RaFe: Ranking Feedback Improves Query Rewriting for RAG

Shengyu Mao[♠], Yong Jiang[♡,*], Boli Chen[♡], Xiao Li[◇], Peng Wang[♠], Xinyu Wang[♡], Pengjun Xie[♡], Fei Huang[♡], Huajun Chen[♠], Ningyu Zhang[♠*]
[♠]Zhejiang University [♡]Alibaba Group, [◇]Nanjing University
{shengyu,zhangningyu}@zju.edu.cn , yongjiang.jy@alibaba-inc.com

Abstract

As Large Language Models (LLMs) and Retrieval Augmentation Generation (RAG) techniques have evolved, query rewriting has been widely incorporated into the RAG system for downstream tasks like open-domain QA. Many works have attempted to utilize small models with reinforcement learning rather than costly LLMs to improve query rewriting. However, current methods require annotations (e.g., labeled relevant documents or downstream answers) or predesigned rewards for feedback, which lack generalization, and fail to utilize signals tailored for query rewriting. In this paper, we propose RaFe, a framework for training query rewriting models free of annotations. By leveraging a publicly available reranker, RaFe provides feedback aligned well with the rewriting objectives. Experimental results demonstrate that RaFe can obtain better performance than baselines.

1 Introduction

Large Language Models (LLMs) have demonstrated strong capacities to solve a variety of tasks (Zhao et al., 2023). However, they still encounter the challenges of hallucinations (Ji et al., 2023; Zhang et al., 2023; Huang et al., 2023) or outdated knowledge (Yao et al., 2023; Zhang et al., 2024). Recently, Retrieval Augmentation Generation (RAG) (Gao et al., 2023) has become an important technology to enhance LLMs' abilities, by incorporating external knowledge. For instance, in open-domain QA, LLMs can first retrieve related documents and then generate answers. Nonetheless, directly retrieving by original query does not always achieve correct and relevant documents. Therefore, query rewriting (Efthimiadis, 1996; Carpineto and Romano, 2012) has been widely employed to reformulate the query to expand the retrieved documents for a better response as illustrated in Figure 1.

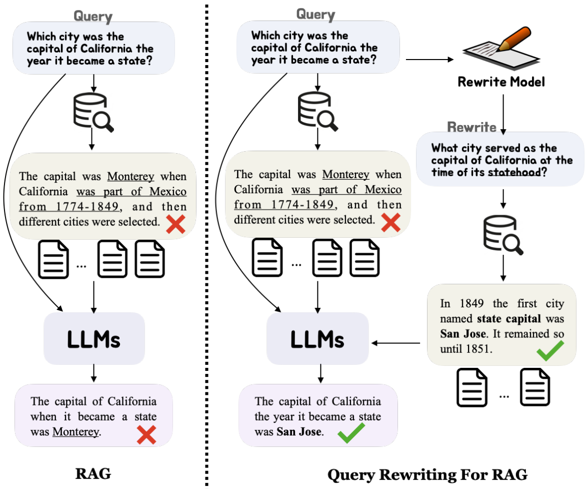

Query
Which city was the capital of California the year it became a state?
The capital was Monterey when California was part of Mexico from 1774-1849, and then different cities were selected.
...
LLMs
The capital of California when it became a state was Monterey.
RAG
Query
Which city was the capital of California the year it became a state?
Rewrite Model
Rewrite
What city served as the capital of California at the time of its statehood?
The capital was Monterey when California was part of Mexico from 1774-1849, and then different cities were selected.
...
In 1849 the first city named state capital was San Jose. It remained so until 1851.
...
LLMs
The capital of California the year it became a state was San Jose.
Query Rewriting For RAG

Figure 1: Illustration of query rewriting for RAG. The left part indicates the normal RAG pipeline, while the right part presents the query rewriting to expand more relevant documents for RAG.

Many efforts have been proposed to leverage the powerful LLMs to generate rewrites (Shen et al., 2023; Wang et al., 2023) directly. While in practical applications, it is more prevalent to implement specific small query rewriting models to avoid the costly use of LLMs (Ma et al., 2023). To improve the performance of query rewriting, reinforcement learning (RL) with feedback (Wu et al., 2022; Chen et al., 2022) can be utilized as a typical solution. For instance, Nogueira and Cho (2017) generates feedback by considering the recall of labeled documents. Meanwhile, Ma et al. (2023) leverages evaluation results from question answering (QA) post-rewriting to generate signals. Additionally, Peng et al. (2023) employs domain-specific annotated rewriting scores for feedback training.

Note that these feedback-driven query rewriting methods rely on either annotated labels such as relevant documents or answers, or pre-designed rewards tailored to specific domains. However, they often lack the utilization of effective and general signals for query rewriting. Meanwhile, consider-

[*] Corresponding Author.

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 884–901
November 12-16, 2024 ©2024 Association for Computational Linguistics

able efforts have been made to harness diverse feedback mechanisms across various domains (Nathani et al., 2023; Li et al., 2023). Notably, Liu et al. (2023b) effectively integrates unit testing feedback into code generation, yielding significant efficacy. Drawing from these, in this paper we attempt to (i) reduce the **cost of annotations for feedback**; and (ii) identify **a signal that better aligns with the objectives of the query rewriting** task.

To address these issues, we introduce **RaFe** (**Ra**nking **Fe**edback improves Query Rewriting), a novel framework that leverages feedback from the reranker to train query rewriting models. This approach is inspired by the reranker module in traditional information retrieval (IR) systems, which scores and sorts retrieved documents based on the query. Intuitively, query rewriting aims to retrieve documents relevant to the original query, which aligns perfectly with the goal of the reranker. Specifically, the reranker is capable of scoring documents without requiring additional labels. Thus, we incorporate a reranker to provide feedback for the query rewriting model.

RaFe comprises a two-stage process. We first train an initial query rewriting model by standard supervised fine-tuning. Subsequently, we utilize the ranking scores from the reranker to conduct feedback training on the query rewriting model. RaFe supports both offline and online RL feedback training. Empirically, we demonstrate that utilizing a general, publicly available reranker, RaFe can drive the training of the query rewriting model, indicating the effectiveness and potential generalizability of the proposed approach. The main contributions of our paper can be summarized as follows:

- We propose RaFe, a novel query rewriting framework that utilizes feedback from the reranker, an especially fitting signal for the objective of retrieving more relevant documents.

- RaFe does not necessitate annotated labels or particularly designed scores, ensuring the generalizability of the training framework.

- We validate the effectiveness of our proposed approach on cross-lingual datasets across wide settings with a general and public reranker, we further conduct a comprehensive investigation of what makes a better query rewriting and how ranking feedback works.

2 Method

2.1 Task Formulation

Within the process of Retrieval Augmented Generation (RAG), when inputting an original query q, a set of relevant documents $D = [d_0, d_1, ..., d_k]$ will be retrieved through a search engine, and the retrieved documents are utilized to enable the model to better accomplish the corresponding task (in this paper, we discuss the task of Open-domain Question Answering). Query rewriting is to reformulate the original query q into another form to better retrieve relevant passages. We aim to obtain a better rewrite model $\mathcal{M}_\theta$ that can rewrite q as:

$$q' = \mathcal{M}_\theta(q), \tag{1}$$

here q' is the rewritten query that is used to retrieve documents D' for completing subsequent tasks. Figure 2 shows the overview of our proposed framework, RaFe for query rewriting training.

2.2 Initial Supervised Fine-Tuning

Before leveraging the ranking feedback, we first initialize the rewrite model with a cold start supervised fine-tuning to gain the rewrite ability. Specifically, we prompt the LLMs to produce the rewrite data, The rewrites generated from LLMs[1] are denoted as $T_{\text{all}} = \{(q, q')|q' \in Q'\}$, here Q' is the rewrite set of original query q. We split the training instances into two parts $T_{\text{all}} = [T_{\text{sft}} : T_{\text{f}}]$, here T_{sft} and T_{f} indicates the instances we use for SFT and feedback, respectively, to separate the query for SFT and feedback and conduct a fair comparison in the following experiments. We train the rewrite model $\mathcal{M}_\theta$ with standard SFT loss as follows:

$$\mathcal{L}_{\text{sft}} = -\sum\nolimits_{q' \in Q'} \sum\nolimits_{t} \log \mathcal{M}_\theta(q'_t | q'_{<t}, q). \tag{2}$$

Note that for each query, we mix all corresponding rewrites together in the dataset for training, to enhance the diversity of generation by our trained model, since in real-world applications, different rewrites are required for a single search query to address different aspects or interpretations.

2.3 Feedback Training

The evaluation of query rewriting is notoriously difficult due to the absence of direct quality assessment methods (Zhu et al., 2023), so previous feedback for QR typically relies on the annotated passages (Nogueira and Cho, 2017; Wu et al., 2022).

[1] We prompt Qwen-max for all the pre-generated rewrites

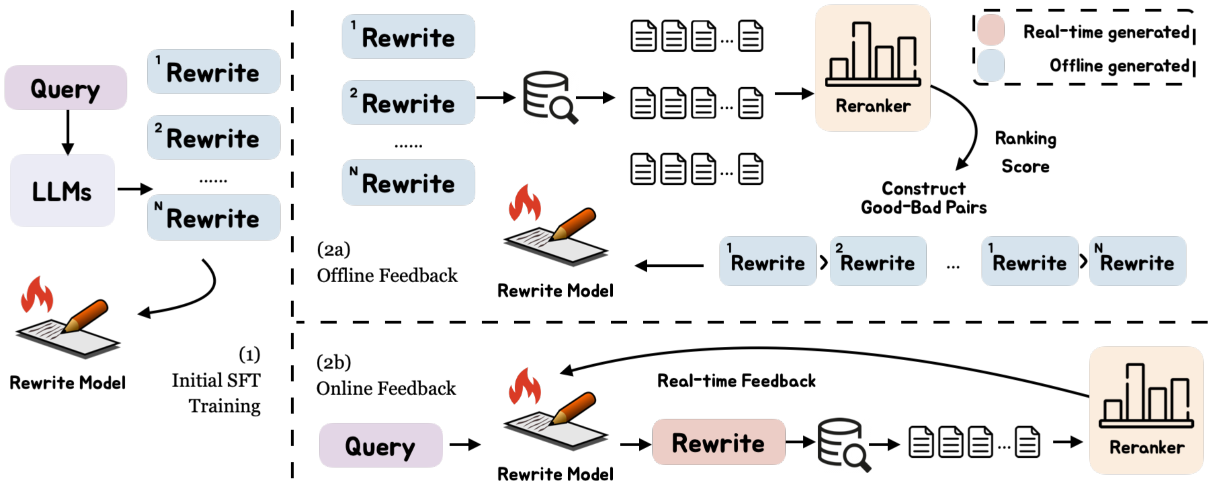


Figure 2: The overview of **RaFe**. The entire procedure consists of two stages: the initial SFT, and subsequent feedback training. RaFe obtains ranking feedback aligned with the goal of query rewriting without annotated data and enables leveraging the feedback in two ways. **Offline training**: Constructing good-bad pairs from offline-generated data. **Online training**: Scoring queries generated in real-time and complete feedback training.

Throughout the traditional IR pipeline, documents expanded by query rewriting are typically subjected to a reranking process. Intuitively, the reranker can serve as a natural feedback for query rewriting. Given a reranker model $\mathcal{M}_r$, the process of scoring a document d with query q can be formulated as $\mathcal{M}_r(q, d)$. The ranking score of a rewrite q' can be denoted as follows:

$$S(q, q') = \frac{1}{|D'|} \sum\nolimits_{d' \in D'} \mathcal{M}_r(q, d'), \quad (3)$$

here D' indicates the documents retrieved by q', and we constrain $|D'| \leq 5$ for computing the scores on top-5 documents. In this way, we can provide reliable feedback for training rewriting models. As illustrated in Figure 2, our proposed method can be applied to both offline and online feedback training.

Offline Feedback For offline feedback, we leverage the ranking score of each document retrieved by a rewritten query to construct the preference data. Specifically, we set a threshold to distinguish the good and bad rewrites formulated as μ, which is computed as the average ranking score for all training instances as follows:

$$\mu = \frac{1}{|T_{\mathrm{f}}|} \sum\nolimits_{(q,q') \in T_{\mathrm{f}}} S(q, q'). \quad (4)$$

Then for every rewrite q' with a score exceeding the threshold μ, we regard it as a good rewrite for the original query q; otherwise, it is deemed a bad rewrite. In this way, we obtain all the preference pairs for open domain QA in the form (q, q'_g, q'_b).

For the offline feedback training, we use DPO (Rafailov et al., 2023) and KTO (Kawin et al., 2023). DPO directly leverages the preference pairs to optimize the model, while KTO is a method that can optimize the model from feedback, only needs the signal of whether a rewrite $q^{'}$ is good or not, rather than needing pairs, formulated as $(q, q^{'}; \rho), \rho \in$ [good, bad]. The specific formulation of $\mathcal{L}_{kto}$ is in Eq 6, and the detailed explanation of the KTO is demonstrated in Appendix A.2.1.

Online Feedback The ranking score can also serve as an online feedback signal. We utilize the Proximal Policy Optimization (PPO) (Schulman et al., 2017) algorithm to implement online feedback training. The training process includes rewriting, retrieving, scoring and ultimately providing feedback, as illustrated in Figure 2(2b). The details of the PPO loss and implementation are provided in Appendix A.2.1.

3 Experimental Setup

As we attempt to improve query rewriting for better RAG, we conduct our experiments on the typical RAG scenarios, Open-Domain Question Answering (ODQA). The process of RAG for ODQA can be formulated as $\mathcal{F}([D : q])$, where $\mathcal{F}$ denotes the LLMs, q is the original query from datasets and D is the documents concatenated for augmentation.

Method	EN								ZH			
	FreshQA		NQ		TriviaQA		HotpotQA		FreshQA		WebQA	
	QA	Prec@5	QA	Prec@5	QA	Prec@5	QA	Prec@5	QA	Prec@5	QA	Prec@5
w/o retrieval	41.70	-	43.74	-	74.99	-	34.80	-	40.98	-	73.95	-
OQR	61.87	27.48	51.36	32.35	79.63	50.32	42.75	17.73	43.70	16.24	81.29	77.25
SUBSTITUTE-Raw												
LLM-Rewrite	57.38	25.23	48.62	29.83	78.43	48.10	40.92	15.32	40.65	15.42	80.56	74.26
Query2Doc	56.52	26.08	46.12	27.65	77.22	50.58	38.85	16.26	42.90	15.20	**81.35**	77.63
SFT$_{(T_{\text{sft}})}$	60.53	25.72	49.86	30.08	78.34	47.77	42.04	16.46	42.44	15.56	77.76	72.65
SFT$_{(T_{\text{all}})}$	60.55	24.88	50.39	30.40	78.63	47.92	42.66	16.89	42.33	15.21	77.80	74.61
RaFe$_{(PPO)}$	62.21	27.72	50.83	31.52	78.56	49.18	43.82	17.64	43.28	16.31	81.28	**77.90**
RaFe$_{(DPO)}$	**62.67**	27.92	51.14	32.25	**79.84**	50.67	**43.82**	**18.91**	**45.25**	**16.92**	80.61	75.37
RaFe$_{(KTO)}$	62.12	**28.12**	**51.61**	**32.71**	79.51	**51.12**	43.27	18.28	45.03	16.40	81.17	76.98
EXPAND-Raw												
LLM-Rewrite	61.17	27.52	51.56	31.79	80.20	50.29	44.50	18.01	45.13	16.98	81.30	78.12
Query2Doc	61.46	27.64	50.75	30.83	80.54	50.04	44.49	18.75	46.68	17.44	81.33	79.48
SFT$_{(T_{\text{sft}})}$	62.01	26.76	50.13	30.63	80.42	50.21	44.93	18.78	47.15	17.82	81.26	71.95
SFT$_{(T_{\text{all}})}$	62.21	26.36	51.79	31.45	80.57	50.24	44.89	18.99	47.51	17.54	81.49	72.48
RaFe$_{(PPO)}$	62.43	28.31	51.63	31.81	80.32	50.01	45.28	18.87	47.53	**18.22**	**82.45**	**80.15**
RaFe$_{(DPO)}$	62.39	28.16	52.30	32.53	80.64	50.92	45.59	19.25	47.25	17.92	81.73	78.85
RaFe$_{(KTO)}$	**62.65**	**28.50**	**52.48**	**32.58**	**80.88**	**51.24**	**45.91**	**19.52**	**47.93**	18.11	82.16	77.66

Table 1: The results showcase the performance in SUBSTITUTE-Raw and EXPAND-Raw settings. "QA" refers to results obtained by Qwen-max, and "w/o retrieval" denotes generating answers directly. Results surpassing the OQR are highlighted in bold to represent the best-performing, while those underlined indicate the second-best.

3.1 Dataset

To comprehensively validate the effectiveness and generalizability of our method, we conduct cross-lingual experiments. Specifically, we evaluate ReFe on both English and Chinese datasets.

English Datasets For English data, we use several open-domain QA datasets including NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), HotpotQA (Yang et al., 2018). For NQ and TriviaQA, we follow the split from previous work (Karpukhin et al., 2020), and the default split for HotpotQA[2]. We randomly gather 60k instances from the training set of the three datasets to conduct T_{all} for training rewrite models. As for evaluation, we collect the test set of NQ and TriviaQA, and the development set of HotpotQA as the held-in evaluation datasets. Additionally, we use FreshQA (Vu et al., 2023) for out-of-domain evaluation.

Chinese Datasets For Chinese data, we gather several open-source queries to conduct the query set, the sources are listed in 6. We use WebQA (Li et al., 2016) for the in-domain evaluation, while FreshQA (Vu et al., 2023) (translated) for the out-of-domain evaluation. The process of translation can be found in Appendix A.2.2.

3.2 Evaluation Settings

In practical retrieval scenarios, query rewriting is commonly used to expand the retrieved documents based on the original query, followed by a re-ranking of the expanded documents. Thus, we validate RaFe in two experimental settings.

SUBSTITUTE Directly use the documents D' retrieved by rewrite q' for evaluation instead of the documents D retrieved by query q.

EXPAND Employing both D and D' for evaluation. We generate two rewrites q'_0, q'_1 for the EXPAND setting with their retrieved D'_0, D'_1.

To further simulate the role of query rewriting in real-world scenarios, our experiments also include the performance under two following settings:

Raw Concatenating top-5 retrieved documents in the default order. For EXPAND setting, the raw documents order is determined by sequentially and cyclically selecting the top documents from D, D'_0, D'_1.

Ranked Concatenating top-5 documents after reranking all the retrieved documents. As regard to EXPAND setting, all retrieved documents from both the query and rewrites are merged for ranking.

[2] https://huggingface.co/datasets/hotpot_qa/viewer/fullwiki

Method	EN								ZH			
	FreshQA		NQ		TriviaQA		HotpotQA		FreshQA		WebQA	
	QA	Prec@5	QA	Prec@5	QA	Prec@5	QA	Prec@5	QA	Prec@5	QA	Prec@5
OQR	62.56	30.88	51.50	35.68	80.17	52.57	43.21	18.32	44.67	17.27	81.37	78.27
SUBSTITUTE-Ranked												
LLM-Rewrite	59.24	27.34	49.75	32.27	78.53	50.43	41.48	16.37	42.85	16.26	80.53	76.32
Query2Doc	58.84	28.32	45.62	30.59	77.26	52.01	42.26	17.73	43.81	16.61	81.22	**79.92**
SFT$_{(T_{\text{sft}})}$	60.69	28.42	50.99	34.01	78.35	50.19	42.26	17.64	43.44	16.56	77.72	74.65
SFT$_{(T_{\text{all}})}$	61.42	28.40	50.93	32.54	78.15	50.33	42.66	17.88	44.40	16.20	78.16	75.61
RaFe$_{(PPO)}$	**63.01**	30.56	51.26	34.61	98.86	51.33	42.57	18.45	43.77	16.79	**81.46**	76.90
RaFe$_{(DPO)}$	62.89	30.28	**51.97**	**35.89**	**80.41**	**53.54**	43.77	19.07	**45.49**	**17.58**	80.53	76.37
RaFe$_{(KTO)}$	62.71	**31.00**	51.86	35.62	80.23	53.09	**44.77**	**19.82**	45.30	17.36	81.14	77.98
EXPAND-Ranked												
LLM-Rewrite	62.34	31.14	51.55	36.34	80.79	54.93	45.73	20.85	45.83	17.52	82.29	78.21
Query2Doc	63.06	31.84	51.83	37.16	80.28	54.47	45.82	23.05	46.58	18.29	83.35	80.75
SFT$_{(T_{\text{sft}})}$	63.16	31.56	51.75	37.44	80.17	54.20	45.18	22.28	47.61	18.86	82.08	79.15
SFT$_{(T_{\text{all}})}$	63.27	28.44	51.94	37.68	80.88	54.25	45.84	22.09	46.95	18.63	82.75	79.43
RaFe$_{(PPO)}$	**64.96**	33.54	52.36	38.44	81.38	55.27	46.73	22.39	48.83	**19.66**	**83.58**	**80.93**
RaFe$_{(DPO)}$	63.98	33.20	52.74	**38.57**	81.74	55.60	46.53	22.78	48.72	18.58	83.04	79.83
RaFe$_{(KTO)}$	64.85	**33.72**	**52.86**	38.37	**81.97**	**55.67**	**46.79**	**23.35**	**48.96**	19.25	82.96	79.52

Table 2: Results of SUBSTITUTE-Ranked and EXPAND-Ranked settings. "OQR" is evaluated after ranking.

We utilize the Exact Match (EM) metric to evaluate the general QA performance. Especially, we use Rouge-L (Lin, 2004) to evaluate the *false premise* set in FreshQA. Given our work focus on open-domain QA, there are no gold documents or relevant annotations, we evaluate the retrieval by determining whether the retrieved documents contain the correct answer. We report the Precision@K and the mean reciprocal rank (MRR) in the results.

3.3 Baseline

Original Query Retrieval (OQR) Retrieve with the original query and utilize the documents by the default returned ranking from the search engine.

LLM Rewrite Directly enable the LLMs to rewrite the original query with a few-shot prompt. In our experiment, we prompt Qwen-max to rewrite the original query.

Query2Doc (Wang et al., 2023) A method creates pseudo-documents through few-shot prompting of LLMs and then the query is expanded with the generated pseudo-documents for retrieving. The used prompts are shown in Appendix A.5.

SFT Use the pre-generated rewrites to directly train the rewrite model. SFT$_{(T_{\text{sft}})}$ represents the rewrite model trained specifically on the T_{sft}, while SFT$_{(T_{\text{all}})}$ denotes the model trained on T_{all}.

3.4 Implementation

Retriever We use an anonymous internal search engine for open domain to retrieve documents for the Chinese datasets, and Google Search for the English datasets. Specifically, we utilize the title and the summary snippet of the searched page as the retrieved documents for retrieval augmentation.

Base Model We employ Qwen-max (Bai et al., 2023) to generate responses and conduct the evaluation with Qwen1.5-32b-chat. Query rewriting models are trained with the Qwen-7b-base.

Reranker For a general RAG task like open-domain QA, If our approach yields positive results with a general reranker, it will perform even better when transferring to a specific domain (where a domain-specific reranker is available). Thus, we employ a publicly available bge-reranker[3] (Xiao et al., 2023) to conduct open-domain QA experiments, which serves to demonstrate the effectiveness of the methods we designed.

4 Results

4.1 Main Result

From Table 1 and Table 2, we can observe that RaFe outperforms other query rewriting baselines and OQR across almost all settings in retrieval and question-answering metrics. It can be noted that

[3] https://huggingface.co/BAAI/bge-reranker-base

Methods	FreshQA		NQ	
	Raw	Ranked	Raw	Ranked
OQR	61.87	62.56	51.36	51.50
SUBSTITUTE				
SFT$_{(T_{\text{all}})}$	60.55	61.42	50.39	50.93
Precision$_{(DPO)}$	60.43	61.03	49.32	50.65
Precision$_{(KTO)}$	60.54	61.34	49.76	50.12
LLM$_{(DPO)}$	61.95	62.45	50.94	51.44
LLM$_{(KTO)}$	**62.32**	62.39	51.34	51.54
RaFe$_{(KTO)}$	62.12	**62.71**	**51.61**	**51.86**
EXPAND				
SFT$_{(T_{\text{all}})}$	61.42	63.27	51.79	51.94
Precision$_{(DPO)}$	61.56	62.84	50.34	51.29
Precision$_{(KTO)}$	61.79	63.15	50.69	51.32
LLM$_{(DPO)}$	62.43	63.53	51.63	52.43
LLM$_{(KTO)}$	61.87	64.08	51.89	52.23
RaFe$_{(KTO)}$	**62.65**	**64.85**	**52.48**	**52.86**

Table 3: Results compared with different feedback, Precision and LLM indicates the retrieval feedback and LLM feedback, respectively.

the performances of most methods decrease slightly compared to OQR under the SUBSTITUTE setting, where RaFe also shows marginal improvements. The weak performance might be attributed to that rewriting tend to deviate from the original query in some challenging cases. We provide a deeper analysis in the Appendix A.4.1.

While under the EXPAND setting, the majority of baseline methods perform better than under SUBSTITUTE setting. Notably, RaFe achieves significant improvements in the Expand-Ranked setting, where the QA results surpass all other baselines including OQR by 2%-3%. A similar conclusion can be drawn from Table 8. By comparing results between Table 1 and Table 2, it can be found that even with feedback provided to the query rewriting models through the use of rerankers, the ranked results continue to show a substantial increase in performance, which are further illustrated in Figure 4. It suggests that in practical applications of RAG, it may yield the greatest benefit by employing query rewriting with the EXPAND-Ranked setting. More retrieval results are shown in Appendix A.3.1.

4.2 Compared with Other Types of Feedback

Previous work on training query rewrite models for the RAG (Ma et al., 2023) has leveraged LLMs performance on QA tasks as the feedback signal. Many works construct feedback based on retrieval metrics from annotated documents (Wu et al., 2022; Nogueira and Cho, 2017). To thoroughly assess the efficacy of our approach, we also experiment with these types of feedback. We obtain good-bad pairs (i.e. true for good and false for bad) for offline training introduced in Sec 2.3. We use Qwen-32b-chat to conduct the LLM feedback. For the retrieval feedback, we utilize the results of Prec@5 to obtain good-bad pairs. The results are shown in Table 3. Additionally, we provide a comparison between reranker feedback and other feedback, demonstrated in Table 4.

Methods	Feedback	Annotation	Cost
LLM	QA Results	yes	78h
Precision	Retrieval	yes	0.01h
RaFe	Reranker	no	0.67h

Table 4: The comparison of different types of Feedback. **Annotation** indicates whether the labeled data is needed for the feedback signals. The **Cost** means the time for constructing the feedback for 30k instances.

The results show that RaFe outperforms the other two types of feedback. Precision feedback yields the worst results, which may be attributed to the rudimentary construction of precision in our dataset—merely considering whether the answer is present within the document. LLM feedback also demonstrates competent performance in the SUBSTITUTE setting. However, from Table 4, we notice that under an equivalent data volume, the cost of employing LLM to construct feedback substantially exceeds that of the other two feedback.

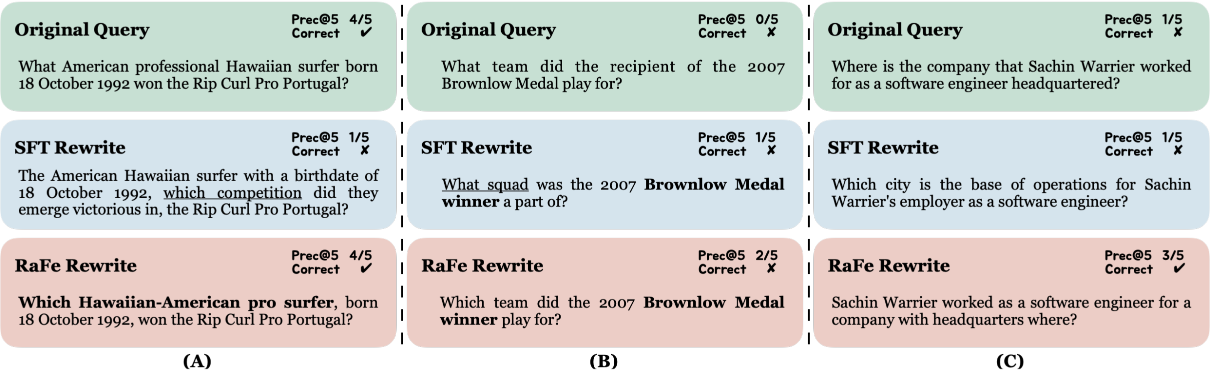
5 Analysis

5.1 How RaFe makes rewriting better?

In this section, we present illustrative case studies to intuitively compare different rewrites and the original query in Figure 3. The benifits of RaFe can be summarized into three types.

(A): RaFe performs better in preserving the semantics of the original query. As shown in Figure 3 (A), it can be observed that RaFe, after alignment through reranker, can rewrite queries in a way that better preserves the semantics of the original query. In contrast, the rewrite by SFT directly shifts the focus of the query from which athlete to which competition.

(B): RaFe's rewrites improve the format of the query for retrieval purposes. RaFe's rewrite is capable of transforming an uncommon term "recipient" into "winner". Although SFT rewrites also re-



Original Query Prec@5 4/5 Correct ✔

What American professional Hawaiian surfer born 18 October 1992 won the Rip Curl Pro Portugal?

SFT Rewrite Prec@5 1/5 Correct ✘

The American Hawaiian surfer with a birthdate of 18 October 1992, which competition did they emerge victorious in, the Rip Curl Pro Portugal?

RaFe Rewrite Prec@5 4/5 Correct ✔

Which Hawaiian-American pro surfer, born 18 October 1992, won the Rip Curl Pro Portugal?

(A)

Original Query Prec@5 0/5 Correct ✘

What team did the recipient of the 2007 Brownlow Medal play for?

SFT Rewrite Prec@5 1/5 Correct ✘

What squad was the 2007 **Brownlow Medal winner** a part of?

RaFe Rewrite Prec@5 2/5 Correct ✘

Which team did the 2007 **Brownlow Medal winner** play for?

(B)

Original Query Prec@5 1/5 Correct ✘

Where is the company that Sachin Warrier worked for as a software engineer headquartered?

SFT Rewrite Prec@5 1/5 Correct ✘

Which city is the base of operations for Sachin Warrier's employer as a software engineer?

RaFe Rewrite Prec@5 3/5 Correct ✔

Sachin Warrier worked as a software engineer for a company with headquarters where?

(C)

Figure 3: Three types of examples, including the original query and rewrites from SFT and RaFe. The Prec@5 results of queries and rewrites are presented, and **“Correct”** denotes that whether the prediction is correct or not.

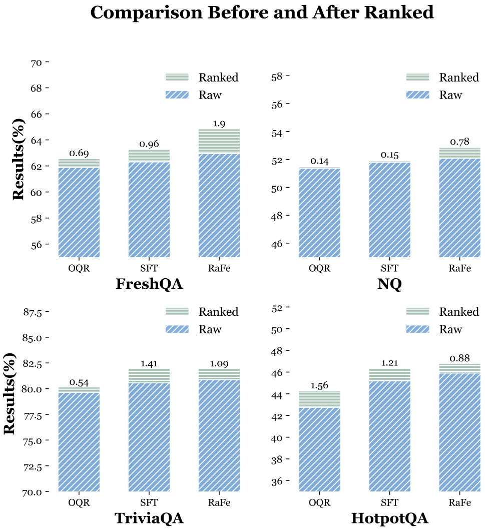


Figure 4: The performance of different rewrite models before and after all the documents are reranked under EXPAND setting. The number displayed on each bar represents the specific improvement from Raw to Ranked.

place “recipient” with “winner”, it changes “team” from a sports competition context to “squad”, a term commonly used in military, police, or other contexts, thereby introducing potential ambiguity.

(C): RaFe’s rewrites sentences for better understanding. This kind of case is not easily discernible as good or bad based on intuition; however, RaFe’s rewrite demonstrates better performance in retrieval results. Such cases show why we require feedback to enhance the QR effectiveness, as we always fail to articulate how a query could be formatted to better suit a retriever.

5.2 How does the Reranker Feedback Work?

To investigate how reranker works for query rewriting, we first ascertain the ability of the publicly

Methods	Prec@5	Prec@10	MRR
Original Query	41.41	39.76	54.11
Bad Rewrite	30.74	28.13	43.64
Good Rewrite	**46.14**	**44.34**	**59.17**

Table 5: The comparison of retrieval results between original query and good/bad rewrites.

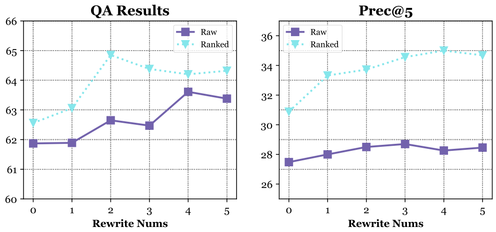


Figure 5: The results of different rewrite nums in EXPAND setting. We list the result from 0 to 5 rewrites. The rewrites are generate by RaFe$_{(KTO)}$.

available reranker to rank on unseen datasets.

The comparing results are presented in Figure 4. It can be clearly seen that all methods yield better QA performance after documents are ranked on all the datasets. This indicates that the reranker’s pattern for document sorting acts as a positive signal for the retrieval system. Meanwhile, we can observe that RaFe performs the better improvements after ranked, which further demonstrates the effectiveness of reranker feedback.

Moreover, we validate the effectiveness of reranker in constructing good and bad pairs within T_{f}. We compare the precision of documents retrieved by different queries in Table 5. It is obvious that the documents retrieved by good rewrites exhibit significantly higher precision compared to those retrieved by the original query, which indicates that the reranker is capable of effectively distinguishing between rewrites that can retrieve high-quality documents and those that cannot. We also provide some examples in Appendix A.4.2.

5.3 How Many Rewrites is Optimal for RAG?

In this section, we delve deeper into the impact that varying numbers of rewrites have on the final performance, since in practical applications of query rewriting, a balance must be struck between the quantity of generated rewrites and performance efficiency, given that generating more rewrites could potentially result in more response time. We generate different numbers of rewrites, the results are depicted in Figure 5. The QA results peak when there are 4-5 rewrites, suggesting that employing more rewrites can yield considerable benefits by retrieving more relevant top documents. However, Prec@5 nearly approaches the best around 2-3 rewrites. When ranking all passages, the performance ceiling is attained with merely 2 rewrites. Considering the time cost, 2-3 rewrites may benefit the most for practical RAG.

Meanwhile, it can be observed that there's a drop when increasing the rewrites from 4 to 5, we provide further analysis in Appendix A.4.5.

6 Related Work

6.1 Query Rewriting

Query rewriting is a critical technique within the retrieval domain (Carpineto and Romano, 2012; Zhu et al., 2023). With the groundbreaking advancements in scaling-up model capabilities, query rewriting has also played a pivotal role in enhancing the abilities of LLMs in RAG (Khattab et al., 2022; Press et al., 2023; Yan et al., 2024). Many works (Wang et al., 2023; Shen et al., 2023; Ye et al., 2023) directly leverage LLMs' strong capabilities to expand or rewrite queries. Nonetheless, in practical application scenarios, a smaller rewriting model is preferred to avoid the costly requests for LLMs. At the same time, feedback training is the most commonly employed method to enhance the smaller rewriting models. Nogueira and Cho (2017) incorporates the ranking signals from annotated passages for better results, as well as previous works on conversational query rewrite (Wu et al., 2022; Mo et al., 2023; Chen et al., 2022). Ma et al. (2023) first generates answers from LLMs and then uses the QA evaluation results as the training signals. Peng et al. (2023) leverages search scoring functions intrinsic to the e-commerce framework to assess rewrite quality, informing feedback signals, which is exceedingly domain-specific, limiting its applicability to other domains.

These works depend on using particularly designed scores or annotated labels for feedback signals, while our proposed method can generically deliver feedback based on ranking results, without needing annotated passages.

6.2 Learning From Feedback

Recent advancements in Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) have been instrumental in aligning the generative capabilities of large models with human preferences, significantly prompting the creation of strong LLMs (OpenAI, 2023). Therefore, a large number of studies about feedback alignment have been emerging (Zheng et al., 2023; Wang et al., 2024; Rafailov et al., 2023; Yuan et al., 2023; Dong et al., 2023; Kawin et al., 2023). Some research efforts are concentrated on devising methods to provide new forms of feedback (Lee et al., 2023; Shinn et al., 2023; Madaan et al., 2023; Pang et al., 2023; Liu et al., 2023a; Akyürek et al., 2023; Nathani et al., 2023). Xu et al. (2023) propose to train models from judgment language feedback. Li et al. (2023) designs two types of ranking feedback drawing from LLMs, to improve the performance.

Despite all these works, the exploration of feedback in rewriting is currently limited to direct feedback from LLMs (Ma et al., 2023) and domain-specific scoring (Peng et al., 2023). Such feedback approaches are costly and fail to utilize the effective signals from the IR system. While Le et al. (2022) and Liu et al. (2023b) effectively leverage the feedback from Unit Test in the domain of code generation, we investigate more appropriate feedback signals for query rewriting in this paper, the reranker feedback.

7 Conclusion and Future Work

This paper proposes a novel feedback training framework named **RaFe** for query rewriting, based on the effectiveness of the reranker in enhancing document ranking during the information retrieval process. By leveraging the feedback signals from reranker, RaFe is capable of effectively and generally conducting feedback training for rewrite models, yielding great improvements. Experimental results indicate that our method achieves exemplary performance across cross-linguistic datasets. In the future, we plan to conduct joint training of reranker and rewrite models, which may yield substantial benefits for RAG.

Limitations

Although our experiments employ a general reranker as the source of feedback signals, there are still some limitations. (1) The Lack of Cross-Domain Validation. As constrained by the lack of domain-specific data, we lack the validation of separately trained rerankers on datasets pertinent to a specific domain. (2) Reliance on the Effectiveness of Rewriting as a Bottleneck. Although we can achieve some improvements by using publicly available rerankers, this enhancement may be limited by the capability of the reranker.

References

Afra Feyza Akyürek, Ekin Akyürek, Ashwin Kalyan, Peter Clark, Derry Tanti Wijaya, and Niket Tandon. 2023. RL4F: generating natural language feedback with reinforcement learning for repairing model outputs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 7716–7733. Association for Computational Linguistics.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *CoRR*, abs/2310.11511.

Jinze Bai, Shuai Bai, Yunfei Chu, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50.

Zhiyu Chen, Jie Zhao, Anjie Fang, Besnik Fetahu, Oleg Rokhlenko, and Shervin Malmasi. 2022. Reinforced question rewriting for conversational question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: EMNLP 2022 - Industry Track, Abu Dhabi, UAE, December 7 - 11, 2022*, pages 357–370. Association for Computational Linguistics.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. RAFT: reward ranked finetuning for generative foundation model alignment. *CoRR*, abs/2304.06767.

Efthimis N Efthimiadis. 1996. Query expansion. *Annual review of information science and technology (ARIST)*, 31:121–87.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *CoRR*, abs/2312.10997.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *CoRR*, abs/2311.05232.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12):248:1–248:38.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. 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.

Ethayarajh Kawin, Xu Winnie, Jurafsky Dan, and Douwe Kiela. 2023. Human-centered loss functions (halos). Technical report, Contextual AI.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive NLP. *CoRR*, abs/2212.14024.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven Chu-Hong Hoi. 2022. Coderl: Mastering code generation through pretrained models and deep reinforcement learning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. 2023. RLAIF: scaling reinforcement learning from human feedback with AI feedback. *CoRR*, abs/2309.00267.

Haoran Li, Yiran Liu, Xingxing Zhang, Wei Lu, and Furu Wei. 2023. Tuna: Instruction tuning using feedback from large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 15146–15163. Association for Computational Linguistics.

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. *arXiv preprint arXiv:1607.06275*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Jiacheng Liu, Ramakanth Pasunuru, Hannaneh Hajishirzi, Yejin Choi, and Asli Celikyilmaz. 2023a. Crystal: Introspective reasoners reinforced with self-feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 11557–11572. Association for Computational Linguistics.

Jiate Liu, Yiqin Zhu, Kaiwen Xiao, Qiang Fu, Xiao Han, Wei Yang, and Deheng Ye. 2023b. RLTF: reinforcement learning from unit test feedback. *CoRR*, abs/2307.04349.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models. *CoRR*, abs/2305.14283.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *CoRR*, abs/2303.17651.

Fengran Mo, Kelong Mao, Yutao Zhu, Yihong Wu, Kaiyu Huang, and Jian-Yun Nie. 2023. Convgqr: Generative query reformulation for conversational search. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 4998–5012. Association for Computational Linguistics.

Deepak Nathani, David Wang, Liangming Pan, and William Yang Wang. 2023. MAF: multi-aspect feedback for improving reasoning in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6591–6616. Association for Computational Linguistics.

Rodrigo Frassetto Nogueira and Kyunghyun Cho. 2017. Task-oriented query reformulation with reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 574–583. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, et al. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Jing-Cheng Pang, Pengyuan Wang, Kaiyuan Li, Xiong-Hui Chen, Jiacheng Xu, Zongzhang Zhang, and Yang Yu. 2023. Language model self-improvement by reinforcement learning contemplation. *CoRR*, abs/2305.14483.

Wenjun Peng, Guiyang Li, Yue Jiang, Zilong Wang, Dan Ou, Xiaoyi Zeng, Derong Xu, Tong Xu, and Enhong Chen. 2023. Large language model based long-tail query rewriting in taobao search. *CoRR*, abs/2311.03758.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5687–5711. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *CoRR*, abs/2305.18290.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2023. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. 2016. High-dimensional continuous control using generalized advantage estimation. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Tianyi Zhou, and Daxin Jiang. 2023. Large language models are strong zero-shot retriever. *CoRR*, abs/2304.14233.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Amos Tversky and Daniel Kahneman. 1992. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5:297–323.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry W. Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc V. Le, and Thang Luong. 2023. Freshllms: Refreshing large language models with search engine augmentation. *CoRR*, abs/2310.03214.

Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, et al. 2024. Secrets of rlhf in large language models part ii: Reward modeling. *arXiv preprint arXiv:2401.06080*.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 9414–9423. Association for Computational Linguistics.

Zeqiu Wu, Yi Luan, Hannah Rashkin, David Reitter, Hannaneh Hajishirzi, Mari Ostendorf, and Gaurav Singh Tomar. 2022. CONQRR: conversational query rewriting for retrieval with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 10000–10014. Association for Computational Linguistics.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. 2023. C-pack: Packaged resources to advance general chinese embedding. *CoRR*, abs/2309.07597.

Weiwen Xu, Deng Cai, Zhisong Zhang, Wai Lam, and Shuming Shi. 2023. Reasons to reject? aligning language models with judgments. *arXiv preprint arXiv:2312.14591*.

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

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 10222–10240. Association for Computational Linguistics.

Fanghua Ye, Meng Fang, Shenghui Li, and Emine Yilmaz. 2023. Enhancing conversational search: Large language model-aided informative query rewriting. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5985–6006. Association for Computational Linguistics.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. RRHF: rank responses to align language models with human feedback without tears. *CoRR*, abs/2304.05302.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. 2024. A comprehensive study of knowledge editing for large language models. *CoRR*, abs/2401.01286.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the AI ocean: A survey on hallucination in large language models. *CoRR*, abs/2309.01219.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, et al. 2023. Secrets of rlhf in large language models part i: Ppo. *arXiv preprint arXiv:2307.04964*.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *CoRR*, abs/2308.07107.

A Appendix

A.1 Feedback Training Loss

A.1.1 DPO Loss

$$\mathcal{L}_{dpo} = -\mathbb{E}_{(q,q_g^{'},q_b^{'})\sim T_f}[\log\sigma \\ (\beta\log\frac{\mathcal{M}_\theta(q_g^{'}|q)}{\mathcal{M}_{\text{ref}}(q_g^{'}|q)} - \beta\log\frac{\mathcal{M}_\theta(q_b^{'}|q)}{\mathcal{M}_{\text{ref}}(q_b^{'}|q)})], \quad (5)$$

where β is the temperature parameter for DPO, $\mathcal{M}_\theta$ is the rewrite model to be updated, and $\mathcal{M}_{\text{ref}}$ is the fixed model during the training phase.

A.1.2 KTO Loss

The KTO (Kawin et al., 2023) (Kahneman-Tversky Optimization) method is based on *prospect theory* (Tversky and Kahneman, 1992), which tells how human decides according to uncertain outcomes. The theory is proposed by the economists Kahneman & Tversky. Compared to DPO, the training based on KTO only needs the signal that whether a rewrite $q^{'}$ is good or not, formulated as $(q, q^{'}; \rho), \rho \in$ [good, bad]. And the $\mathcal{L}_{kto}$ is computed as follows:

$$\begin{aligned}\mathcal{L}_{kto} &= \mathbb{E}_{(q,q^{'};\rho)\sim T_f}[w(q^{'})(1-\hat{h}(q,q^{'};\rho))],\\ g(q,q^{'};\rho) &= \beta\log\frac{\mathcal{M}_\theta(q^{'}|q)}{\mathcal{M}_{\text{ref}}(q^{'}|q)} - \\ &\quad \mathbb{E}_{q^{'}\sim T_f}[\beta\text{KL}(\mathcal{M}_\theta||\mathcal{M}_{\text{ref}})],\\ h(q,q^{'};\rho) &= \begin{cases}\sigma(g(q,q^{'};\rho)) & \text{if } \rho \text{ is good}\\ \sigma(-g(q,q^{'};\rho)) & \text{if } \rho \text{ is bad}\end{cases},\\ w(q^{'}) &= \begin{cases}\lambda_{good} & \text{if } \rho \text{ is good}\\ \lambda_{bad} & \text{if } \rho \text{ is bad}\end{cases}. \end{aligned} \quad (6)$$

The default values for λ_{good} and λ_{bad} are set to 1. When there is an imbalance between the number of good and bad samples, specific values are determined using the following formula:

$$\frac{\lambda_{\text{good}} n_{\text{good}}}{\lambda_{\text{bad}} n_{\text{bad}}} \in [1, \frac{4}{3}] \quad (7)$$

A.1.3 PPO Loss

When implementing PPO training, we indicate the action a_t at step t as generating the next token $\hat{q}_t^{'}$, while the current state $s_t = (q, \hat{q}_{<t}^{'})$ is composed of the original query and generated rewrite tokens. Here we directly use the ranking score as a reward, and by adding a KL-divergence regularization (Ramamurthy et al., 2023; Carpineto and Romano, 2012), the reward is computed as follow:

$$R(s_t, a_t) = S_{\text{reranker}}(q^{'}|q) - \beta_{\text{KL}}\text{KL}(\mathcal{M}_\theta||\mathcal{M}_{\text{ref}}) \quad (8)$$

and then with a value network V_ϕ initialized from $\mathcal{M}_\theta$, the advantages function follows GAE (Schulman et al., 2016) can be formulated as:

$$\begin{aligned}\delta_t &= R(s_t, a_t) + V_\phi(s_t+1) - V_\phi(s_t),\\ A(s_t, a_t) &= \sum\nolimits_{t^{'}=0}^{\infty}\lambda^{t^{'}}\delta_{t+t^{'}}\end{aligned} \quad (9)$$

and the final objective function is composed of value loss and policy loss (Zheng et al., 2023).

$$\begin{aligned}&\mathcal{L}_\theta = \mathbb{E}_{(s_t,a_t)\sim\mathcal{M}_\theta}[min(\frac{\mathcal{M}_\theta(s_t,a_t)}{\mathcal{M}_{\text{ref}}(s_t,a_t)}A(s_t,a_t),\\ &\text{clip}(\frac{\mathcal{M}_\theta(s_t,a_t)}{\mathcal{M}_{\text{ref}}(s_t,a_t)}, 1-\epsilon, 1+\epsilon)A(s_t,a_t))],\\ &\mathcal{L}_\phi = \mathbb{E}_{(s_t,a_t)\sim\mathcal{M}_\theta}(V_\phi(s_t) - R_t)^2,\\ &\mathcal{L}_{\text{ppo}} = \mathcal{L}_\theta + \mathcal{L}_\phi\end{aligned} \quad (10)$$

A.2 Training Details

Language	Source	Num
ZH	baike	6552
	webqa	16486
	sougouqa	9488
	squadzen	6294
	balle	9601
	coig	15080
EN	hotpotqa	12471
	triviaqa	28083
	nq	19445

Table 6: Data Source of the Training Instances for Open Domain QA.

A.2.1 Implementation

All model training is completed on a single machine with 4×A100 GPUs. And the training prompt for the rewrite is listed in Table 15.

SFT We train the rewrite model with 2 epochs and set the learning rate to 5e-5.

Method	EN								ZH			
	FreshQA		**NQ**		**TriviaQA**		**HotpotQA**		**FreshQA**		**WebQA**	
	Prec@10	MRR	Prec@10	MRR	Prec@10	MRR	Prec@10	MRR	Prec@10	MRR	Prec@10	MRR
OQR	26.34	38.43	30.41	45.59	48.66	61.70	15.49	29.25	15.38	24.32	74.97	84.48
SUBSTITUTE-Raw												
LLM-Rewrite	24.31	35.18	27.27	41.74	46.35	59.89	13.44	25.86	15.64	22.00	73.86	83.87
Query2Doc	24.42	35.95	26.05	37.82	48.71	59.24	14.96	25.54	14.98	24.23	**75.77**	**85.80**
SFT$_{(T_{\text{sft}})}$	24.43	36.13	28.75	43.81	45.51	59.69	14.48	28.27	15.27	24.94	70.83	80.09
SFT$_{(T_{\text{all}})}$	24.13	34.69	28.45	43.08	45.67	59.68	14.73	28.78	14.25	23.40	72.10	82.73
RaFe$_{(PPO)}$	25.73	37.23	29.44	44.16	46.59	60.45	15.10	29.32	15.44	**26.36**	72.47	84.64
RaFe$_{(DPO)}$	26.42	28.75	30.18	45.34	48.20	61.91	**16.42**	**31.14**	**16.20**	25.01	74.47	83.87
RaFe$_{(KTO)}$	**26.59**	**39.19**	**30.78**	**45.92**	**48.86**	**62.09**	15.75	29.93	15.65	25.97	73.47	84.60
EXPAND-Raw												
LLM-Rewrite	26.28	38.46	30.88	44.42	48.96	61.80	16.25	28.72	16.24	24.79	76.27	86.09
Query2Doc	26.76	38.48	29.99	44.77	48.78	60.44	17.15	30.18	17.51	25.80	**77.93**	89.05
SFT$_{(T_{\text{sft}})}$	25.78	39.07	30.40	44.38	48.62	61.93	17.04	30.51	17.02	26.64	69.80	88.68
SFT$_{(T_{\text{all}})}$	25.48	39.14	30.59	44.44	48.86	61.89	17.24	**30.56**	16.62	25.75	70.35	88.86
RaFe$_{(PPO)}$	27.12	39.25	30.46	45.42	48.67	61.73	17.24	30.41	**17.82**	26.41	76.21	**89.12**
RaFe$_{(DPO)}$	26.98	38.85	31.18	45.45	49.63	61.96	17.38	30.43	16.20	25.01	74.42	89.05
RaFe$_{(KTO)}$	**27.80**	**39.56**	**31.22**	**45.73**	**49.82**	**62.02**	**17.67**	30.53	17.66	**26.86**	74.98	89.10

Table 7: The retrieval results of **SUBSTITUTE-Raw** and **EXPAND-Raw** settings.

Method	**FreshQA**		**NQ**	
	Raw	Ranked	Raw	Ranked
w/o retrieval	32.83	-	36.67	-
OQR	39.79	41.13	42.53	44.16
SUBSTITUTE				
LLM-Rewrite	35.24	36.75	40.24	40.27
Query2Doc	34.97	35.63	40.05	41.32
SFT$_{(T_{\text{sft}})}$	40.07	40.66	42.27	43.24
SFT$_{(T_{\text{all}})}$	38.92	40.01	42.34	43.80
RaFe$_{(PPO)}$	**41.15**	**42.13**	42.57	44.23
RaFe$_{(DPO)}$	38.18	39.73	42.82	44.84
RaFe$_{(KTO)}$	40.46	41.77	**43.78**	**44.90**
EXPAND				
LLM-Rewrite	37.24	39.14	43.40	44.43
Query2Doc	38.78	39.29	44.13	45.07
SFT$_{(T_{\text{sft}})}$	39.49	39.29	43.54	44.17
SFT$_{(T_{\text{all}})}$	39.91	41.68	43.89	44.21
RaFe$_{(PPO)}$	40.05	42.64	44.39	44.87
RaFe$_{(DPO)}$	40.41	42.37	44.49	45.34
RaFe$_{(KTO)}$	**40.74**	**43.79**	**44.56**	**45.64**

Table 8: The QA results on Qwen1.5-32b-chat.

PPO The PPO implementation is carried out according to the TRL repo[4](von Werra et al., 2020). In line with the empirical configurations in previous work (Zheng et al., 2023), we set the batch size to 32, and conduct the training for 1000 optimization steps, which is approximately equivalent to 1.067 epochs. The clip range parameter ϵ, and the coefficient β_{KL} for the KL divergence in Eq 8, are both set to 0.2 as defaulted.

DPO & KTO We conduct the offline training for 1 epoch on all the good-bad rewrite data, with a learning rate of 5e-6. We set the temperature parameter β to 0.1, following the default setting of the previous implementation[5]

A.2.2 Dataset Details

We list the sources and numbers of training instances in Table 6.

Initial Training Set of Rewrite Model For the open-domain QA task, we use qwen-max (Bai et al., 2023) to conduct the data production for both English and Chinese dataset.

The Construction of Translated FreshQA We first translate the entire set of 500 FreshQA test questions, and then manually review and filter each translation to identify those that were relatively more relevant to the Chinese internet. Ultimately, we obtained a set of 293 Chinese-translated FreshQA dataset.

A.3 Additional Experimental Results

A.3.1 The Retrieval Results

We report complete retrieval results of Prec@10 and MRR in this section. The results of SUBSTITUTE-Raw and EXPAND-Raw are shown in Table 7, while the results of SUBSTITUTE-Ranked and EXPAND-Ranked are in Table 9.

[4] https://github.com/huggingface/trl

[5] https://github.com/ContextualAI/HALOs

Method	EN								ZH			
	FreshQA		NQ		TriviaQA		HotpotQA		FreshQA		WebQA	
	Prec@10	MRR	Prec@10	MRR	Prec@10	MRR	Prec@10	MRR	Prec@10	MRR	Prec@10	MRR
OQR	26.34	43.92	30.41	49.06	48.66	64.28	15.49	31.03	15.38	26.67	74.97	87.92
SUBSTITUTE-Ranked												
LLM-Rewrite	24.31	40.31	27.27	47.28	46.35	62.79	13.44	27.20	15.64	24.53	73.86	85.23
Query2Doc	24.42	39.53	26.05	42.55	48.71	61.18	14.96	27.24	14.98	25.29	**75.77**	**88.23**
SFT$_{(T_{\text{sft}})}$	24.43	42.28	28.75	48.06	45.51	62.86	14.48	30.58	15.27	**26.94**	70.83	80.71
SFT$_{(T_{\text{all}})}$	24.13	41.17	28.45	47.87	45.67	62.94	14.73	30.19	14.25	21.39	72.10	80.26
RaFe$_{(PPO)}$	25.73	43.14	29.44	48.52	46.59	63.52	15.10	30.60	15.44	26.15	72.47	86.77
RaFe$_{(DPO)}$	24.42	43.19	30.18	48.97	48.20	64.52	**16.42**	31.52	**16.20**	25.46	74.47	85.54
RaFe$_{(KTO)}$	**26.59**	43.08	**30.78**	**49.48**	**48.86**	**65.17**	15.75	**32.28**	15.65	26.50	73.47	85.89
EXPAND-Ranked												
LLM-Rewrite	29.45	42.14	32.42	48.97	52.14	64.78	18.32	32.06	18.02	26.32	77.23	87.12
Query2Doc	30.50	44.51	32.73	49.21	52.25	64.88	19.24	33.66	18.18	26.64	**79.81**	**88.81**
SFT$_{(T_{\text{sft}})}$	30.52	44.62	34.02	49.39	52.55	66.06	19.29	33.03	18.87	27.55	77.02	87.86
SFT$_{(T_{\text{all}})}$	23.71	41.31	34.36	49.64	52.65	66.14	19.34	33.22	18.16	27.79	77.21	87.90
RaFe$_{(PPO)}$	30.28	44.29	35.10	50.37	52.63	65.86	19.66	**33.92**	18.56	**29.17**	79.26	88.47
RaFe$_{(DPO)}$	30.62	44.54	**35.22**	50.10	**53.55**	66.05	19.77	33.81	16.19	25.46	78.18	88.28
RaFe$_{(KTO)}$	**31.14**	**45.24**	35.18	**50.54**	53.09	**66.46**	**19.89**	33.75	**18.90**	27.43	77.84	88.09

Table 9: The retrieval results of **SUBSTITUTE-Ranked** and **EXPAND-Ranked** settings.

Method	FreshQA				NQ			
	Raw		Ranked		Raw		Ranked	
	QA	Prec@5	QA	Prec@5	QA	Prec@5	QA	Prec@5
w/o retrieval	**41.70**	-	-	-	43.74	-	-	-
OQR	34.08	11.86	34.98	13.48	50.13	24.14	50.91	28.27
SUBSTITUTE								
SFT$_{(T_{\text{all}})}$	32.44	11.28	34.28	12.90	48.24	22.84	50.17	27.43
Prec Feedback	34.31	11.92	**35.53**	13.32	48.12	23.54	49.31	27.28
LLM Feedback	33.64	11.52	35.38	13.28	50.16	23.38	50.47	27.98
RaFe$_{(KTO)}$	**34.36**	**12.44**	35.48	**13.60**	**50.33**	23.86	50.87	**28.36**
EXPAND								
SFT$_{(T_{\text{all}})}$	33.31	11.00	35.58	13.44	49.66	23.32	50.33	27.27
Prec Feedback	33.83	11.76	36.20	14.68	49.42	23.12	50.94	28.17
LLM Feedback	33.64	11.88	**36.50**	14.44	50.47	23.76	51.24	28.12
RaFe$_{(KTO)}$	**34.31**	**12.20**	36.39	**14.92**	**50.78**	23.81	**51.49**	**28.62**

Table 10: Result conducted by dense retriever.

Method	FreshQA				NQ			
	Raw		Ranked		Raw		Ranked	
	QA	Prec@5	QA	Prec@5	QA	Prec@5	QA	Prec@5
OQR	61.87	27.48	62.56	30.88	51.36	32.35	51.50	35.68
SUBSTITUTE								
SFT$_{\text{llama}}$	32.44	11.28	34.28	12.90	48.24	22.84	50.17	27.43
SFT$_{\text{qwen}}$	60.53	25.72	60.69	28.42	49.86	30.08	50.99	34.01
RaFe$_{\text{llama}}$	61.79	27.93	62.37	30.15	51.73	32.14	51.77	35.27
RaFe$_{\text{qwen}}$	62.12	28.12	62.71	31.00	51.61	32.71	51.97	35.89
EXPAND								
SFT$_{\text{llama}}$	60.58	26.04	62.72	32.12	50.41	31.33	51.57	37.13
SFT$_{\text{qwen}}$	62.01	26.76	63.16	31.56	50.13	30.63	51.75	37.44
RaFe$_{\text{llama}}$	62.43	**28.53**	63.77	33.45	51.63	31.43	52.45	37.73
RaFe$_{\text{qwen}}$	**62.65**	28.50	**64.85**	**33.72**	**52.48**	**32.58**	**52.86**	**38.37**

Table 11: Results with different backbones.

Comparing the results between SUBSTITUTE and EXPAND, it can be found that methods with lower retrieval results under the SUBSTITUTE setting tended to show greater improvement under EXPAND. However, the retrieval results for RaFe do not exhibit great improvement under the EXPAND-Raw setting. Further comparison between the QA results and retrieval metrics reveals that, generally, the improvement trends in retrieval results align with those in QA performance.

A.3.2 QA Results of Qwen-32b

To further demonstrate the results of our methods, we conduct experiments on different sizes of models. Specifically, we choose Qwen1.5-32b-chat for evaluation. The results are shown in Table 8. The results indicate that RaFe consistently outperforms across almost all settings. Moreover, it is observed that compared to Qwen-max for QA, the 32B model exhibits lower performances.

In the SUBSTITUTE-Raw setting of the NQ dataset, utilizing Qwen-max does not yield great results. However, a significant improvement can be observed with Qwen-32b. This may suggest that for some cases beyond the capability coverage of qwen-32b, query rewriting can benefit the retrieval augmentation. As models increase in size, their inherent capabilities may become sufficient to handle these cases effectively, negating the need for query rewriting.

A.3.3 Results with Dense Retrieval

To widely validate our method, we extend the experiment on dense retrievers. We conduct additional experiments with the retriever (Contriever) and the corpus (Wikipedia) used in Self-Rag (Asai et al., 2023) to rebuild the ranking feedback for training the rewrite model. The experimental results are presented in Table 10.

From the results, it can be found that most

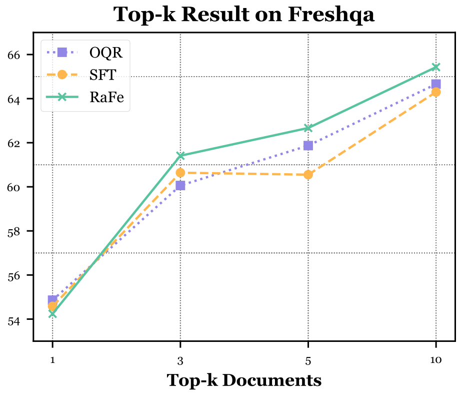


Figure 6: The results under SUBSTITUTE setting on FreshQA with a different number of documents.

rewrite baselines do not perform as well as the original query under substitution, which might be attributed to the fact that dense retrievers use embeddings for retrieval, and the rewritten query's representation certainly varies from that of the original query. While our method's rewrites still outperform in most settings.

Moreover, retrieval with a fixed corpus does not yield better results than web search retrieval, especially for changing questions (i.e. freshqa, the rag results are lower than w/o retrieval), where it tends to introduce more noise or outdating knowledge.

A.3.4 Results with Different Backbone

For a more comprehensive evaluation, we implement RaFe on llama3-8b-base. The experimental results are presented in Table 11. It can be easily observed that our method also works on Llama, surpassing other rewrite baselines.

A.3.5 Top-k Documents Results

Additionally, we explore the performance of our proposed method when concatenating a different number of documents. We experiment with the Chinese version of the FreshQA. The results presented in Figure 6 reveal that when solely the first document is utilized, the retrieval using the original query yields the best results. As the number of concatenated documents increases, RaFe consistently outperforms both SFT and the original query results.

A.4 Additional Analysis

A.4.1 The Relatively Weak Performance

From the results, it can be observed that there are only marginal improvements in some datasets, especially in SUBSTITUTE-Raw setting. Taking the

Case Set	Model	OQR	RaFe
Good	Qwen-max	59.10	59.30
	Qwen-32b	60.12	59.98
Bad	Qwen-max	5.37	5.73
	Qwen-32b	11.21	11.69

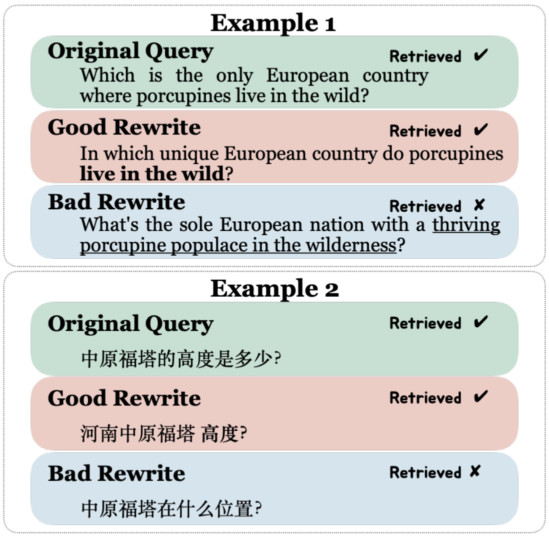
Table 12: The **Prec@5** results of NQ datasets answered by different sizes of Models under SUBSITUTE-Raw setting. **Good** indicates the cases correctly answered by both OQR and RaFe, while **Bad** refers to both incorrect.

NQ dataset as an example, we attempt to investigate the difference between. The NQ dataset is quite hard, so for challenging cases, the minor reformulation of key phrases could cause the wrong retrieval. For instance, comparing Original Query: *"what is the cross on a letter t called?"* and RaFe Rewrite: *"What do you call the cross-like symbol on a letter 't'?"*, it can be found in the original query explicitly using "cross on a letter t" to a specific term related to typography. The rewritten query adds complexity and potential vagueness with a "cross-like symbol", which may mislead search engines towards broader symbol recognition or confuse with other types of crosses, thereby reducing the precision of the search results.

Additionally, the results on smaller models revealed that RaFe could achieve noteworthy improvements even in SUBSTITUTE-Raw results. Thus, we obtain the cases answered both correctly and wrong by different size models. As shown in Table 12, the average prec@5 on 'good' cases is comparable between models of different sizes. However, in 'bad' cases, smaller models exhibit higher average precision. In contrast, when comparing the the results between Qwen-max and Qwen-32b, the improvements from RaFe diminish. This suggests that the benefits RaFe brings in simple cases are reduced as the model's parameter increases. Meanwhile, the deviations in more challenging cases are retained, which could lead to less impressive results. This further implies that query rewriting for RAG might be better suited for the EXPAND setting, to broaden the scope of the query to increase the chances of retrieving relevant information.

A.4.2 Good-Bad Pairs Cases

In this section, we delve deeper into how rerankers take effect by presenting case studies. We investigate cases of how rerankers distinguish between


Example 1

Original Query Retrieved ✔
Which is the only European country where porcupines live in the wild?

Good Rewrite Retrieved ✔
In which unique European country do porcupines **live in the wild?**

Bad Rewrite Retrieved ✘
What's the sole European nation with a thriving porcupine populace in the wilderness?

Example 2

Original Query Retrieved ✔
中原福塔的高度是多少?

Good Rewrite Retrieved ✔
河南中原福塔 高度?

Bad Rewrite Retrieved ✘
中原福塔在什么位置?

Figure 7: Two examples of good-bad rewrite pairs, each containing an original query, the good rewrite and bad rewrite. The **"Retrieved"** sign indicates whether the top 5 documents contain the answer or not.

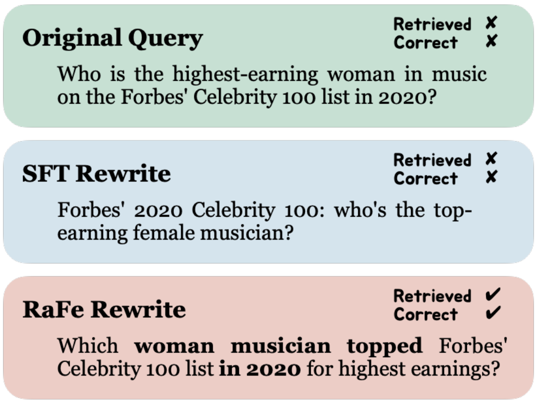
good-bad pairs. Figure 7 provides two examples.

In the first example, the original query pertains to the only European country where wild porcupines reside. The good rewrite simplifies to a more direct question: "In which unique European country do porcupines live in the wild?" This rewrite is clear and precise. In contrast, the bad rewrite, "What's the sole European nation with a thriving porcupine populace in the wilderness?" Although conveying similar information, it appears excessively verbose and unnecessarily complex in its wording, resulting in failure in retrieval.

The second example's original query asks about the height of the Zhongyuan Pagoda in Henan Province, China. The good rewrite poses the same question in a more concise manner: "Height of Zhongyuan Pagoda in Henan?" This succinct rewrite may be better suited for rapid information retrieval. The bad rewrite, on the other hand, is: "Where is the Zhongyuan Pagoda located?" It fails to correctly rephrase the original question, as it shifts the focus from "height" to "location", causing a deviation from the original query's intent. These cases demonstrate that the reranker's scoring of retrieved documents can effectively differentiate between good and bad rewrites.

A.4.3 Additional Case for Better Format Rewriting

We provide one more case in this section. The original question used the phrase "*woman in music*" to inquire about the highest-earning female musician in Forbes' Celebrity 100 list in 2020, which may not have been as intuitive for search engines, resulting in a failure to retrieve the documents. While the RaFe rewrite directly refines "*woman in music*" into "*woman musician*", rephrasing the question with vocabulary more suitable for retrieval purposes.


Original Query Retrieved ✘ Correct ✘
Who is the highest-earning woman in music on the Forbes' Celebrity 100 list in 2020?

SFT Rewrite Retrieved ✘ Correct ✘
Forbes' 2020 Celebrity 100: who's the top-earning female musician?

RaFe Rewrite Retrieved ✔ Correct ✔
Which **woman musician topped** Forbes' Celebrity 100 list **in 2020** for highest earnings?

Figure 8: An example includes the original query and rewrite from SFT and RaFe. The label **"Retrieved"** denotes whether the answer is present within the top 5 retrieved documents, and **"Correct"** denotes whether the prediction is correct.

In contrast, the rewrite from SFT also conveys a clearer expression of "*female musician*", but its format more closely resembles a headline or newspaper title, which may not be as suitable for a search query as a direct interrogative format. Additionally, it does not clearly express that the search is specifically for the year.

A.4.4 Different Performances for Feedback Training

In our experiments, it can be observed that KTO > DPO > PPO across most settings, which we believe may be related to the number of training instances.

When conducting the 'good bad rewrites', we encounter instances where all rewrites were either labeled as bad or good, leading to samples that can not be paired, and thus reducing the training data for DPO. While the KTO training process does not require paired preference data, it only needs the preference labels (i.e. good or bad) for training. So the actual amounts of training instances for KTO is larger. Compared to PPO's training process, where each iteration only generates a single rewrite for feedback optimization, the number of samples used to optimize the rewriting model is lower.

Moreover, we have found that online PPO train-

Method	Epoch Nums					
	0.1	0.3	0.5	0.7	0.9	1
PPO	0.5056	0.1213	0.0449	0.0083	0.0497	0.0093
DPO	0.6023	0.3856	0.2891	0.1232	0.0729	0.0657
KTO	0.5005	0.3214	0.2416	0.2142	0.1819	0.1787

Table 13: The training loss with different feedback training methods.

Method	Rewrite Nums					
	0	1	2	3	4	5
valid num	9.09	15.42	20.26	24.38	27.95	30.89
total num	9.16	18.31	27.37	36.53	45.68	54.83
valid ratio(%)	99.17	84.19	74.03	66.73	61.19	56.34

Table 14: The numbers of valid documents and total documents as the rewrite number grows.

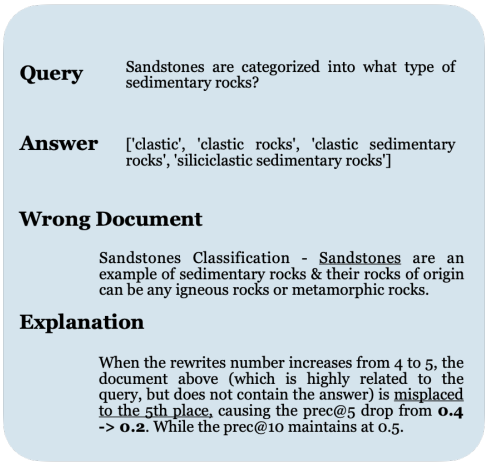
ing is more unstable compared to offline training, with a pattern collapse phenomenon (the model can not generate coherent text) occurring between approximately 0.7 to 0.8 epochs. However, from additional experiments we conducted, offline models did not collapse until approximately 1.5-2 epochs.

We further present the loss for different training methods within one epoch in Table 13. PPO's loss demonstrates noticeable instability after 0.7 epochs, indicating its unstable training pattern, which has been detailedly investigated by Zheng et al. (2023).

A.4.5 Why the Prec@5 Drops as Rewrite Number Grows?

It can be observed in Figure 5 that the prec@5 drops when the number increases from 4 to 5. In this section, we further analyze this phenomenon. We count the number of all documents retrieved and the valid (non-repetitive) documents with different numbers of rewrites, presented in Table 14. It can be found that, although the number of rewrites increases, the quantity of duplicate documents also grows, which means that performance may plateau at a peak value. As seen in the table, when the quantity of rewrites grows, the increase in valid documents slows down, and the proportion of effective documents decreases, leading to a plateau in recall improvement.

Regarding the decrease in Prec@5, the main reason is that our experiments are mainly conducted on the top 5 documents. As the reranker's performance is limited, within the scope of the top 5, the reranker may sometimes misplace the documents that, while related to the question, do not contain the correct answer, to a higher place. We provide a case in Figure 9. However, when comparing prec@10 among the different numbers of rewrites, the overall trend still shows an increase, although the magnitude of this increase diminishes. Therefore, we believe that in practical applications, having 2-3 rewrites might be more appropriate.



Query Sandstones are categorized into what type of sedimentary rocks?

Answer ['clastic', 'clastic rocks', 'clastic sedimentary rocks', 'siliciclastic sedimentary rocks']

Wrong Document

Sandstones Classification - Sandstones are an example of sedimentary rocks & their rocks of origin can be any igneous rocks or metamorphic rocks.

Explanation

When the rewrites number increases from 4 to 5, the document above (which is highly related to the query, but does not contain the answer) is misplaced to the 5th place, causing the prec@5 drop from **0.4 -> 0.2**. While the prec@10 maintains at 0.5.

Figure 9: A case for the misplacing caused by reranker.

A.5 Prompts

In this section, we list the prompt we used in this paper. The instruction prompt for the rewrite model is shown in Table 15, and the prompt for evaluation is in Table 16. The few-shot prompts used for Query2Doc are derived from Wang et al. (2023).

Prompt
Instruction: output the rewrite of input query
Query: [ORIGINAL QUERY]
Output: [TARGET]

Table 15: The instruction prompt for rewriting models, both training and inference.

Prompt
USER The following information may help answering questions: <TOP-K DOCUMENTS>
LLMs Sure, I have noted the information above. Is there anything I can assist you with or any questions I can help answer?
USER <QUESTION>

Table 16: The evaluation prompt when employing Qwen-max for open-domain QA.

Prompt
Please provide a rewrite to express the same query based on the given query. Here are some example Query: what state is this zip code 85282? Output: Welcome to TEMPE, AZ 85282. 85282 is a rural zip code in Tempe, Arizona. The population is primarily white, and mostly single. At $200,200 the average home value here is a bit higher than average for the Phoenix-Mesa-Scottsdale metro area, so this probably isn't the place to look for housing bargains.5282 Zip code is located in the Mountain time zone at 33 degrees latitude (Fun Fact: this is the same latitude as Damascus, Syria!) and -112 degrees longitude. Query: why is gibbs model of reflection good Output: In this reflection, I am going to use Gibbs (1988) Reflective Cycle. This model is a recognised framework for my reflection. Gibbs (1988) consists of six stages to complete one cycle which is able to improve my nursing practice continuously and learning from the experience for better practice in the future.n conclusion of my reflective assignment, I mention the model that I chose, Gibbs (1988) Reflective Cycle as my framework of my reflective. I state the reasons why I am choosing the model as well as some discussion on the important of doing reflection in nursing practice. Query: what does a thousand pardons means Output: Oh, that's all right, that's all right, give us a rest; never mind about the direction, hang the direction - I beg pardon, I beg a thousand pardons, I am not well to-day; pay no attention when I soliloquize, it is an old habit, an old, bad habit, and hard to get rid of when one's digestion is all disordered with eating food that was raised forever and ever before he was born; good land! a man can't keep his functions regular on spring chickens thirteen hundred years old. Query: what is a macro warning Output: Macro virus warning appears when no macros exist in the file in Word. When you open a Microsoft Word 2002 document or template, you may receive the following macro virus warning, even though the document or template does not contain macros: C:\<path>\<file name>contains macros. Macros may contain viruses. Query: {} Output:

Table 17: The few-shot prompt for Query2Doc when generating pseudo documents from LLMs adopted from (Wang et al., 2023).