathcal{S} \cup \mathcal{E}$	49.8	83.9	53.2	62.3
	$\mathcal{S} \cup \mathcal{E} \cup \mathcal{O}$	76.6	96.5	84.5	85.9

Table 2: **Comparison of RAG-EF performance with incrementally added feedback sets.** We compare the performance of **RAG-EF** across various retrievers by sequentially incorporating feedback sets in ascending order of relevance to the user query q ($\mathcal{S} < \mathcal{E} < \mathcal{O}$)

on Sec. 2.2, these benchmarks include one for generation using LLM ($\mathcal{S}$) and two that incorporate feedback from the expert ($\mathcal{E}$ and $\mathcal{O}$), to demonstrate the performance improvement when feedback is included in the database (See Appendix C).

Synthesis($\mathcal{S}$). It is created with the generative method described in Sec. 2.2, utilizing the Llama-3 8B with each chunk. As it is created without a test set, it is hard to collect the relevant Q&A-Chunk pairs to the user query q within a test set compared to the other benchmarks using the test set.

Expert($\mathcal{E}$). Adhering to the expert feedback method detailed in Sec. 2.2, it is generated by the expert (employing a GPT-4 turbo) with a test set. Initially, the LLM rephrases the user query q, following which it generates answers for the rephrased query using a golden chunk. Consequently, each Q&A-Chunk pair p_i is composed of the rephrased query, the LLM's response, and a golden chunk.

Omniscient($\mathcal{O}$). It also constitutes of the feedbacks from the expert, but it is totally same as a test set. Therefore, it is the most relevant to the test set among three benchmarks, but impractical.

Implementation Details. To show the effectiveness of **RAG-EF** for various retrievers, we adopt three retrievers $\mathcal{R}$ that are BM25 (Robertson et al., 2009), DPR (Karpukhin et al., 2020), and bge-m3 (Chen et al., 2024). All the retrievers select five pairs in the database using Eq. 2 with $\gamma = 0.5$. As a generator $\mathcal{G}$, we utilized Llama-3 8B (Dubey et al., 2024), and we adopted GPT-4 turbo as an expert. Note that we evaluate all experiments using Exact Match (EM) for Natural Language (NQ) and TriviaQA datasets, while the HotpotQA dataset was evaluated using the F1 score (See Appendix D).

3.2 Experimental Results

Effectiveness of RAG-EF. RAG-EF demonstrates notable performance improvements over existing RAG-based methods across all evaluated benchmarks. As shown in Tab. 1, **RAG-EF** achieves an average score of 85.9, surpassing the best baseline, RankRAG, by nearly 26 points with the same model size. Even when RankRAG leverages a larger model, **RAG-EF** still outperforms it by almost 19 points. These results highlight the effectiveness and efficiency of **RAG-EF**, particularly when relevant question-and-answer pairs for the user query q existed in the database. In such cases, **RAG-EF** consistently demonstrates superior RAG capabilities compared to models with significantly larger parameters.

Various Retrievers. Tab. 2 illustrates the impact of different retrievers and feedback set configurations on **RAG-EF**'s performance. The results reveal a clear trend: as the feedback set becomes more relevant ($\mathcal{S} \rightarrow \mathcal{S} \cup \mathcal{E} \rightarrow \mathcal{S} \cup \mathcal{E} \cup \mathcal{O}$), the overall performance improves significantly. Across all retrievers, incorporating $\mathcal{E}$ and $\mathcal{O}$ in addition to $\mathcal{S}$ consistently enhances both retrieval and generation quality. The bge-m3 retriever, for example, attains an average score of 85.9 when all three sets are combined, compared to only 52.6 when using $\mathcal{S}$.

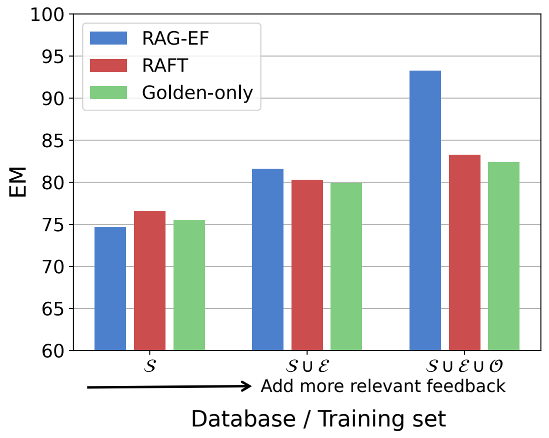

Figure 2: **Performance comparison between RAG-EF (ours) and train-based RAG (*i.e.,* RAFT) in TriviaQA.** Despite of necessities of extra computational resources, RAFT fails to meet RAG-EF even with more feedback sets (Details in Appendix D.3).

3.3 Further Analyses

We conducted various additional analyses to further demonstrate the superiority of **RAG-EF**. For the sake of space, we present only two key analyses, which are comparison of train-based RAG and component-wise analysis of **RAG-EF**, while the rest can be found in Appendix E.

Comparison of train-based RAG. As shown in Fig. 2, **RAG-EF** significantly outperforms train-based RAG methods such as RAFT and Golden-Only when the database contains more relevant feedback pairs p_i. The retrieval mechanism proposed from Eq. 2 plays a crucial role in this performance boost by efficiently selecting relevant feedback. Furthermore, the retrieved question-and-answer pairs are directly incorporated into the generator's input prompt, creating an effect similar to in-context learning. As a result, **RAG-EF** leads to a noticeable performance gap as we move from $\mathcal{S}$ to $\mathcal{S} \cup \mathcal{E} \cup \mathcal{O}$, highlighting the unique advantage when the database closely matches the test set.

Component-wise analysis. Tab. 3 presents an ablation study where we progressively replace components of Standard RAG with our proposed methods to measure their impact on performance. The baseline achieves an average score of 45.0 across three datasets. First, applying our revised similarity metric from Eq. 2 improves the performance with 12 points, demonstrating the effectiveness of balancing query-to-question and query-to-chunk similarities. Moreover, incorporating selected Q&A pairs into the input prompt boosts performance to 61.4, highlighting the benefit of providing richer contextual knowledge for the generator.

Method	NQ	TriviaQA	HotpotQA	Average
Standard RAG	28.7	67.1	39.1	45.0
+ Retr. Eq. 2	45.6	77.8	47.6	57.0
+ Add Q&A (ours)	50.0	81.6	52.7	61.4

Table 3: **Ablation Study of RAG-EF.** Our proposed approach, which integrates the revised retrieval method from Eq. 2 and the addition of Q&A pairs into the input prompt, achieves the highest average performance.

4 Related Works

Train-based RAG. Several papers (Lin et al., 2024; Wang et al., 2024) have been proposed in terms of improving RAG performance with training. They focused on instruction tuning pretrained LLM to be familiar with RAG tasks, preparing the training set with several domains for generability. On the contrary, RAFT (Zhang et al., 2024) focused on finetuning the generator(*i.e.,* LLM) for adapting the specific domain, and RAG-end2end (Siriwardhana et al., 2023) introduced the joint training method of the retriever and the generator. Recently, Chain-of-Note (Yu et al., 2024b) proposed the reasoning method by training the generator to generate both summaries for each context and an answer inspired by Chain-of-Thought (Wei et al., 2022), and showed the performance improvement.

Train-free RAG. While RAG mostly solves the hallucination of pretrained LLM, RAG has the severe problem called *lost-in-the-middle* (Liu et al., 2024a), which LLM generates the wrong answers with high probability when the golden context is located in the middle of the prompt. To solve this problem, *Reranking* concept was proposed. CRAG (Yan et al., 2024) proposed a *retrieval evaluator*, which decides whether to put contexts selected from the retriever into the LLM's prompt or not. Rank-RAG (Yu et al., 2024c) proposed unified LLM that can conduct ranking the contexts and generating the answer, and G-RAG (Dong et al., 2024) proposed graph-based reranking system. Similarly, Self-RAG (Asai et al., 2024) proposed *critique* to determine whether each context is dropped or not.

5 Conclusions

We introduced **RAG-EF**, a novel RAG enhancement system by integrating feedback mechanisms with an updated database. Our unique retrieval method utilizes context and Q&A to effectively select pairs in the database. The efficacy of **RAG-EF** was validated, making it the optimal RAG enhancement system for handling updated databases.

6 Limitations

While **RAG-EF** achieves strong performance, there are some minor, yet addressable, aspects to consider. The input length of the generator $\mathcal{G}$ increases in our method compared to conventional RAG systems because the **RAG-EF** database incorporates both Q&A pairs and contexts; however, most Q&A pairs are *under 100 tokens*, much shorter than the retrieved contexts, so the overall increase remains manageable. Moreover, our system additionally requires more space to store the feedback database, but the space size is reasonable. For instance, with the DPR retriever, each feedback entry is represented as a 1024-dimensional embedding in float16 format, requiring roughly 2KB per entry. In other words, about *500,000 feedbacks* can be stored in one giga byte. Notably, a larger database significantly boosts the likelihood of retrieving highly relevant Q&A chunk pairs and enhances performance. To further optimize storage usage for resource-constrained settings without compromising these benefits, future work will explore *efficient database management strategies*, such as cache management or scheduling to replace infrequently accessed feedback with new entries, as well as mechanisms to detect and refresh outdated contexts.

7 Potential Risks

Retrieval-Augmented Generation (RAG) models pose potential privacy risks when the document incorporates the sensitive information such as personal information. **RAG-EF** focuses on enhancement of RAG system for failure cases, and it is not directly related to the negative ethical and societal impacts. Nonetheless, when **RAG-EF** is applied to the application service, it needs to consider carefully whether external database (*i.e.,* document) contains the private information or not.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *ArXiv:2303.08774*.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *Proceedings on International Conference on Learning Representations*.

Jihwan Bang, Juntae Lee, Kyuhong Shim, Seunghan Yang, and Simyung Chang. 2024. Crayon: Customized on-device llm via instant adapter blending and edge-server hybrid inference. In *Proceedings on Association for Computational Linguistics*, pages 3720–3731.

Lochan Basyal and Mihir Sanghvi. 2023. Text summarization using large language models: a comparative study of mpt-7b-instruct, falcon-7b-instruct, and openai chat-gpt models. *ArXiv:2310.10449*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *ArXiv:2402.03216*.

Jialin Dong, Bahare Fatemi, Bryan Perozzi, Lin F Yang, and Anton Tsitsulin. 2024. Don't forget to connect! improving rag with graph-based reranking. *ArXiv:2405.18414*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *ArXiv:2407.21783*.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. In *Proceedings on International Conference on Learning Representations*.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *Proceedings on International Conference on Learning Representations*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings on Association for Computational Linguistics*, pages 1601–1611.

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 on Empirical Methods in Natural Language Processing*, pages 6769–6781.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the*

Association for Computational Linguistics, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings on Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, et al. 2024. Ra-dit: Retrieval-augmented dual instruction tuning. In *Proceedings on International Conference on Learning Representations*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 11:157–173.

Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Mohammad Shoeybi, and Bryan Catanzaro. 2024b. Chatqa: Building gpt-4 level conversational qa models. *ArXiv:2401.10225*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

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.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *ArXiv:2408.00118*.

Boxin Wang, Wei Ping, Lawrence McAfee, Peng Xu, Bo Li, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Instructretro: Instruction tuning post retrieval-augmented pretraining. In *Proceedings on International Conference on Machine Learning*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings on Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *ArXiv:2401.15884*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Tian Yu, Shaolei Zhang, and Yang Feng. 2024a. Auto-rag: Autonomous retrieval-augmented generation for large language models. *ArXiv:2411.19443*.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2024b. Chain-of-note: Enhancing robustness in retrieval-augmented language models. In *Proceedings on Empirical Methods in Natural Language Processing*.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024c. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *ArXiv:2407.02485*.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024. Raft: Adapting language model to domain specific rag. *ArXiv:2403.10131*.

-Supplementary Material-

RAG-EF: Train-free RAG Enhancement from Expert Feedbacks

A Q&A Generation from LLM

To build the synthesis dataset $\mathcal{S}$, we ask LLM (*i.e.,* Llama-3 8B) for each chunk from the document by utilizing following prompt template. After getting responses from Llama-3 8B, we parse them into the question-and-answer pairs. In other words, we can get the Q&A-Chunk pairs p_i for synthesis set $\mathcal{S}$ by conducting those process.

Prompt template for Q&A Generation from LLM

```
please generate question and answer pairs to check
the understandability of following context.
Format should be Q: question A: answer.
{CONTEXT}
```

B Prompt Template for RAG

Prompt engineering with retrieved chunks or Q&As is the another important factor to improve the performance. For fair comparison between **RAG-EF** and conventional RAG, we tried to unify the prompt template as shown below. Especially, **RAG-EF** retrieves not only chunks but also question and answer pairs, they added Q&As into the prompts before the contexts. Since it can be possible to duplicate the chunks, the number of retrieved contexts, denoted as N, can be five or fewer. Aside from this difference, the prompt templates employed by **RAG-EF** and the conventional RAG remain largely consistent.

Prompt template for RAG-EF

```
Question: {QUESTION #1}
Answer: {ANSWER #1}
Question: {QUESTION #2}
Answer: {ANSWER #2}
...
Question: {QUESTION #5}
Answer: {ANSWER #5}
Context1: {CONTEXT #1}
Context2: {CONTEXT #2}
...
ContextN: {CONTEXT #N}
Please answer the below question based on given
above question and answer pairs and contexts.
Note that you should generate the response only
for answering the question within a few words.
Do not contain extra comments.
Question: {TARGET QUESTION}
```

Prompt template for conventional RAG

```
Context1: {CONTEXT #1}
Context2: {CONTEXT #2}
...
Context5: {CONTEXT #5}
Please answer the below question based on given
above contexts. Note that you should generate
the response only for answering the question
within a few words. Do not contain extra comments.
Question: {TARGET QUESTION}
```

C RAG-EF Database Benchmark

C.1 Examples in Three Types of Database

Tab. 4 shows the examples of three database benchmarks $\mathcal{S}$, $\mathcal{E}$, and $\mathcal{O}$ that are most relevant examples to the test set example using the DPR retriever. As described in Sec. 3.1, since $\mathcal{S}$ is built by LLaMA-3 8B with the chunk c_i, it is hard to generate the Q&A that are exactly same as the test set example. Nonetheless, the example from $\mathcal{S}$ shows the correlation of test set example, and it is sufficient to generate the correct answer for the question of the test set by referring the Q&A of $\mathcal{S}$. In $\mathcal{E}$, the only difference from the test set example is the question, and you can observe that they have the same meaning. Sometimes, it can get the different (*i.e.,* incorrect) answers due to answer generation from LLM (See the LLM's performance in below section). Lastly, the example of $\mathcal{O}$ is same as the test set example.

C.2 Performance of LLM when Building $\mathcal{E}$

To build the dataset $\mathcal{E}$ that is relevant and realistic, we make GPT paraphrase the question and generate the answer given rewritten question and whole document to regard the GPT as a human. As such, Tab. 5 shows the results of GPT for various datasets. Due to constrained performance of GPT, Tab. 1 shows that there is a gap between $\mathcal{S} \cup \mathcal{E}$ and $\mathcal{S} \cup \mathcal{O}$. We believe that the results of $\mathcal{S} \cup \mathcal{E}$ can significantly be improved when the performance of GPT increases (The humans can have more accurate results than GPT).

D Implementation Details

D.1 Datasets

TriviaQA (Joshi et al., 2017). It offers a complex question answering dataset featuring 950,000

Set	Question	Answer	Chunk
test	Who was known by his stage name Aladin and helped organizations improve their performance as a consultant?	Eenasul Fateh	Eenasul Fateh (born 3 April 1959), also known by his stage name Aladin, is a Bangladeshi-British cultural practitioner, magician, live artist and former international management consultant.
$\mathcal{S}$	Who is Eenasul Fateh also known as?	Aladin	Eenasul Fateh (born 3 April 1959), also known by his stage name Aladin, is a Bangladeshi-British cultural practitioner, magician, live artist and former international management consultant.
$\mathcal{E}$	Who was recognized under the stage name Aladin and provided consultancy services to enhance organizational performance?	Eanasul Fateh	Eenasul Fateh (born 3 April 1959), also known by his stage name Aladin, is a Bangladeshi-British cultural practitioner, magician, live artist and former international management consultant.
$\mathcal{O}$	Who was known by his stage name Aladin and helped organizations improve their performance as a consultant?	Eenasul Fateh	Eenasul Fateh (born 3 April 1959), also known by his stage name Aladin, is a Bangladeshi-British cultural practitioner, magician, live artist and former international management consultant.

Table 4: **Examples of three database benchmarks that are the most relevant to ground truth in HotpotQA.**

Dataset	EM	F1
Natural Question	54.64	72.98
TriviaQA	72.05	82.57
HotpotQA	35.10	46.18

Table 5: **Expert (*i.e.,* GPT-4 turbo) performance in various datasets given whole document.** Since some examples do not have the key passages that contain the answers, the expert has especially poor performance in HotpotQA.

Parameters	Value
Rank	8
Alpha	16
Train on Inputs	True
LoRA Modules	Q, K, V, O
Dropout	0.05
Batch Size	128
Loss Function	Cross Entropy Loss
Learning Rate	3e-4
Scheduler	Cosine Annealing
Optimizer	AdamW
Epochs	1

Table 6: **Training hyperparameters.**

pairs of questions and answers sourced from over 662,000 documents on Wikipedia and the internet. Unlike simpler QA benchmarks like SQuAD, TriviaQA presents a tougher challenge because answers often require more than just predicting a text span from lengthy contexts. The dataset includes both subsets that are verified by humans and those created by machines, and we only utilized verified datasets in this experiment.

Natural Questions (Kwiatkowski et al., 2019). It is designed for training question answering systems, featuring over 300,000 training, nearly 8,000 development, and approximately 8,000 test examples, each pairing a Google search query with a relevant Wikipedia article. These articles include marked sections that provide a detailed response to the query, as well as shorter snippets that directly answer the question, although some annotations may be absent, indicating no answer is available. Additionally, a small fraction of the dataset, about 1%, contains binary "yes" or "no" answers instead of detailed excerpts.

HotpotQA (Yang et al., 2018). It is a dataset featuring roughly 113,000 questions based on English Wikipedia, designed to need the lead sections from two articles for answers. Accompanying each question are the key paragraphs and a selection of sentences marked by crowdworkers as essential facts for responding. However, there are several examples that are missing some key paragraphs, and it causes the performance drop in RAG.

D.2 Evaluation Metrics

EM. Exact Match is a strict metric that measures the percentage of predictions that match the ground truth exactly. It is often used in the context of machine comprehension and question answering tasks where the goal is to produce an exact answer. For a given dataset, the Exact Match score is calculated by dividing the number of predictions that are exactly the same as the true answers by the total number of predictions made.

F1. The F1 score is a more nuanced metric that considers both precision and recall. It is the harmonic mean of precision and recall, providing a balance between the two.

D.3 Training Configuration

To validate the superiority of our method compared to train-based RAG, RAFT and golden-only finetuning methods followed the parameters as de-

Question: What do you practice in a dojo? Answer: martial art		
Method	Type	Contents
Standard RAG	context	Sport Gichin Funakoshi said, "There are no contests in karate." In pre–World War II Okinawa, kumite was not part of karate training. Shigeru Egami relates that, in 1940, some karateka were ousted from their dojo because they adopted sparring after having learned it in Tokyo. Karate is divided into style organizations. These organizations sometimes cooperate in non-style specific sport karate organizations or federations. Examples of sport organizations include AAKF/ITKF, AOK, TKL, AKA, WKF, NWUKO, WUKF and WKC. Organizations hold competitions (tournaments) from local to international level. Tournaments are designed to match
	context	and in some rare cases even time-limited grappling on the ground are also allowed. Free sparring is performed in a marked or closed area. The bout runs for a fixed time (2 to 3 minutes.) The time can run continuously (iri kume) or be stopped for referee judgment. In light contact or semi contact kumite, points are awarded based on the criteria: good form, sporting attitude, vigorous application, awareness/zanshin, good timing and correct distance. In full contact karate kumite, points are based on the results of the impact, rather than the formal appearance of the scoring technique.
RAG-EF	context	Karate appeared in the Soviet Union in the mid-1960s, during Nikita Khrushchev's policy of improved international relations. The first Shotokan clubs were opened in Moscow's universities. In 1973, however, the government banned karate—together with all other foreign martial arts—endorsing only the Soviet martial art of sambo. Failing to suppress these uncontrolled groups, the USSR's Sport Committee formed the Karate Federation of USSR in December 1978. On 17 May 1984, the Soviet Karate Federation was disbanded and all karate became illegal again. In 1989, karate practice became legal again, but under strict government regulations, only after the dissolution of the Soviet Union in 1991 did independent karate schools resume functioning, and so federations were formed and national tournaments in authentic styles began.
	context	Shigeru Egami, Chief Instructor of Shotokan Dojo, opined that "the majority of followers of karate in overseas countries pursue karate only for its fighting techniques...Movies and television... depict karate as a mysterious way of fighting capable of causing death or injury with a single blow... the mass media present a pseudo art far from the real thing." Shoshin Nagamine said, "Karate may be considered as the conflict within oneself or as a life-long marathon which can be won only through self-discipline, hard training and one's own creative efforts.
	Q&A	Question: When did karate practice become legal again in the Soviet Union? Answer: In 1989
	Q&A	Question: What do Shigeru Egami think is the main reason why people in overseas countries practice karate? Answer: They practice karate only for its fighting techniques.
	Q&A	Question: What did the JKA do for Bell's grades? Answer: They ratified his black belt on February 5, 1964, but he had to relinquish his Yoseikan grade
	Q&A	Question: What do organizations hold? Answer: Organizations hold competitions (tournaments) from local to international level.
	Q&A	Question: What is a useful tool to understand a kata? Answer: Bunkai

Table 7: **Qualitative results in bad scenario.** For the sake of space, we only selected top-2 context for both methods. Note that yello highlight indicates the answer for the given question, and it means that context or Q&A are relevant to the user query q.

scribed in Tab. 6. We adopted LoRA tuning (Hu et al., 2022) that is one of the parameter efficient finetuning methods. As a training set, we utilized $\mathcal{S}$, $\mathcal{E}$, and $\mathcal{O}$ of which sizes are 147,160, 725, and 725, respectively. Despite of imbalance problem of three datasets, both RAFT and golden-only methods get higher EM as using more relevant dataset as a training set, but they have lower performance than **RAG-EF** that is only updating the dataset into the database. Lastly, all the experiments for training are conducted on two Nvidia A5000s.

E Further Analysis

E.1 Inference Computation Cost

To assess the inference computation cost, we measured the inference time of **RAG-EF** and train-based RAG (*i.e.,* RAFT) using dense retrieval with DPR. The inference time was averaged over 100 examples after a warm-up run with another 100 examples, conducted on a server equipped with two NVIDIA A5000 GPUs and an Intel Xeon Gold 6342 CPU @ 2.80GHz. As shown in Tab. 8, while train-based RAG achieves faster retrieval (23.7 ms vs. 24.6 ms), it incurs higher generation latency (459.4 ms vs. 450.5 ms), leading to a slightly higher total inference time (483.2 ms vs. 475.2 ms). This increased generation time stems from the integration of LoRA into the base LLM in our setup. Although train-based RAG benefits from marginally faster retrieval, the overall time difference is negligible. Given the significant performance improvement of **RAG-EF**, this minor retrieval overhead is justifiable.

E.2 Bad Scenario in RAG-EF

We investigate the bad scenario where the retrieved question-answer pairs may be irrelevant to the final query, even when applying the retrieval strategy described in Eq. 2. As shown in Tab. 7, we observe that the retrieved chunks from **RAG-EF** still contain more relevant information related to the final query

Method	Retrieval	Generation	Total
RAFT	23.7	459.4	483.2
`RAG-EF`	24.6	450.5	475.2

Table 8: **Inference time (milliseconds) comparison between RAFT and `RAG-EF`.**

Context	Question	Answer	NQ	TriviaQA	HotpotQA
✓	✓		48.1	79.6	38.3
✓	✓	✓	47.3	79.2	37.9
✓	✓	✓	77.9	93.2	81.3

Table 9: **Impact of prompt component.** Note that ✓ and ✓ indicates correct and incorrect answer, respectively.

compared to standard RAG, which can potentially guide the model toward the correct answer.

E.3 Prompt Analysis

As shown in Appendix B, **`RAG-EF`** consists of three input prompt components: context, question, and answer. To investigate the impact of each component, we conducted experiments by incrementally adding different components: using only context and question, adding an incorrect answer, and finally adding a correct answer. Specifically, to generate an incorrect answer, we replaced the original answer with one from a different Q&A pair, ensuring that it does not match the question in the prompt. As summarized in Tab. 9, while adding an incorrect answer yields performance comparable to using only context and question, incorporating a correct answer significantly boosts performance. This suggests that **`RAG-EF`** is robust against incorrect Q&A pairs, as it primarily relies on the provided context.

E.4 Weighting Analysis in Eq. 2

The motivation for adopting the geometric mean in Eq. 2 is to enhance robustness against outliers. Since feedback generated by LLMs can sometimes include outliers due to hallucinations, the geometric mean is expected to mitigate their negative impact. To evaluate this, we conducted additional experiments using the arithmetic mean and summarized the results in Tab. 10. As shown in the table, the performance differences between the arithmetic and geometric means are negligible: for example, on Natural Question, the arithmetic mean achieved 50.5 while the geometric mean achieved 50.0; on TriviaQA, the scores were 81.5 and 81.6 respectively; and on HotpotQA, both methods yielded 40.0. These results suggest that, although the geometric weighting approach is theoretically more robust against outliers, both methods perform comparably in practice.

Method	NQ	TriviaQA	HotpotQA
Arithmetic	50.5	81.5	40.0
Geometric	50.0	81.6	40.0

Table 10: **Weighting analysis in Eq. 2.**

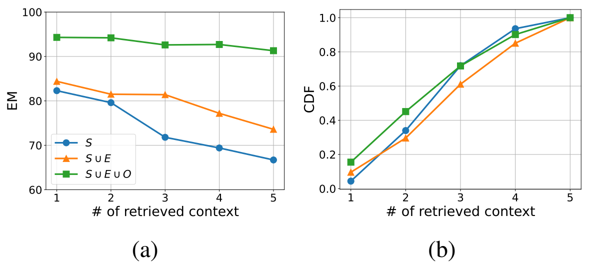

Figure 3: **(a) performance and (b) distribution changes as the number of retrieved context increases on TriviaQA.**

E.5 Impact of Varying Context Count

Unlike Standard RAG, **`RAG-EF`** allows the number of retrieved contexts to vary from one to five, potentially including duplicate contexts. This raises a question regarding *how the performance changes as the number of retrieved contexts varies*. As shown in Fig. 3a, the performance decreases when more contexts are retrieved. While larger retrieval sets might intuitively increase the chance of including a golden context, the additional noise and duplicates can confuse the model and degrade overall performance. By contrast, retrieving fewer contexts tends to preserve higher-quality, high-ranking documents, thus raising the probability—or coverage—of golden contexts. Furthermore, Fig. 3b illustrates that most queries retrieve between two and four contexts, suggesting a practical sweet spot where the coverage of golden contexts remains relatively high without incurring excessive noise. Overall, these findings indicate that simply increasing the number of retrieved contexts does not guarantee better performance, underscoring the importance of balancing coverage with noise reduction.