DynQR: Dynamic Uncertainty-Guided Query Rewriting for Effective Retrieval-Augmented Generation

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have shown impressive performance across numerous tasks but often produce hallucinated or inaccurate responses, reducing their reliability. Retrieval-Augmented Generation (RAG) mitigates this issue by incorporating external knowledge into the generation process, yet the effectiveness of the retrieval depends heavily on the search queries and query rewriting techniques are typically adopted to improve the retrieval quality. However, current rewriting methods rely on indirect feedback or costly direct feedback with annotated labels, limiting their practicality and effectiveness. We introduce DynQR, an annotation-free query rewriting framework that uses uncertainty from the reader LLM to provide direct feedback, effectively bridging the gap between the input queries and the needed knowledge in retrieval. DynQR follows a three-stage approach to train a rewriter that reduces uncertainty in the reader's responses. Additionally, DynQR employs an active rewriting mechanism and post-verification process to minimize unnecessary rewriting and avoid potential noise. Our experiments on five datasets across three QA tasks show that DynQR consistently outperforms existing baselines.

1 Introduction

Large Language Models (LLMs) (Taylor et al., 2022; Chowdhery et al., 2022; Zhao et al., 2023) have recently demonstrated exceptional performance across a wide range of downstream tasks (Xia et al., 2024; Yamauchi et al., 2023; Imani et al., 2023; Lewkowycz et al., 2022). Despite these advancements, LLMs frequently produce responses containing hallucinated facts or inaccurate information (Ji et al., 2023; Shuster et al., 2021; Zhang et al., 2023), which undermines their overall reliability. To address this issue, researchers have leveraged Retrieval-Augmented Generation (RAG) to integrate external knowledge into the generation

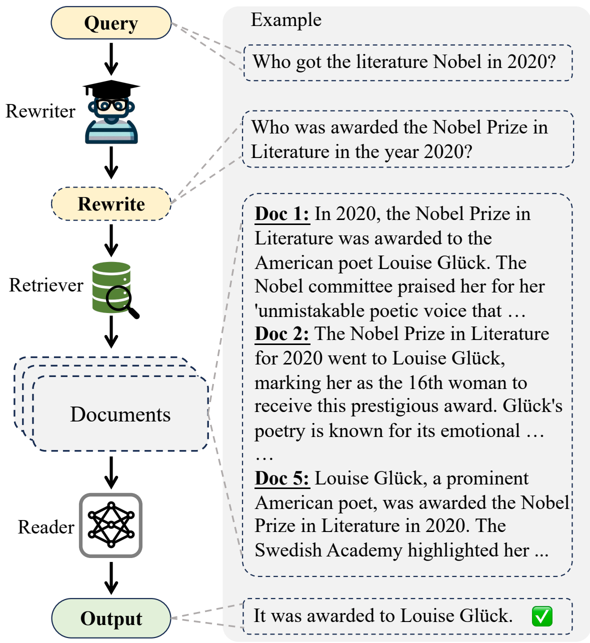


Figure 1: Illustration of Query Rewriting for RAG.

process (Ram et al., 2023; Shi et al., 2023; Rashkin et al., 2021; Gao et al., 2022; Bohnet et al., 2022; Menick et al., 2022). In a typical RAG system, a user's query is used to retrieve relevant documents from external sources, which are then combined with the model's internal knowledge to generate more accurate and informative responses. However, the effectiveness of this approach hinges on the quality of the retrieved documents, which in turn depends on the formulation of the initial user query. A major challenge in RAG systems arises from the ambiguity and vagueness of user queries. Users often submit incomplete or overly broad queries, expecting the system to infer their intent. This defect in query formulation can lead to suboptimal generation responses, as the system may fail to retrieve the most relevant information.

To mitigate this issue, query rewriting has emerged as a promising technique to improve the retrieval process by refining the original query. Existing studies (Ye et al., 2023; Wang et al., 2023; Shen et al., 2023) have leveraged the strong reasoning capabilities of LLMs to expand or rewrite queries effectively. To further reduce the inference cost associated with these rewriters, researchers have employed feedback training (Zheng et al., 2023; Wang et al., 2024; Rafailov et al., 2024; Yuan et al., 2023) to enhance smaller query rewriting models, utilizing both supervised and unsupervised methods. For supervised approaches, RRR (Ma et al., 2023) uses the feedback regarding whether the rewritten query leads the reader LLM to generate the correct answer as a reward signal to train the rewriter. Similarly, RETPO (Yoon et al., 2024) uses the signal of whether the documents retrieved by the rewritten query contain the correct answer as the reward to guide the training of the rewriter. To reduce the dependency on labeled data, the unsupervised method RaFe (Mao et al., 2024) proposes utilizing the relevance between documents retrieved by the rewritten query and the original query as a reward for training the rewriter model.

Despite their superior performance, these methods suffer from several limitations. Supervised approaches rely on manually labeled data, which is costly and time-consuming to obtain at scale. Unsupervised methods, while more scalable, often rely on indirect feedback, such as the relevance of retrieved documents, which may not align well with the actual needs of the reader LLM. For instance, while RaFe might generate queries that retrieve documents more relevant to the original query, these documents do not necessarily provide the information the reader LLM truly requires. As a result, such indirect feedback can sometimes be misleading and lead to suboptimal results. Moreover, most existing approaches apply query rewriting universally, assuming that all queries require rewriting. However, we argue that not every query benefits from rewriting, as it may introduce additional inference costs. Therefore, selectively rewriting only those queries that would substantially benefit from it could strike a better balance between performance and computational efficiency.

Recent studies have highlighted a strong correlation between the uncertainty of large language models and their correctness across various tasks (Kadavath et al., 2022; Jiang et al., 2021; Hua et al., 2023; Plaut et al., 2024; Fadeeva et al., 2023; Weller et al., 2023). As an unsupervised metric, uncertainty is derived directly from the model itself, reflecting its own assessment of the given input. Motivated by this insight, we propose **DynQR**, an unsupervised query rewriting method that leverages direct feedback from the reader LLM without requiring hand-crafted labels. Specifically, our approach consists of three stages: Supervised Distillation, Uncertainty-Aware Sampling, and Preference Alignment. In Supervised Distillation, we construct a query rewriting dataset to train the rewriter model, thereby equipping it with a basic query rewriting capability. In Uncertainty-Aware Sampling, we utilize the trained rewriter model to generate new queries and record the uncertainty of the reader LLM based on the documents retrieved by these queries. In Preference Alignment, we train the rewriter to favor generating queries that result in the reader LLM producing answers with lower uncertainty. The resulting rewriter model can effectively generate queries that retrieve high-quality documents, enabling the reader LLM to produce more accurate answers with lower uncertainty. During inference, we introduce an active rewriting mechanism that selectively triggers query rewriting only when the LLM exhibits high uncertainty in its initial response. Additionally, we implement a post-verification step that compares the uncertainties of the answers generated from the original and rewritten queries, ensuring that the final response is based on the query that results in lower uncertainty.

To summarize, our contributions can be summarized as follows:

- We propose an unsupervised query rewriting method, DynQR, which directly leverages uncertainty-based feedback from the reader LLM, eliminating the need for labeled data from downstream tasks.
- DynQR introduces an active rewriting mechanism to minimize query costs and incorporates a post-verification mechanism to avoid potential noise from unnecessary query rewriting.
- We conduct extensive experiments on five datasets across three knowledge-intensive tasks, verifying the effectiveness of DynQR.

2 Methodology

2.1 Preliminary

In Retrieval Augmented Generation (RAG), given an original query q, a retriever is first used to retrieve a set of similar documents $\mathcal{D} =$

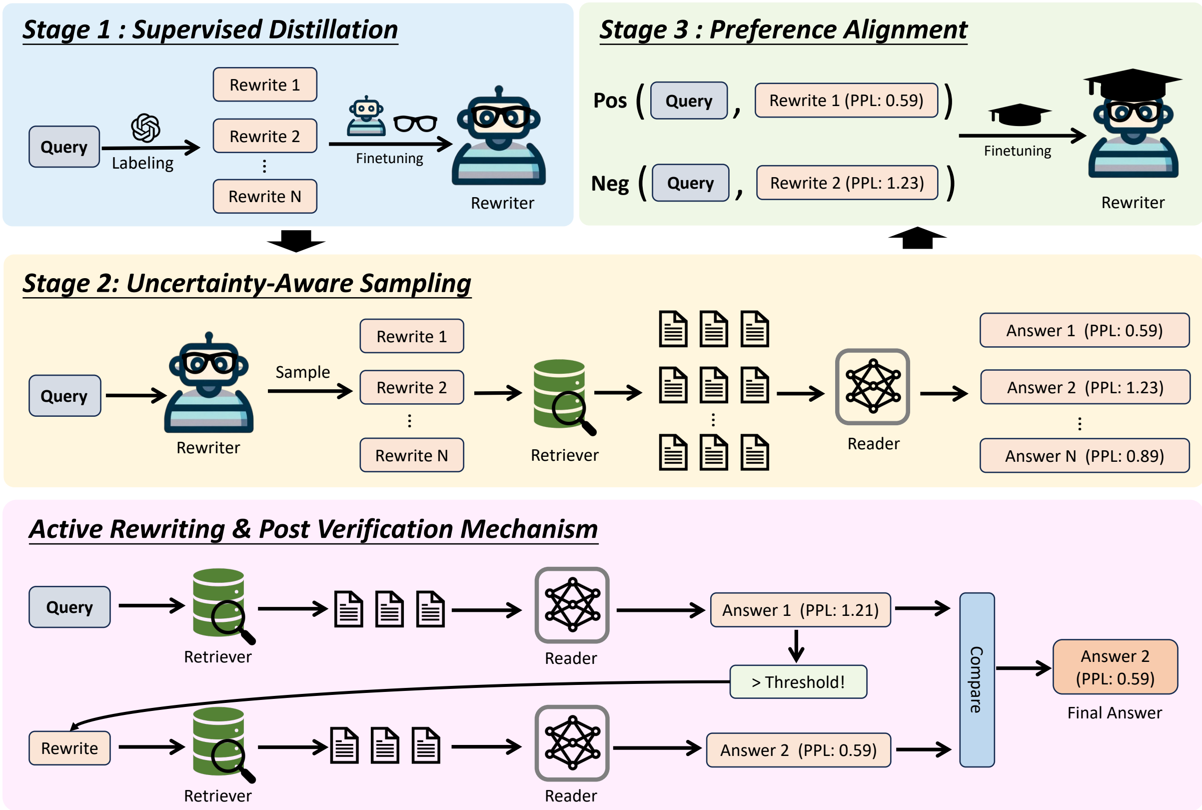


Figure 2: The illustration of DynQR. 1) Supervised Distillation: The rewriter learns basic rewriting skills. 2) Uncertainty-Aware Sampling: The rewriter generates multiple rewrites for each query, which are used to retrieve relevant documents. The uncertainty of the reader LLM's answers is recorded. 3) Preference Alignment: The rewriter is trained to generate queries that lead the reader LLM to produce answers with lower uncertainty. During inference, query rewriting is triggered only when the LLM exhibits high uncertainty in its initial response. The final answer is selected based on the query that results in lower uncertainty.

$\{d_0, d_1, \dots, d_m\}$. A reader LLM then answers the query based on these retrieved documents. The goal of query rewriting is to develop a better rewriter model M_θ, which rewrites the original query q into a refined query r:

$$r = M_\theta(q), \tag{1}$$

where r represents the rewritten query, which will be used to retrieve relevant documents for augmented generation.

2.2 DynQR Framework

As illustrated in Figure 2, DynQR consists of three stages: Supervised Distillation, Uncertainty-Aware Sampling, and Preference Alignment. In the Supervised Distillation stage, the rewriter is trained to develop basic query rewriting capabilities. During Uncertainty-Aware Sampling, the rewriter generates multiple rewrites for each query, and the reader LLM uses the retrieved documents to generate answers, with the uncertainty of each answer recorded. Finally, in the Preference Alignment stage, preference pairs are constructed by labeling rewrites that result in lower uncertainty as positive samples, and those with higher uncertainty as negative samples.

Supervised Distillation In the first stage, a large language model is used as a data labeler to rewrite queries in the training set, constructing a dataset for rewriter training. The rewriter model is then trained on this dataset to acquire the basic capability to generate effective rewrites for given queries.

Uncertainty-Aware Sampling Given a query, its rewrites $\mathcal{R} = \{r_1, r_2, ...r_n\}$, the corresponding document set will be retrieved using each rewrite:

$$D_i = Retrieve(r_i) \tag{2}$$

These retrieved documents D_i are then combined with the original query to generate an answer using the reader LLM. We employ an uncertainty estimator $U(\cdot)$ to evaluate the uncertainty of each generated response:

$$s_i = U(q, D_i, LLM), \tag{3}$$

where s_i represents the uncertainty of the generation using documents retrieved from rewrite r_i. The uncertainty score provides a quantitative measure of the model's confidence in its generated answer, reflecting how well the retrieved information aligns with the query's intent. Consequently, it serves as a direct indicator of the quality of the rewritten queries—where lower uncertainty generally indicates more relevant retrieval and a more effective rewriting process.

Preference Alignment For a given query and its set of rewrites $\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$, we enumerate all possible combinations $\langle r_i, r_j \rangle$, where the uncertainty score of r_i is lower than that of r_j. We then select the three combinations with the largest uncertainty differences between r_i and r_j. These pairs are used to construct preference triplets $\langle q, r_i, r_j \rangle$, which are utilized to train the rewriter model using Direct Preference Optimization (Rafailov et al., 2024).

Active Rewriting Existing query rewriting approaches often assume that rewriting should be applied universally to all queries. However, we argue this may not always be necessary. In many cases, the documents retrieved by the original query already contain sufficient information for the reader LLM to generate an accurate response. Additionally, applying query rewriting to every query introduces unnecessary inference costs for the RAG system. To address this, we propose an active rewriting mechanism. In our approach, the reader LLM first attempts to generate an answer using documents retrieved by the original query. If the uncertainty of the generated answer falls below a predefined threshold θ, indicating high confidence, the answer is directly used as the final response. If the uncertainty exceeds the threshold—indicating a higher potential for hallucination—the query rewriter is activated to refine the original query. The reader LLM then generates a revised answer using documents retrieved from this rewritten query.

Post Verification To ensure that query rewriting enhances the final response without introducing additional noise, we implement a post-verification process. Specifically, we compare the uncertainties of the answers generated using documents retrieved from both the original and rewritten queries. The answer with the lower uncertainty score is selected as the final output, ensuring that the response with higher confidence is used, while avoiding potential noise introduced by unsuccessful rewritings.

3 Experiment Setup

3.1 Datasets and Metrics

Datasets We conduct experiments on five datasets across three knowledge-intensive tasks: (1) **Open-domain QA,** including NQ dataset (Kwiatkowski et al., 2019), TriviaQA dataset (Joshi et al., 2017) and PopQA dataset (Mallen et al., 2022); (2) **Multi-hop QA**, including 2WikiMultiHopQA dataset (Ho et al., 2020). (3) **Ambiguous QA**, including ASQA dataset (Stelmakh et al., 2022).

Metrics We evaluate performance using two key metrics: Exact Match (EM) and F1 Score. A predicted answer is considered correct under the EM metric if its normalized form exactly matches any of the normalized versions of the reference answers in the answer list. The F1 score, on the other hand, measures the word-level overlap between the normalized predicted answer and the reference answers in the provided answer list.

3.2 Baselines

We compare our methods with the following baselines:

- **Direct**: Directly answer the question without retrieving any external documents.
- **OriQR**: Use the original query to retrieve documents and then answer the question.
- **LLMQR**: Use an LLM to rewrite the query, then retrieve relevant documents.
- **RRR** (Ma et al., 2023): Utilize the downstream task answers as supervision signals.
- **RETPO** (Yoon et al., 2024): Utilize the retrieval results as supervision signals.
- **RaFe** (Mao et al., 2024): Utilize the relevance results as supervision signals.

To ensure a fair comparison, we replace the reward signals in our framework with those used by these methods and evaluate their performance.

Following Mao et al. (2024), we compare our method's performance with the baselines in the following two settings:

- **SUBSTITUTE**: Use the documents retrieved by the rewritten query to answer the question.

Methods	NQ		TriviaQA		ASQA		2WikiMQA		PopQA		Avg.	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Direct	30.90	38.45	59.90	65.91	36.31	45.90	25.70	29.57	25.50	27.75	35.66	41.51
OriQR	40.20	49.04	62.00	67.31	47.71	56.60	24.00	27.53	27.10	28.88	40.20	45.87
SUBSTITUTE												
LLMQR	40.50	48.94	62.42	68.37	48.83	56.96	25.83	29.51	28.10	29.44	41.14	46.64
RetPO	41.00	49.58	61.90	68.23	48.60	56.74	25.30	28.77	29.20	30.87	41.20	46.84
RRR	40.70	49.57	62.50	68.50	48.94	56.74	25.50	28.74	28.90	30.66	41.31	46.84
RaFe	40.30	48.32	61.90	68.08	47.82	56.07	25.70	29.36	29.20	31.04	40.98	46.57
DynQR	**42.10**	**49.94**	**63.30**	**68.67**	**50.50**	**58.86**	**26.20**	**29.77**	**29.60**	**31.23**	**42.34**	**47.69**
EXPAND												
LLMQR	40.44	49.32	61.96	67.77	47.71	56.29	24.42	27.98	28.90	30.72	40.69	46.42
RetPO	41.30	49.72	62.20	67.93	48.60	56.88	23.90	27.60	29.40	31.14	41.08	46.65
RRR	40.70	49.29	62.40	68.22	47.82	56.49	24.50	27.80	29.10	30.52	40.90	46.46
RaFe	39.90	48.40	61.80	67.86	49.05	57.35	24.80	28.41	29.30	30.82	40.97	46.57
DynQR	**41.80**	**50.19**	**62.70**	**68.23**	**50.50**	**58.77**	**25.10**	**28.74**	**29.60**	**31.67**	**41.94**	**47.52**

Table 1: Performance comparison on five QA datasets under both the Substitute and Expand settings.

- **EXPAND**: Use documents from both the original and rewritten query, applying a circulating mechanism to iteratively gather documents until the desired number is reached.

3.3 Implementation Details

In our experiment, the rewriter model is initialized with the Llama-2-7B[1]. We employ Llama-2-7B, Meta-Llama-3-8B[2], and Llama-2-13B[3] as the reader LLMs. We use GPT-4-Turbo as the data labeler in the supervised distillation stage. We use Wikipedia dump from Jan. 27, 2020 as our retrieval corpus and use DPR (Karpukhin et al., 2020) as our dense retriever. For each query, we retrieve the top-5 most similar documents from the corpus. For more details, please refer to Appendix B.

4 Experimental Results

4.1 Main Results

In this section, we present the results of experiments conducted on five QA datasets under both the Substitute and Expand settings, using Meta-Llama-3-8B as the reader. Based on the results in Table 1, several key observations can be made:

First, our method achieves the best performance across all datasets in both the Substitute and Expand settings. This is primarily because our query rewriter effectively caters to the reader's information needs by retrieving documents that significantly reduce the reader's uncertainty. Furthermore, the post-verification and active rewriting mechanisms help minimize noise from potentially suboptimal rewrites, thus improving the robustness of the query rewriting process.

Second, between the two settings, our method shows more substantial improvement in the Substitute setting. This is mainly because, in the Substitute setting, all retrieved documents originate from the rewritten query, whereas in the Expand setting, documents come from both the original and rewritten queries. As a result, when the method is particularly effective, the Substitute setting yields greater improvements, further confirming the effectiveness of our approach.

Third, among the baselines, RETPO performs relatively well due to its effective use of question answers as supervision. Although RRR also leverages question answers, its labels are highly sparse due to the rigorous requirements of the Exact Match metric. This sparsity minimizes the distinction between nearly correct answers and incorrect ones, resulting in weaker performance. In contrast, our method utilizes uncertainty metrics to evaluate the quality of rewritten queries, capturing subtle differences between query qualities and enriching the supervisory signals.

4.2 Ablation Study

In this section, we assess the impact of each component of our model by gradually removing them one at a time. Specifically, we conduct experiments

[1] https://huggingface.co/meta-llama/Llama-2-7b-hf
[2] https://huggingface.co/meta-llama/Meta-Llama-3-8B
[3] https://huggingface.co/meta-llama/Llama-2-13b-hf

Methods	EM	F1
DynQR	50.50	58.77
-w/o Post Verification	50.28	58.50
-w/o Active Rewriting	49.39	57.38
-w/o Preference Alignment	48.38	56.94

Table 2: Ablation Study. We experiment by gradually removing all components on the ASQA dataset using the EXPAND setting.

	NQ		**ASQA**		**2WikiMQA**	
θ	**EM**	**Freq**	**EM**	**Freq**	**EM**	**Freq**
1.0	41.60	1.00	49.83	1.00	26.80	1.00
1.1	41.80	0.98	49.94	0.99	26.90	0.99
1.2	41.90	0.87	49.83	0.88	26.70	0.88
1.3	41.80	0.75	49.72	0.72	26.40	0.68
1.4	41.60	0.63	49.50	0.58	26.20	0.54

Table 3: Performance with different rewriting threshold.

Methods	**TriviaQA**		**ASQA**		**PopQA**	
	EM	**F1**	**EM**	**F1**	**EM**	**F1**
LLAMA-2-7B						
Direct	52.16	60.03	32.74	42.69	20.04	22.26
OriQR	56.30	63.97	44.67	54.69	29.20	30.50
LLMQR	57.76	65.27	45.25	54.28	29.70	30.84
RetPO	58.30	66.06	46.70	55.59	31.00	32.28
RRR	57.80	65.22	46.70	54.75	27.90	29.21
RaFe	57.80	65.79	47.71	56.47	31.50	32.78
DynQR	**58.60**	**66.19**	**48.38**	**57.91**	**31.60**	**32.84**
LLAMA-2-13B						
Direct	60.10	66.70	37.99	48.26	18.80	22.31
OriQR	61.40	68.91	49.50	58.78	29.00	30.06
LLMQR	61.92	69.21	48.94	58.52	28.00	29.13
RetPO	62.30	70.01	50.95	59.71	31.90	33.35
RRR	62.50	69.62	52.18	60.49	29.30	30.65
RaFe	62.50	69.95	50.39	58.90	30.20	31.65
DynQR	**62.90**	**70.60**	**53.07**	**61.53**	**32.60**	**34.05**

Table 4: Result comparison using readers of different parameter sizes under the Substitute setting.

on the NQ dataset under both rewriting settings.

As shown in Table 2, removing any component results in performance degradation, confirming the significance of each part. Notably, removing Preference Alignment causes the largest drop in performance. This is because preference alignment guides the rewriter to generate queries that better meet the reader's information needs by retrieving documents that significantly reduce the reader's uncertainty. Without preference alignment, the rewriter generates semantically similar queries without targeted optimization, leading to inferior results. Additionally, both the Post-Verification and Active Rewriting mechanisms contribute to improved robustness by mitigating suboptimal rewrites that could introduce noise, thereby enhancing overall performance.

4.3 Hyper-parameter Study

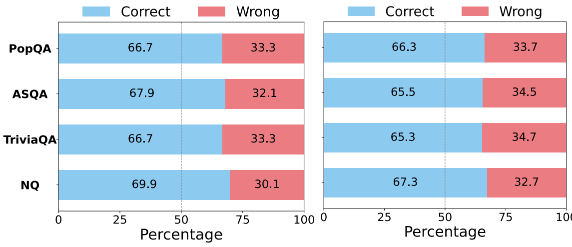
In DynQR, we use a predefined hyperparameter to determine whether to activate the query rewriter. In this section, we analyze the impact of the threshold value p on model performance. Specifically, we tune the threshold on the NQ, ASQA, and 2WikiMQA datasets, with the corresponding results presented in Table 3.

The results indicate that as the threshold decreases, the frequency of query rewriting increases, leading to higher inference costs. However, performance does not consistently improve with increased rewriting frequency; instead, it initially improves and then declines. This behavior can be attributed to the fact that, with a low threshold, the model tends to rewrite queries that are already effective in retrieving the necessary information, resulting in redundant rewritings. Conversely, when the threshold is set too high, queries that would benefit from rewriting remain unchanged, leading to suboptimal performance.

4.4 Analysis

Generalization Ability In this section, we evaluate the generalization ability of our methods by conducting experiments using readers of varying parameter sizes. Specifically, we use Llama-2-7B and Llama-2-13B as the reader LLMs.

As shown in Table 4, switching from Llama-2-7B to Llama-2-13B generally results in performance improvements across all methods, attributed to the enhanced reasoning ability of the larger reader model. Importantly, our method consistently achieves the best performance across all datasets, regardless of the reader LLMs used, demonstrating its strong generalization capability. Notably, achieving performance gains with more advanced readers is typically challenging due to their already strong baseline performance. However, our method maintains comparable improvements even with Llama-2-13B. We attribute this to the fact that as the parameter size of the readers increases, the uncertainty metrics provide a more accurate reflection of the answer quality, as also noted by Chen et al. (2024). As a result, the preference alignment



(a) On Meta-Llama-3-8B (b) On Llama-2-13B

Figure 3: Uncertainty reliability study, where "Correct" means uncertainty decreases with the ground truth document, and "Wrong" means the opposite.

labels become more precise, leading to a more effective query rewriter.

Uncertainty Reliability In DynQR, we utilize the uncertainty of answers to represent the quality of the queries, under the assumption that answers with low uncertainty indicate that the retrieved documents likely contain the information needed to answer the question. In this section, we verify this assumption by examining how the uncertainty of answers changes when the quality of the retrieved documents is improved. Specifically, we randomly replace one document in the retrieved documents with a ground truth document that contains the correct answer, and then prompt the reader LLM to answer the question. We compare the uncertainty of the answers before and after the inclusion of the ground truth document and record the percentage of cases where the uncertainty decreases.

As shown in Figure 3, after adding the ground truth document, the uncertainty of the answers decreases in most cases. This indicates that improving the quality of the retrieved documents can indeed lead to a reduction in the reader's uncertainty. This finding verifies that by comparing the uncertainties of two answers, we can accurately assess the quality of the documents, and by extension, the quality of the queries used to retrieve them. Moreover, we observe that the decrease in uncertainty is more pronounced with Meta-Llama-3-8B. This is likely because stronger LLMs can better reflect the quality of the documents through their uncertainty measures, a phenomenon also observed in Chen et al. (2024). Therefore, we believe that the uncertainty-based labeling method can achieve even better performance with LLMs that possess stronger reasoning abilities.

Uncertainty Categories In this section, we explore various metrics for estimating LLM uncertainty. Perplexity estimates uncertainty based on the log probabilities of generated tokens (Fomicheva et al., 2020). Length Normalized Entropy (LN-Entropy) is a normalized version of entropy (Malinin and Gales, 2020). Probability-based estimation assesses uncertainty by focusing on the tokens with the lowest probabilities (Jiang et al., 2023). Finally, the energy-based method evaluates uncertainty in the logit space, aiming to detect out-of-distribution samples (Liu et al., 2020).

Metrics	TriviaQA		ASQA		PopQA	
	EM	F1	EM	F1	EM	F1
Perplexity	58.40	66.01	46.70	54.72	28.90	30.32
LN-Entropy	57.70	65.41	45.92	54.55	28.40	29.78
Probability	57.70	65.15	45.47	53.90	26.50	27.47
Energy	57.60	65.31	45.59	54.56	25.70	27.31

Table 5: Performance with different uncertainty metrics.

We conducted experiments on subsets of the TriviaQA, ASQA, and PopQA datasets, using Llama-2-7B as the reader. As shown in Table 5, the perplexity-based method consistently outperforms all other metrics across the datasets, while the energy-based method performs the worst, aligning with findings in Yao et al. (2024). Additionally, the perplexity-based method exhibits a more stable value range, typically between [1, 2], which simplifies the tuning of the activation threshold. Based on these observations, we selected perplexity as the uncertainty measure for our experiments.

4.5 Case Study

In this section, we analyze the effectiveness of our method using cases from the NQ and ASQA datasets, as shown in Table 6. After rewriting, queries generally exhibit improved formatting, specificity, and grammar, which enhances the accuracy of retrieved answers. In Case 1 (Better Format), the original query "Who plays elsa's aunt in once upon a time?" is rewritten to improve capitalization and formatting, resulting in the correct answer, Elizabeth Mitchell. In Case 2 (Enhanced Specificity), "Who has won the most f1 grand prix?" is rewritten to clarify that it refers to a "driver," which helps accurately identify Michael Schumacher as the answer. In Case 3 (Corrected Grammar), "When is season 14 of grey's anatomy coming back?" is rewritten with proper grammar and formality, leading to the correct premiere date of September 28, 2017. These cases illustrate that our method significantly enhances query quality, improving document retrieval and answer accuracy through better formatting, clarity, and specificity.

Case 1: Better Format
Original Query: Who plays elsa's aunt in once upon a time? Rewrite Query: In the show "Once Upon a Time," what is the identity of Elsa's aunt? Retrieved Documents: **Document 1**: Rumplestiltskin told her that her parents were afraid of Elsa. She does not tell this to Elsa, but is shocked to see that Elsa is learning to control her power—due to a new woman by the name of Ingrid (Elizabeth Mitchell), who claims she is... **Document 2**: As she searches for her sister Anna (Elizabeth Lail) with the aid of the main characters, they encounter the Snow Queen (Elizabeth Mitchell). Meanwhile, Regina seeks the Author of Henryś Once Upon a Time book so that she can finally... Answer: Elizabeth Mitchell [CORRECT]
Case 2: Enhanced Specificity
Original Query: Who has won the most f1 grand prix? Rewrite Query: Which driver has the greatest number of Formula 1 victories? Retrieved Documents: **Document 1**: Formula One drivers have won the World Drivers's Championship, with Michael Schumacher holding the record for most championships with seven, as well as holding the race wins record. Juan Manuel Fangio and Lewis Hamilton have... **Document 2**: There have been 52 Formula One drivers from Germany including three world champions, one of whom is currently racing in the sport. Michael Schumacher holds many records in F1 including the most world championship titles... Answer: Michael Schumacher [CORRECT]
Case 3: Corrected Grammar
Original Query: When is season 14 of grey's anatomy coming back? Rewrite Query: When does Grey's Anatomy return for its fourteenth season? Retrieved Documents: **Document 1**: The fourteenth season of the American television medical drama Greyś Anatomy was ordered on February 10, 2017, by American Broadcasting Company (ABC), and premiered on September 28, 2017 with a special two-hour premiere... **Document 2**: U.S. viewers in millions refers to the number of Americans in millions who watched the episodes live. The fourteenth season of the American television medical drama Greyś Anatomy was premiered on September 28, 2017 with... Answer: September 28, 2017 [CORRECT]

Table 6: Case studies of rewritten queries. Blue text indicates the stem, pink text indicates the effective hint, [CORRECT] indicates the judgment of whether the answer is correct.

5 Related Work

5.1 Query Rewriting

Query rewriting is commonly used in retrieval tasks (Wu et al., 2021; Qian and Dou, 2022; Anand et al., 2023) and significantly enhances LLM capabilities in Retrieval Augmented Generation (RAG)(Ram et al., 2023; Jiang et al., 2023; Yao et al., 2024). Many studies leverage LLMs for query rewriting to improve retrieval(Ye et al., 2023; Wang et al., 2023; Shen et al., 2023). For example, RRR (Ma et al., 2023) and RETPO (Yoon et al., 2024), which use downstream performance signals, and RaFe (Mao et al., 2024), which uses document relevance to minimize labeling. However, these methods either rely on human-crafted labels or use indirect, potentially suboptimal feedback. In this paper, we propose using uncertainty as direct feedback, which eliminates the need for handcrafted labels and offers a more effective approach.

5.2 Feedback Learning

Feedback learning has recently been instrumental in aligning LLM outputs with human preferences. Various optimization methods have been developed to enhance LLM capabilities (Zheng et al., 2023; Wang et al., 2024; Rafailov et al., 2024; Yuan et al., 2023), and new feedback signals have been constructed from different perspectives (Lee et al., 2023; Shinn et al., 2024; Pang et al., 2023; Liu et al., 2023; Xu et al., 2023). Feedback learning has also been employed in query rewriting, as seen in RRR (Ma et al., 2023), RETPO (Yoon et al., 2024), and RaFe (Mao et al., 2024). These approaches either depend on hand-crafted labels or rely on indirect signals, limiting their effectiveness. Our method addresses these limitations by using LLM uncertainty as direct feedback, thus eliminating the need for handcrafted labels and improving the effectiveness of feedback-based optimization.

6 Conclusion

In this work, we propose DynQR, an unsupervised query rewriting method that leverages uncertainty-based feedback from the reader LLM, eliminating the need for labeled data from downstream tasks. DynQR employs an active rewriting mechanism and a post-verification process to minimize unnecessary rewrites and reduce noise. We conduct extensive experiments on five datasets across three knowledge-intensive tasks, and the results demonstrate the effectiveness of DynQR.

Limitations

In this paper, we utilize reader LLM's uncertainty as a supervision signal for training the query rewriter. We acknowledge two limitations:

(1) The effectiveness of uncertainty feedback relies on a strong correlation between uncertainty and response quality, which may require the reader LLM to have significant reasoning abilities (e.g., parameter sizes larger than 7B);

(2) Our method incurs a small additional computational cost for uncertainty calculations.

Ethics Statement

This work complies with the ACL Ethics Policy. All datasets and LLMs used are publicly available. Our research focuses on an annotation-free method for training query rewriters, and we do not anticipate any negative ethical impacts.

References

Abhijit Anand, Vinay Setty, Avishek Anand, et al. 2023. Context aware query rewriting for text rankers using llm. *arXiv preprint arXiv:2308.16753*.

Bernd Bohnet, Vinh Q Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, Kai Hui, et al. 2022. Attributed question answering: Evaluation and modeling for attributed large language models. *arXiv preprint arXiv:2212.08037*.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. Inside: Llms' internal states retain the power of hallucination detection. *arXiv preprint arXiv:2402.03744*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, et al. 2023. Lm-polygraph: Uncertainty estimation for language models. *arXiv preprint arXiv:2311.07383*.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. 2022. Rarr: Researching and revising what language models say, using language models. *arXiv preprint arXiv:2210.08726*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Wenyue Hua, Lifeng Jin, Linfeng Song, Haitao Mi, Yongfeng Zhang, and Dong Yu. 2023. Discover, explain, improve: An automatic slice detection benchmark for natural language processing. *Transactions of the Association for Computational Linguistics*, 11:1537–1552.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models. *arXiv preprint arXiv:2303.05398*.

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.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*.

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

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, 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. *TACL 2019*, pages 452–466.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857.

Jiacheng Liu, Ramakanth Pasunuru, Hannaneh Hajishirzi, Yejin Choi, and Asli Celikyilmaz. 2023. Crystal: Introspective reasoners reinforced with self-feedback. *arXiv preprint arXiv:2310.04921*.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. 2020. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475.

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

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*.

Shengyu Mao, Yong Jiang, Boli Chen, Xiao Li, Peng Wang, Xinyu Wang, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. 2024. Rafe: Ranking feedback improves query rewriting for rag. *arXiv preprint arXiv:2405.14431*.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. 2022. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*.

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. *arXiv preprint arXiv:2305.14483*.

Benjamin Plaut, Khanh Nguyen, and Tu Trinh. 2024. Softmax probabilities (mostly) predict large language model correctness on multiple-choice q&a. *arXiv preprint arXiv:2402.13213*.

Hongjin Qian and Zhicheng Dou. 2022. Explicit query rewriting for conversational dense retrieval. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4725–4737.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2021. Measuring attribution in natural language generation models. *arXiv preprint arXiv:2112.12870*.

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

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: factoid questions meet long-form answers. 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 8273–8288. Association for Computational Linguistics.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *CoRR*, abs/2211.09085.

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. *arXiv preprint arXiv:2303.07678*.

Orion Weller, Kyle Lo, David Wadden, Dawn Lawrie, Benjamin Van Durme, Arman Cohan, and Luca Soldaini. 2023. When do generative query and document expansions fail? a comprehensive study across methods, retrievers, and datasets. *arXiv preprint arXiv:2309.08541*.

Zeqiu Wu, Yi Luan, Hannah Rashkin, David Reitter, Hannaneh Hajishirzi, Mari Ostendorf, and Gaurav Singh Tomar. 2021. Conqrr: Conversational query rewriting for retrieval with reinforcement learning. *arXiv preprint arXiv:2112.08558*.

Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. 2024. Evaluating mathematical reasoning beyond accuracy. *arXiv preprint arXiv:2404.05692*.

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*.

Ryutaro Yamauchi, Sho Sonoda, Akiyoshi Sannai, and Wataru Kumagai. 2023. Lpml: llm-prompting markup language for mathematical reasoning. *arXiv preprint arXiv:2309.13078*.

Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, Linmei Hu, Weichuan Liu, Lei Hou, and Juanzi Li. 2024. Seakr: Self-aware knowledge retrieval for adaptive retrieval augmented generation. *arXiv preprint arXiv:2406.19215*.

Fanghua Ye, Meng Fang, Shenghui Li, and Emine Yilmaz. 2023. Enhancing conversational search: Large language model-aided informative query rewriting. *arXiv preprint arXiv:2310.09716*.

Chanwoong Yoon, Gangwoo Kim, Byeongguk Jeon, Sungdong Kim, Yohan Jo, and Jaewoo Kang. 2024. Ask optimal questions: Aligning large language models with retriever's preference in conversational search. *arXiv preprint arXiv:2402.11827*.

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. *arXiv preprint arXiv:2304.05302*.

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley Malin, and Sricharan Kumar. 2023. Sac3: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15445–15458.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv: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*.

A Dataset Statistics

The dataset statistics used in this paper are shown in Table 7.

Settings	NQ (Kwiatkowski et al., 2019)	TriviaQA (Joshi et al., 2017)	PopQA (Mallen et al., 2022)	2WikiMQA (Ho et al., 2020)	ASQA (Stelmakh et al., 2022)
			Dataset statistics		
Task	Open-domain QA	Open-domain QA	Open-domain QA	Multi-hop QA	Ambiguous QA
Train Data	60,000	60,000	0	0	0
Test Data	1,000	1,000	1,000	1,000	895
			Evaluation settings		
Metrics	EM, F1	EM, F1	EM, F1	EM, F1	EM, F1
			Retrieval settings		
Corpus	Wikipedia	Wikipedia	Wikipedia	Wikipedia	Wikipedia
Retriever	DPR	DPR	DPR	DPR	DPR

Table 7: Statistics and experimental settings of different tasks/datasets.

B Implementation Details

Reward Signals The reward calculation method for baselines are:

- **RRR** (Ma et al., 2023): The reward signal is based on whether the retrieved documents lead to a correct answer when processed by the reader.
- **RETPO** (Yoon et al., 2024): The reward comes from whether the retrieved documents contain a correct answer.
- **RaFe** (Mao et al., 2024): The reward signal is derived from whether the rewritten query leads to documents that are more relevant to the original query.

Training Process We conducted full parameter fine-tuning during both stages.

- **Supervised Distillation Stage**: We randomly sampled 30,000 queries from the NQ dataset and 30,000 queries from the TriviaQA dataset for supervised fine-tuning. The model (LLama-2-7B) was fully fine-tuned for 1 epoch with a learning rate of 1e-6 and a batch size of 100.
- **Preference Alignment Stage**: In this stage, we sample another 30,000 queries from the NQ dataset and another 30,000 queries from the TriviaQA dataset. Then we conduct query rewriting for these queries and construct the preference labeling based on the uncertainty of different rewrites for each reader LLM. The rewriter model was further fine-tuned for 2 epochs with a learning rate of 1e-5 and a batch size of 20 using Direct Preference Optimization (Rafailov et al., 2024).

Active Rewriting Threshold In our experiments, we sample 100 queries from the test dataset as the validation set, and the remaining queries are used as the test set. We then tuned the active rewriting threshold based on its performance on the validation set and selected the one that performed the best.

C Prompts

The prompts used in our experiments are listed as follows.

Prompt: Answering with Retrieval

Instruction: Refer to the documents and answer the question with only one entity without giving any explanation. Here is an example:
Documents:....
Question: who did lebron james play for before the cleveland cavalier?
The answer is: Miami Heat
Now refer to the documents below and answer the question with only one entity without giving any explanation:
Documents: {background}
Question: {query}
The answer is

Prompt: Answering without Retrieval

Instruction: Answer the question as short as possible without giving any explanation.
Question: who did lebron james play for before the cleveland cavalier?
The answer is: Miami Heat.
Question: {query}
The answer is:

Prompt: Query Rewriting

Instruction: output the rewrite of input query.
Query:{query}
Output: