Ask Optimal Questions: Aligning Large Language Models with Retriever's Preference in Conversation

Chanwoong Yoon[1*] **Gangwoo Kim**[1*] **Byeongguk Jeon**[2] **Sungdong Kim**[2]
Yohan Jo[3] **Jaewoo Kang**[1,4†]
Korea University[1] KAIST AI[2] Seoul National University[3] AIGEN Sciences[4]
{cwyoon99, gangwoo_kim, kangj}@korea.ac.kr
{sungdong.kim, byeongguk}@kaist.ac.kr yohan.jo@snu.ac.kr

Abstract

Conversational search, unlike single-turn retrieval tasks, requires understanding the current question within a dialogue context. The common approach of *rewrite-then-retrieve* aims to decontextualize questions to be self-sufficient for off-the-shelf retrievers, but most existing methods produce sub-optimal query rewrites due to the limited ability to incorporate signals from the retrieval results. To overcome this limitation, we present a novel framework RETPO (**Ret**riever's **P**reference **O**ptimization), which is designed to optimize a language model (LM) for reformulating search queries in line with the preferences of the target retrieval systems. The process begins by prompting a large LM to produce various potential rewrites and then collects retrieval performance for these rewrites as the retrievers' preferences. Through the process, we construct a large-scale dataset called RF COLLECTION, containing **R**etrievers' **F**eedback on over 410K query rewrites across 12K conversations. Furthermore, we fine-tune a smaller LM on this dataset to align it with the retrievers' feedback. Our resulting model demonstrates superiority on two benchmarks, surpassing the previous state-of-the-art performance of *rewrite-then-retrieve* approaches. [1]

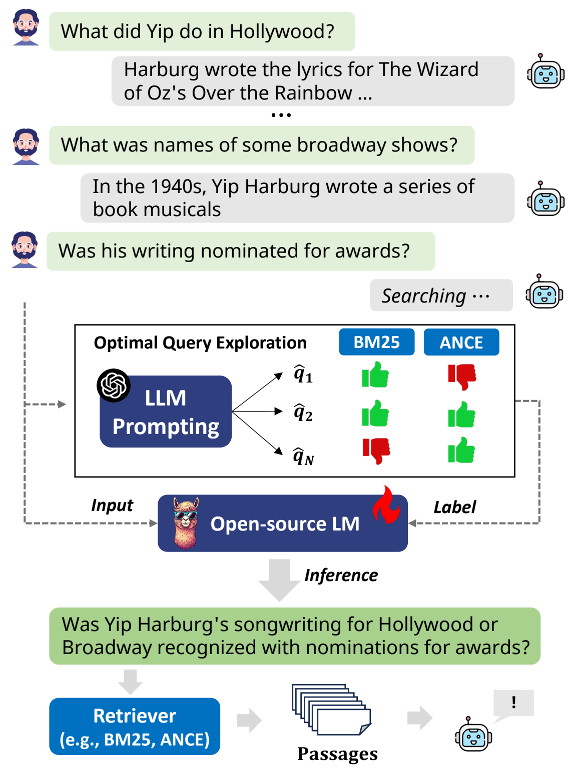

Figure 1: Overview of RETPO. Given a conversation and a follow-up question, (1) potential rewrites $\hat{q}_i$ are generated by prompting an LLM. (2) Retriever's preferences for each rewrite are collected. (3) A smaller LM is trained to be aligned with the retriever's preferences. The resulting model can generate clear and specific rewrites.

1 Introduction

Conversational search extends the information retrieval to encompass nuances of dialogue context. Unlike standard retrieval tasks in open-domain question answering (QA) (Joshi et al., 2017; Kwiatkowski et al., 2019), the task is characterized by conversational dependencies in questions (e.g., omission, ambiguity, and coreference) (Qu et al., 2020; Anantha et al., 2021; Adlakha et al., 2022). As depicted in Figure 1, the question in the last turn "*Was his writing nominated for awards?*" could only be understood within the context. Hence, conventional retrieval systems that are not designed to consider dialogue context tend to yield poor retrieval performance.

A prevalent approach to overcome this challenge is *rewrite-then-retrieve*, where questions are decontextualized and made self-contained before being used for retrieval systems. In many prior works, language models (LMs) are trained for question rewriting (QR) using human rewrites as ground truth (Elgohary et al., 2019; Anantha et al., 2021;

[*] Equal contribution [†] Corresponding author
[1]The code and dataset are available at github.com/dmis-lab/RetPO

Findings of the Association for Computational Linguistics: NAACL 2025, pages 5914–5936
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

Vakulenko et al., 2021; Qian and Dou, 2022). However, this approach often results in less effective rewrites for search purposes, as human rewrites are typically created without considering their impact on retrieval performance. Although recent studies (Wu et al., 2022; Mo et al., 2023) suggest incorporating signals from retrieval results into the training of QR models, there is still a challenge in fully utilizing the retrievers' preferences across various potential rewrites.

To align a QR model with retrievers' preferences, we present RETPO (**Ret**riever's **P**reference **O**ptimization). This novel framework aims to optimize a language model (LM) to produce query rewrites tailored to a target retriever's feedback. RETPO involves several key steps: (1) we begin with instructing a superior large LM (LLM), GPT-4 (OpenAI, 2023), to provide a variety of potential rewrites with several prompting methods. (2) We then gather the retriever's feedback on each rewrite (i.e., retrieval performance), resulting in a large-scale dataset RF COLLECTION, containing **R**etrievers' **F**eedback on over 410K query rewrites refined for search purpose across 12K conversations. (3) Based on our dataset, we further align an open-source LM, such as Llama2-7b (Touvron et al., 2023), with preference-driven optimization. The LM is optimized to generate preferred rewrites over less preferred ones and then is used for the inference phase.

Our experimental results demonstrate that RETPO largely advances retrieval performances on two recent conversational search benchmarks, QReCC (Anantha et al., 2021) and TopiOCQA (Adlakha et al., 2022). Notably, our 7-billion-parameter model outperforms existing QR approaches, including its teacher model GPT-4. It also surpasses the previous state-of-the-art performance of BM25 by significant margins 11.8 (MRR) and 19.0 (Recall@10) on QReCC. Furthermore, we thoroughly analyze our rewrites from RF COLLECTION and RETPO. The results demonstrate our methods tend to produce specific and detailed rewrites as exemplified in Figure 1, contributing to the superior retrieval performance. In GPT-4 evaluation, our rewrites are more favored than human rewrites in terms of clarity and informativeness.

Our contributions are summarized in threefold:

- We define optimal query in conversational search and propose how to explore and exploit it. To our knowledge, RETPO is the first to leverage retriever preference-driven optimization for query reformulation.
- We construct and release RF COLLECTION, a large-scale dataset of **R**etriever's **F**eedback on query rewrites in dialogue. Our rewrites are superior to human rewrites in retrieval tasks and GPT-4 evaluation.
- We align an open-source LM with our dataset. It achieves new state-of-the-art performance of *rewrite-then-retrieve* approaches on two benchmarks, QReCC and TopiOCQA.

2 Background

2.1 Task Formulation

In conversational search, given the current question q_t and the conversation history of question-answer pairs $H_{<t} = \{q_i, a_i\}_{i=1}^{t-1}$,[2] a retrieval system $\text{Ret}(q)$ returns the top-k relevant passages $D_k = \{d_i\}_{i=1}^{k}$ from the target corpus. In recent *rewrite-then-retrieve* approaches (Anantha et al., 2021; Adlakha et al., 2022), a question rewriting model π_{QR} is trained to generate a self-contained question q by encoding a concatenation of the utterances so far $x = \text{Concat}(H_{<t}, q_t)$; then it predicts a rewrite $\hat{q}$ for use with off-the-shelf retrievers. Since self-contained rewrites are not always available in natural conversation, most studies rely on the human rewrites for supervision (Elgohary et al., 2019).

2.2 Definition of Optimal Query

Given an evaluation metric $\text{Eval}(\cdot, d^+)$ assessing the retrieved passages based on the gold passage d^+, we define optimal query q^* as a query that maximizes the evaluation score as follows:

$$q^* = \arg\max_q \text{Eval}(\text{Ret}(q), d^+)$$

Note that we assume $\text{Ret}(\cdot)$ as frozen. Under the definition, we argue that previous works using human rewrites as ground truth would result in suboptimal queries. The human rewrites are crafted without considering the subsequent retrieval process and its end performance, simply focusing on resolving conversational dependencies. Although a few studies (Wu et al., 2022; Mo et al., 2023) try to incorporate the training signals from the retrieval step, they could not exploit training signals from contrasting multiple queries explored with various reasoning types.

[2]We drop the subscript in the later sections to avoid clutter.

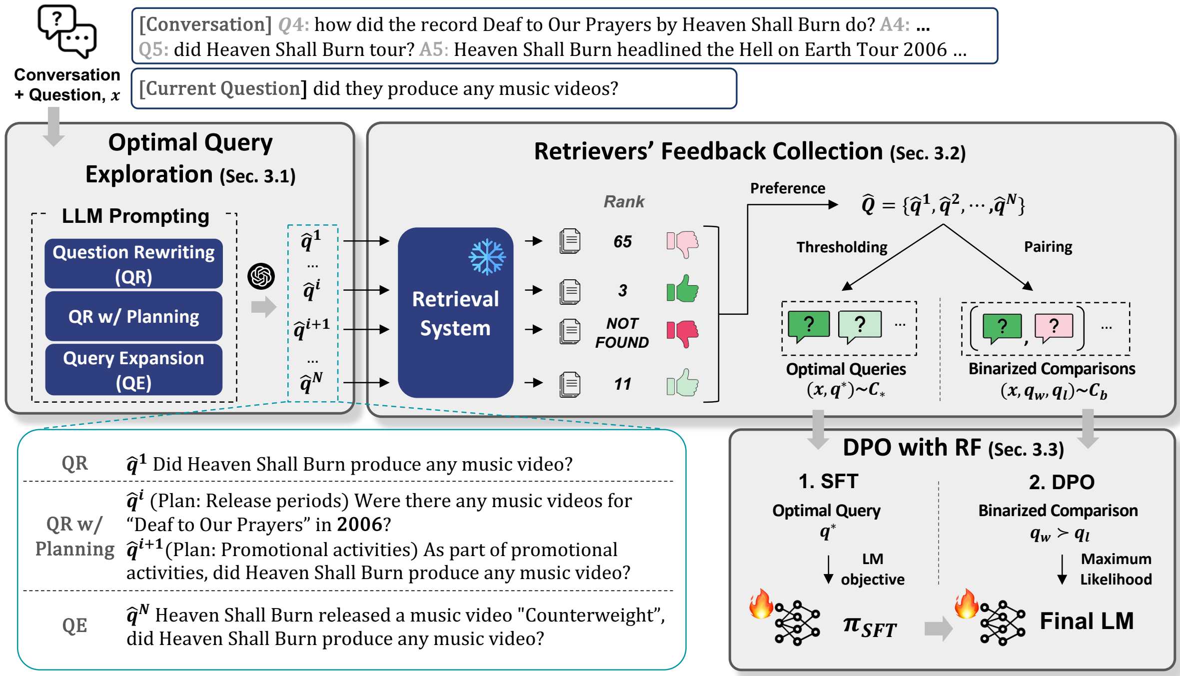

Figure 2: Components of RETPO designed to align an LM with retrievers' preferences. Given a conversation and a user question, we first prompt a capable LLM to provide potential rewrites using various prompting methods (*Optimal Query Exploration*; Sec. 3.1). We then collect the retrievers' feedback on each rewrite by measuring their retrieval performance, leading to two datasets: optimal queries $\mathcal{C}_*$ and query pairs $\mathcal{C}_b$ (RF COLLECTION; Sec. 3.2) Lastly, we optimize an open-source LM with our datasets, encouraging it to generate preferable rewrites (Sec. 3.3).

3 Retriever's Preference Optimization

We newly introduce RETPO (**Ret**riever's **P**reference **O**ptimization) designed to optimize a query reformulation model with the preference of the retrieval system as illustrated in Figure 2. We first explore a range of potential rewrites with various prompting methods *(Optimal Query Exploration*; Sec. 3.1). We then collect the retriever's feedback on each rewrite by measuring the retrieval performance, resulting in a large-scale dataset, RF COLLECTION (Sec. 3.2). By using the dataset, we further align an open-source LM with the preference-driven optimization (Sec. 3.3).

3.1 Optimal Query Exploration

To explore a broad range of effective search queries, we first prompt a superior LLM to provide a number of potential rewrites. Based on the conversation and the current question, we prompt GPT-4 (OpenAI, 2023) with various prompting methods based on different reasoning abilities and purposes.

We adopt three prompting methods: (a) **Question Rewriting**[3] requests the LLM to contextualize the question by resolving coreferences and ellipses. For example, in Figure 2, it finds what a pronoun "*they*" in the current question indicates and then replaces it with the exact entity "*Heaven Shall Burn*" in the rewrite $\hat{q}^1$. We initiate our task instruction following Ye et al. (2023) to enhance informativeness and consistency of the rewrite by mentioning '*The resulting question should retain its original meaning and be as informative as possible*.'

Moving beyond resolving the explicit dependencies, we devise (b) **QR with Planning**[4] that allows the LLM to identify an important point to be asked and specify the question's aim. For example, in Figure 2, the rewrite $\hat{q}^i$ inquires about the specific music video and release period mentioned in the conversation. To this end, it performs an intermediate reasoning step before generating the rewrite, inspired by Chain-of-Thought prompting (Wei et al., 2022). In particular, we encourage the LLM to elicit relevant information from its parametric knowledge or the held-out conversation.

In addition, we adopt (c) **Query Expansion**,[5] recently known to be effective in retrieval tasks (Mao et al., 2021; Wang et al., 2023; Mo et al., 2023).

[3] See Table 17 for the question rewriting prompt.

[4] See Table 18 for the planning prompt.

[5] See Table 19 for the query expansion prompt.

We first instruct the LLM to provide a plausible answer or relevant information without access to external knowledge. We then append the pseudo-answer to a self-contained rewrite, either a human rewrite if available or the result of the QR prompting method. As exemplified in Figure 2, the rewrite $\hat{q}^N$ is composed of multiple sentences containing the potential answer "*Counterweight*". It increases the chance of keyword overlap between the query and the gold passage, providing informative clues to the retrieval system.

With each prompting method, the LLM generates a long text containing from five to ten queries separated by the special token in a single call. By doing so, it prevents the LLM from generating duplicated queries, resulting in more diverse queries. As a result, our synthetic queries vary in terms of format and intent.

3.2 Retrievers' Feedback Collection

Upon the queries collected through the *Optimal Query Exploration*, we gather feedback from target retrievers. In particular, we feed each query candidate to the frozen retriever and evaluate the outcome. The retrieval performance is considered as a measurement of the preference. We use the relative rank of the gold passage in the retrieved passage set. We eventually construct a synthetic dataset, RF COLLECTION, **R**etrievers' **F**eedback on 410K query rewrites across 12K conversations.[6]

Our dataset consists of two sets, one for supervised fine-tuning and one for preference optimization (discussed in the later section). We first construct a collection of optimal queries $\mathcal{C}_*$ under our definition. Specifically, we choose the five highest-ranked rewrites whose ranks are within a pre-defined threshold. If all generated queries fail to surpass the threshold, we select the highest-ranked rewrite. It is used for fine-tuning our model with the language modeling objective, potentially replacing human rewrites.

For the preference optimization, we construct a collection of binarized comparisons $\mathcal{C}_b$ based on the retriever's feedback. Given all rewrite candidates for the same input x, we first sort them by their rank in ascending order, resulting in $\hat{Q} = \{\hat{q}^1, \hat{q}^2, \cdots, \hat{q}^{|\hat{Q}|}\}$, where the preference becomes $\hat{q}^1 \succ \hat{q}^2 \succ \cdots \succ \hat{q}^{|\hat{Q}|}$. We then obtain valid pairs of distinct queries $\{(\hat{q}^j, \hat{q}^k) : j < k\}$ without duplication of query or rank. We randomly sample comparison pairs (q_w, q_l) of 'preferred' query q_w and 'dispreferred' query q_l. We filter out cases where the preferred query fails to surpass a rank threshold.

[6]We thoroughly analyze the dataset in Sec. E.3.

3.3 Direct Preference Optimization with Retrievers' Feedback

Based on RF COLLECTION, we align a smaller open-source LM with the retriever's preference. We first fine-tune an LM on the collection of optimal queries in a supervised manner (SFT). We further align the fine-tuned model with direct preference optimization (DPO) (Rafailov et al., 2023).

Supervised Fine-Tuning To build an LM that effectively reformulates a question, we fine-tune it in two steps. The LM is first trained to replicate the ground-truth response following the utterances. It also aims to benefit the capability to generate pseudo-answers in the query expansion. We subsequently fine-tune the LM on the optimal queries we collect. To this end, it learns to generate self-contained and preferable rewrites. Specifically, we optimize the LM to maximize the log-likelihood for returning the tokens of optimal rewrites q^* from the collection $\mathcal{C}_*$. Given the input $x = \text{Concat}(H_{<t}, q_t)$, the LM π is trained as:

$$\pi_{SFT} = \max_{\pi} \mathbb{E}_{(x,q^*)\sim\mathcal{C}_*} \log \pi(q^* \mid x)$$

Direct Preference Optimization Initiating with the SFT model, we further align the LM with the retrievers' preferences. In particular, we apply DPO, a method recently highlighted by Rafailov et al. (2023), for its efficacy in alignment learning. It optimizes the student model π_θ to maximize the likelihood of generating the preferred q^w over q^l, starting from the π_{SFT}.

$$J(\theta) = \mathbb{E}_{(x,q_w,q_l)\sim\mathcal{C}_b} \log \sigma(r_\theta(x, q_w) - r_\theta(x, q_l))$$

Following Rafailov et al. (2023), we simplify $r_\theta(x, q) = \beta \log \pi(q \mid x) - \beta \log \pi_{SFT}(q \mid x)$ with the likelihood difference with the SFT model. This process is guided by the principle of maximizing the contrast between preferred and dispreferred rewrites, thereby providing a clear signal for model training. DPO enables the model to directly learn from the contrast by focusing on the relative merits of each rewrite as judged by the retrieval system. Through this targeted optimization, the SFT model is further trained to generate rewrites that reflect the nuanced preferences of the target retriever.

Type	Query Reform.	TopiOCQA MRR	NDCG	R@10	R@100	QReCC MRR	NDCG	R@10	R@100
Sparse (BM25)	`Original`	2.1	1.8	4.0	9.1	6.5	5.6	11.1	21.5
	`Human Rewrite`	-	-	-	-	39.8	36.3	62.7	98.5
	T5QR	11.3	9.8	22.1	44.7	33.4	30.2	53.8	86.1
	CONQRR	-	-	-	-	38.3	-	60.1	88.9
	ConvGQR	12.4	10.7	23.8	45.6	44.1	41.0	64.4	88.0
	EDIRCS	-	-	-	-	41.2	-	62.7	**90.2**
	LLM IQR	-	-	-	-	49.4	-	67.1	88.2
	IterCQR	16.5	14.9	29.3	54.1	46.7	44.1	64.4	85.5
	RETPO (*Ours*)	**28.3**	**26.5**	**48.3**	**73.1**	**50.0**	**47.3**	**69.5**	89.5
Dense (ANCE)	`Original`	3.0	2.7	6.0	10.2	10.8	9.8	16.8	23.9
	`Human Rewrite`	-	-	-	-	41.3	38.3	63.3	81.7
	T5QR	23.0	22.2	37.6	54.4	34.5	31.8	53.1	72.8
	CONQRR	-	-	-	-	41.8	-	65.1	84.7
	ConvGQR	25.6	24.3	41.8	58.8	42.0	39.1	63.5	81.8
	EDIRCS	-	-	-	-	42.1	-	65.6	**85.3**
	IterCQR	26.3	25.1	42.6	62.0	42.9	40.2	65.5	84.1
	LLM4CS (*Single*)	22.6	21.2	40.1	57.5	33.2	30.8	50.3	66.0
	LLM4CS (*Ensemble*)	32.0	31.1	50.5	67.7	41.8	39.1	62.3	76.2
	RETPO (*Ours*)	**32.2**	**31.1**	**51.6**	**69.5**	**44.0**	**41.1**	**66.7**	84.6

Table 1: Evaluation results of various retrieval system types on the development sets of QReCC (Anantha et al., 2021) and TopiOCQA (Adlakha et al., 2022). We include baselines that integrate the retrievers without fine-tuning. In LLM4CS, *Single* and *Ensemble* refer to REW and RAR prompting, respectively.

4 Experiment

Datasets We test our models on two recent open-domain CQA benchmarks, QReCC (Anantha et al., 2021) and TopiOCQA (Adlakha et al., 2022). QReCC contains 14K conversations with 81K question-answer pairs and self-contained rewrites. TopiOCQA is a more recent benchmark consisting of 3.9K conversations, presenting a challenge of topic switches. To test the models in the *zero-shot* setup, we also include CAsT-20 (Dalton et al., 2021) that does not contain the train set[7].

Retrieval Systems To investigate the impact of different types of retrieval systems, we adopt a sparse retriever BM25 and dense retrievers ANCE (Xiong et al., 2020) and Contriever (Izacard and Grave, 2021), widely used in the task. Specifically, we use the checkpoints trained on MS-MARCO (Bajaj et al., 2016) passage retrieval task. Note that we do not further fine-tune the retrievers for our target task.

Evaluation Metrics We use several evaluation metrics, following previous works. Mean Reciprocal Rank (**MRR**) is the average of the ranks measuring how effectively the retriever can locate gold passages. Normalized Discounted Cumulative Gain (**NDCG@3**) evaluates retrieval results by considering both relevance and rank of top-3 results. **Recall@**k verifies whether the retriever succeeds in locating gold passages within top-k results.

Query Reformulation	TopiOCQA MRR	R@10	R@100
GPT-4 Prompting (Teacher)	18.5	35.1	62.9
Distillation to Llama2-7b	19.0	35.5	64.6
RETPO (*Ours*)	28.3	48.3	73.1
w/o. DPO	23.4	41.6	67.7
w/o. Query Expansion	22.0	40.2	68.5
w/o. QE and Planning	21.8	39.2	67.7

Table 2: Ablation study for each component of RETPO. We compare the baselines that prompt GPT-4 to generate the rewrites and fine-tune smaller LM on them.

Baselines We select several baselines from *rewrite-then-retrieve* approaches. (1) T5QR (Lin et al., 2020) fine-tunes T5-base (Raffel et al., 2020) to replicate human rewrites. (2) CONQRR (Wu et al., 2022) introduces a reinforcement learning framework that leverages retrieval performance as a reward signal. (3) ConvGQR (Mo et al., 2023) fine-tunes QR models with an auxiliary loss function for injecting the knowledge of the target retriever. (4) EDIRCS (Mao et al., 2023a) extracts tokens from the dialogue and adds a few newly generated tokens. (5) IterCQR (Jang et al., 2023) incorporates iterative training of QR model driven

[7] See more details in Appendix A

Query Reformulation	First			TopiOCQA Topic-concentrated			Topic-shifted		
	MRR	R@10	R@100	MRR	R@10	R@100	MRR	R@10	R@100
`Original`	14.7	29.3	64.4	0.9	1.7	4.2	1.1	1.9	4.2
Fine-tuned T5	14.7	29.3	64.4	14.4	28.2	52.4	9.4	18.2	36.9
GPT-4 Prompting	15.6	31.2	62.0	19.7	37.2	65.3	16.4	31.3	57.4
Distillation to Llama2-7b	17.9	34.2	63.9	20.0	37.1	66.3	17.0	32.0	60.7
RETPO (*Ours*)	**32.0**	**51.7**	**75.1**	**27.4**	**47.1**	**72.4**	**29.6**	**50.0**	**74.3**

Table 3: Breakdown evaluation of BM25 on the development set of TopiOCQA, segmented by question type: initial turn (**First**), topic-consistent turns with their preceding one (**Topic-Concentrated**) and topic-switched turns (**Topic-Shifted**). Following Adlakha et al. (2022), we identify a switch of topic if the gold passage is based on a different Wikipedia document.

by query rewrites explored with GPT-3.5. (6) LLM IQR (Ye et al., 2023) prompts GPT-3.5 multiple times to reformulate questions according to pre-determined criteria. LLM4CS (Mao et al., 2023b) proposes three ensemble methods that generate unified query representations by aggregating rewrites and hypothetical responses obtained from multiple inferences of GPT-3.5. We report two variants: the basic approach, REW prompting with MaxProb aggregation, and the best-performing method, RAR prompting with Mean aggregation (refer to *Single* and *Ensemble*, respectively).

We also include baselines that fine-tune retrievers in Sec. 5.3, such as ConvDR (Yu et al., 2021), ConvANCE and LeCoRE (Mao et al., 2023c). The most recent study, InstructoR (Jin et al., 2023), instructs GPT-3.5 to augment the train set for fine-tuning the retriever.

4.1 Main Results

Table 1 shows the evaluation results of various types of retrieval systems.

Leveraging signal from the retriever enhances the end performance. Encoding the current question without modification (`Original`) performs poorly. Performance of T5QR using the human rewrites as supervision is bounded by its label (`Human Rewrite`). Other baselines using the same backbone with signals from retrievers (CONQRR, ConvGQR, and IterCQR) largely advance performances on QReCC but struggle with TopiOCQA, implying that TopiOCQA is more complex and challenging than QReCC.

While baselines with GPT-3.5 show competitive performances, our 7-billion-parameter model surpass them. Our model outperforms or competes consistently against baselines that utilize the much larger QR model, GPT-3.5 (LLM IQR, LLM4CS). Moreover, despite LLM4CS relying on multiple inference calls and directly accessing extensive world knowledge of GPT-3.5, our approach achieves higher scores with a single inference from a much smaller LM[8]. This indicates that our model has effectively learned to generate rewrites that are more effective for and preferable to the retriever.

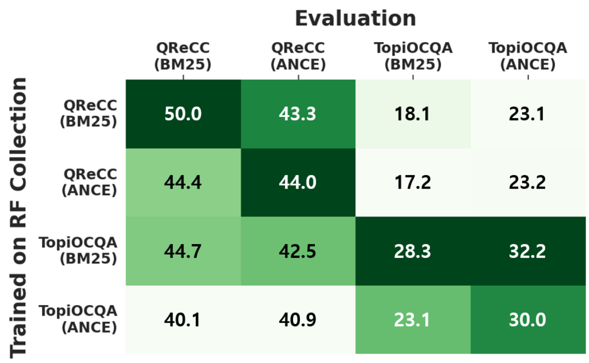

Figure 3: Heatmap of MRR scores when generalizing toward different settings. The shades are normalized per column to depict relative performance

RETPO achieves new state-of-the-art performances in most settings. Notably, for TopiOCQA, it advances the previous state-of-the-art of BM25 with a prominent gap; 11.8, 19.0, and 19.0 in MRR, R@10, and R@100, respectively. In the other benchmark and retriever type, RETPO similarly outperforms the prior best results. The only exception is R@100 scores[9] on QReCC known to exhibit the shortcut between the held-out conversation and the gold passage (Kim and Kim, 2022). It could make the token extraction method from the conversation (EDIRCS) perform better. Overall, RETPO shows a consistent improvement over other models across both sparse and dense retrieval systems. These results suggest that RETPO highlights the potential of preference-driven training in tailoring more favorable rewrites in various environments.

[8] See Appendix D for a detailed comparison with LLM4CS.

[9] We observe RETPO sacrifices R@100 score due to its tendency to produce longer and detailed rewrites. See Appendix F.1 for case study.

Query Reform.	#Qs	MRR	R@10	R@100
Dense (ANCE)				
`Original`	1	10.2	15.7	22.7
`Concat` $(H_{<t}, q_t)$	1	42.8	63.7	79.9
`Human Rewrite`	1	41.3	63.3	81.7
`+ Gold Answer`	1	57.8	79.3	90.1
GPT-4 Prompting	1	40.4	61.7	79.7
RF COLLECTION				
Ques. Rewriting	10	57.4	75.1	87.9
QR w/ Planning	10	61.7	78.9	89.8
Query Expansion	5	62.2	81.3	92.3
`Union`	25	73.6	86.8	94.5

Table 4: Effectiveness of optimal queries in RF COLLECTION. We generate a certain number (**#Qs**) of rewrites using each method and report the best retrieval performances among them.

4.2 Ablation Study

Table 2 shows ablation results for RETPO on TopiOCQA, by removing its components gradually. We start with simple baselines that prompt our teacher model GPT-4 to generate rewrites (row 1), and then fine-tune the smaller LM Llama2-7B on them (row 2). RETPO (row 3) significantly outperforms the baselines by using preference-driven optimization as useful supervision. Without DPO (row 4), the performance drops, indicating the importance of integrating the retriever's preferences for certain rewrites over others. Similarly, omitting prompting methods (Query Expansion and Planning) from RF COLLECTION (rows 5 and 6) results in degraded performance, underscoring their contribution to exploring optimal queries. The degradation across ablation clearly shows that every component of RETPO is crucial for its superior results in conversational search tasks.

4.3 Robustness to Topic Shifts in Dialogues

We report the results segmented by the question types in Table 3. We delve into the unique challenge, topic-switching, posed within TopiOCQA, where topics may abruptly change between turns. RETPO exhibits exceptional robustness in handling these topic shifts, significantly outperforming baselines. Its performances on *Topic-shifted* queries are even higher than those on *Topic-concentrated* queries, in contrast to the tendency of the baselines. [10] Additionally, RETPO boosts performance even on the context-independent queries (*first*), suggesting its potential for enhancing single-turn retrieval tasks as well.

[10] This improvement might be related to RETPO's tendency to specify details . See Appendix E.1 for detailed analysis

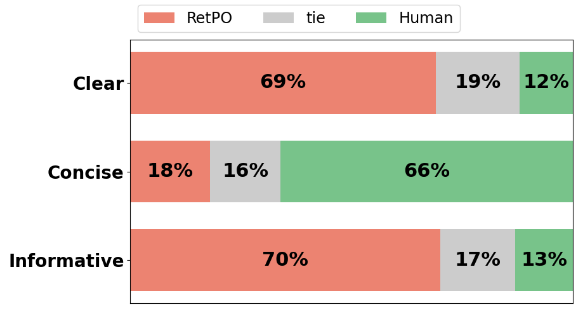

Figure 4: Pairwise evaluation with GPT-4. RETPO's rewrites are compared with the human rewrites.

4.4 Generalizing to Different Preferences

In Figure 3, we explore how well models generalize across datasets with varying scenarios. The performances along the heatmap's diagonal reveal that models typically excel when the dataset and the retriever are the same between training and evaluation, as expected. For TopiOCQA, however, we observe that the model aligned with BM25 performs better even when ANCE is used for evaluation. This might be linked to the effectiveness of query expansion strategies more favored by BM25.[11] Additionally, models generalize relatively well from TopiOCQA to QReCC, compared to the opposite direction. It again indicates that the challenges posed by TopiOCQA are more complex than QReCC. Furthermore, the results showcase the potential utility of our method to identify and select the most effective combination of strategies.

5 Analysis and Discussion

5.1 GPT-4 Evaluation

In Figure 4, we perform an automatic pairwise comparison, contrasting queries in three criteria: clarity, conciseness, and informativeness. To this end, we randomly sample 100 examples in the validation set and leverage a superior LLM, GPT-4 (OpenAI, 2023), as a judge. For the same input conversation, we pair query rewrites from a human and RETPO for comparison. Figure 4 indicates that RETPO typically generates question rewrites that are more informative and less ambiguous compared to human rewrites, though they are less concise. The extended rewrites from RETPO, despite sacrificing conciseness, contain valuable details, leading to superior performance. We observe a similar tendency for RF COLLECTION.[12]

[11] See Appendix E.1 for detailed analysis

[12] More details are in Appendix E.2 and Appendix E.3.

Model	LM	Retriever	TopiOCQA	QReCC	CAsT-19	CAsT-20
With Fine-tuning of Retriever						
ConvDR	-	ANCE	26.4	35.7	43.9	32.4
Conv-ANCE	-	ANCE	20.5	45.6	34.1	27.5
HAConvDR	-	ANCE	28.5	45.6	-	-
LeCoRE	-	SPLADE	31.4	<u>48.5</u>	42.2	29.0
InstructoR	GPT-3.5	ANCE	23.7	40.5	46.6	29.6
	GPT-3.5	Contriever	**37.0**	**50.4**	**55.1**	32.8
Without Fine-tuning of Retriever						
T5QR	T5-base	ANCE	22.2	31.8	41.7	29.9
ConvGQR	T5-base	ANCE	24.3	39.1	43.4	33.1
LLM4CS (*Single*)	GPT-3.5	ANCE	21.2	30.8	43.1	35.7
LLM4CS (*Ensemble*)	GPT-3.5	ANCE	31.1	39.1	<u>51.5</u>	**45.5**
RETPO$_{BM25}$ (*Ours*)						
	Llama2-7B	ANCE	31.1	40.4	44.8	36.9
	Llama2-7B	Contriever	<u>33.7</u>	45.7	42.8	<u>41.1</u>

Table 5: Evaluation results of our method and the baselines that fine-tunes the retrievers in NDCG@3 scores. **The best scores** are in bold and <u>the second-best scores</u> are underlined. We use the rewrites aligned with BM25 feedback from RETPO$_{BM25}$.

Method	MRR	QReCC R@10	R@100
`Human Rewrite`	41.3	63.3	81.7
RF COLLECTION	73.6	86.8	94.5
T5QR (Lin et al., 2020)	34.5	53.1	72.8
Fine-tuned T5			
w/ `Human Rewrite`	35.5	55.4	73.8
w/ RF COLLECTION	39.2	59.9	78.2

Table 6: Evaluation results of a small language model (T5-base) on the QReCC benchmark. We fine-tune T5-base using two datasets—`Human Rewrite` and RF COLLECTION—and compare the retrieval performance based on the rewrites generated by each model.

5.2 RF COLLECTION with Small LM

Table 6 presents experimental results with a small LM, T5-base to assess the effectiveness of RF COLLECTION beyond our proposed method. To examine its utility as a general training resource, we fine-tune a non-LLM model, T5-base, on RF COLLECTION and evaluate its performance in retrieval tasks by using the dense retriever ANCE. Notably, T5 models trained on RF COLLECTION achieve substantial improvements over those trained on `Human Rewrite`, surpassing both the T5QR baseline and our reproduced scores. These findings underscore RF COLLECTION as a valuable resource for query reformulation, extending its impact beyond large-scale LLM-based methods and reinforcing its role as a standalone contribution to the field.

5.3 Fine-tuning Retriever or Rewriter?

Table 5 compares our method with the baselines that fine-tune retrievers in in-domain (QreCC, TopiOCQA, and CAsT-19) and out-of-domain (CAsT-20) scenarios. Fine-tuning retrievers generally yields good in-domain performance but the performance tends to be highly sensitive to retrievers. For example, InstructoR is effective for fine-tuning Contriever but its effectiveness drops substantially for a different retriever ANCE, showing the instability of the method. This highlights the difficulty of fine-tuning retrievers, which requires sophisticated engineering and retriever-specific optimization.

Our method consistently enhances the performance of off-the-shelf retrievers without additional feedback for the target retrievers. By utilizing rewrites aligned with BM25, RETPO$_{BM25}$ surpasses existing baselines except for InstructoR on TopiOCQA. It particularly shows superior effectiveness in the *zero-shot* scenario. On the CAsT-20 dataset, which lacks a training set, RETPO$_{BM25}$ trained on TopiOCQA successfully generalizes to the unseen dataset, outperforming all baselines except for LLM4CS which requires multiple LLM inferences[13]. Our findings implicate that RETPO is easy to deploy and shows competitive performance across diverse scenarios.

[13] See Appendix D for a detailed comparison with LLM4CS.

6 Related Works

Conversational Search Conversational search is the precedent task of open-domain conversational QA and several benchmarks are released (Qu et al., 2020; Dalton et al., 2020). A line of studies proposes to fine-tune the dense retriever, enabling it to encode conversational context. Most studies follow the approach (Lin et al., 2021b; Kim et al., 2022; Mao et al., 2022; Ma et al., 2023; Huang et al., 2023; Mo et al., 2024b,c; Chen et al., 2024). Concurrently, Mao et al. (2024) proposes to use LLM as a retrieval backbone and achieve superior performance in the task. Although they show the dominant performances in the task, they require retriever-specific engineering.

Query Reformulation Recent studies prompt LMs to provide detailed information such as the expected document (Wang et al., 2023; Jagerman et al., 2023). The recent study propose to use reward signals to optimize the QR model (Ma et al., 2023). In conversational search, Anantha et al. (2021) introduces the *rewrite-then-retrieve* pipeline to handle the conversational dependency. Most studies fine-tune QR models to generate the standalone question (Voskarides et al., 2020; Lin et al., 2021c; Kumar and Callan, 2020). Recently, commercial models have been employed for QR tasks, leveraging their robust generative capabilities and extensive world knowledge (Ye et al., 2023; Mao et al., 2023b; Mo et al., 2024a). In contrast, RETPO is the first to utilize preference-driven optimization for reformulating queries in conversational search.

Aligning Language Models with Feedback Studies on LLM alignment utilize human feedback (Bai et al., 2022a; Ouyang et al., 2022; Rafailov et al., 2023). Recently, AI feedback is also actively explored as an alternative to human feedback (Bai et al., 2022b; Sun et al., 2023). Kim et al. (2023) automatically construct synthetic feedback, leveraging prior knowledge, instead of collecting feedback. Tian et al. (2023) obtain synthetic feedback utilizing truthfulness measurements like FactScore (Min et al., 2023). Our method is similar to these studies in that it includes the synthetic dataset construction; however, we focus on a specific target task, question rewriting, and reflecting a target retriever's feedback. Concurrently, to reduce the reliance on labeled data for aligning QR models, Zhang et al. (2024) utilize marginalized rewards derived from conversation answers, while Lai et al. (2024) leverage the LM as a reference-free evaluator to align retrievers with only a small set of training data.

7 Conclusion

Our paper introduces RETPO, a framework for optimizing an LM to generate retriever-preferred query rewrites. Utilizing the LLM-based process, we construct and release a large-scale dataset RF COLLECTION. Based on it, we enhance an open-source LM, significantly outperforming *rewrite-then-retrieve* baselines on two recent benchmarks QReCC and TopiOCQA. Our work, which pioneers preference-driven optimization in query reformulation advances conversational search performance and shows promising results in generalization.

Limitation

One limitation of our study is the exclusive focus on larger-scale language models. Consequently, our model tends to generate longer queries rich in specific information and keywords, possibly relying on the emergent abilities of large LMs, which we leverage to boost performance. However, exploring smaller-scale LMs could offer insights into the scalability and efficiency of our approach.

Additionally, due to budget constraints, we utilized only half of the TopiOCQA training set. Access to the full dataset could potentially yield further improvements in model performance.

Our framework has been tested solely within the realm of conversational search, yet its application is not limited to this task. Future research could adapt our framework to a broader range of tasks and domains, potentially enhancing its utility and impact.

While we employed three prompting methods, there is a vast landscape of alternative approaches that we did not explore. Future studies could investigate additional prompting strategies tailored to specific tasks and retriever systems.

Finally, pairing our method with more advanced retrieval systems presents a promising avenue for research. Despite the clarity and consistency of the generated queries, we noted instances of retrieval failure, indicating that there is room for improvement in retriever performance, which could, in turn, further enhance the overall efficacy of our method.

Acknowledgement

This work was supported by the National Research Foundation of Korea [NRF2023R1A2C3004176, RS-2023-00262002], the Ministry of Health & Welfare, Republic of Korea [HR20C0021], and the ICT Creative Consilience program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [IITP-2025-20200-01819]. This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00414981) and Institute of Information & communications Technology Planning & Evaluation (IITP) under the Leading Generative AI Human Resources Development (IITP-2025-RS-2024-00397085) grant funded by the Korea government (MSIT).

References

Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2022. Topiocqa: Open-domain conversational question answering with topic switching. *Transactions of the Association for Computational Linguistics*, 10:468–483.

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-domain question answering goes conversational via question rewriting. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 520–534.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Haonan Chen, Zhicheng Dou, Kelong Mao, Jiongnan Liu, and Ziliang Zhao. 2024. Generalizing conversational dense retrieval via LLM-cognition data augmentation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2700–2718, Bangkok, Thailand. Association for Computational Linguistics.

Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2021. Cast 2020: The conversational assistance track overview. In *In Proceedings of TREC*.

Jeffrey Dalton, Chenyan Xiong, Vaibhav Kumar, and Jamie Callan. 2020. Cast-19: A dataset for conversational information seeking. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1985–1988.

Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can you unpack that? learning to rewrite questions-in-context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5918–5924.

Chao-Wei Huang, Chen-Yu Hsu, Tsu-Yuan Hsu, Chen-An Li, and Yun-Nung Chen. 2023. CONVERSER: Few-shot conversational dense retrieval with synthetic data generation. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 381–387, Prague, Czechia. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. *arXiv preprint arXiv:2305.03653*.

Yunah Jang, Kang-il Lee, Hyunkyung Bae, Seungpil Won, Hwanhee Lee, and Kyomin Jung. 2023. Itercqr: Iterative conversational query reformulation without human supervision. *arXiv preprint arXiv:2311.09820*.

Zhuoran Jin, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2023. InstructoR: Instructing unsupervised conversational dense retrieval with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6649–6675, Singapore. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

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 (Volume 1: Long Papers)*, pages 1601–1611.

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. *arXiv preprint arXiv:2212.14024*.

Gangwoo Kim, Sungdong Kim, Kang Min Yoo, and Jaewoo Kang. 2022. Generating information-seeking conversations from unlabeled documents. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2362–2378.

Sungdong Kim, Sanghwan Bae, Jamin Shin, Soyoung Kang, Donghyun Kwak, Kang Min Yoo, and Minjoon Seo. 2023. Aligning large language models through synthetic feedback. *arXiv preprint arXiv:2305.13735*.

Sungdong Kim and Gangwoo Kim. 2022. Saving dense retriever from shortcut dependency in conversational search. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10278–10287.

Vaibhav Kumar and Jamie Callan. 2020. Making information seeking easier: An improved pipeline for conversational search. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3971–3980.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Yilong Lai, Jialong Wu, Congzhi Zhang, Haowen Sun, and Deyu Zhou. 2024. Adacqr: Enhancing query reformulation for conversational search via sparse and dense retrieval alignment. *arXiv preprint arXiv:2407.01965*.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021a. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2356–2362, New York, NY, USA. Association for Computing Machinery.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021b. Contextualized query embeddings for conversational search. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1004–1015.

Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020. Conversational question reformulation via sequence-to-sequence architectures and pretrained language models. *arXiv preprint arXiv:2004.01909*.

Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2021c. Multi-stage conversational passage retrieval: An approach to fusing term importance estimation and neural query rewriting. *ACM Transactions on Information Systems (TOIS)*, 39(4):1–29.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

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*.

Kelong Mao, Chenlong Deng, Haonan Chen, Fengran Mo, Zheng Liu, Tetsuya Sakai, and Zhicheng Dou. 2024. Chatretriever: Adapting large language models for generalized and robust conversational dense retrieval. *arXiv preprint arXiv:2404.13556*.

Kelong Mao, Zhicheng Dou, Bang Liu, Hongjin Qian, Fengran Mo, Xiangli Wu, Xiaohua Cheng, and Zhao Cao. 2023a. Search-oriented conversational query editing. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4160–4172.

Kelong Mao, Zhicheng Dou, Fengran Mo, Jiewen Hou, Haonan Chen, and Hongjin Qian. 2023b. Large language models know your contextual search intent: A prompting framework for conversational search. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1211–1225, Singapore. Association for Computational Linguistics.

Kelong Mao, Zhicheng Dou, Hongjin Qian, Fengran Mo, Xiaohua Cheng, and Zhao Cao. 2022. Convtrans: Transforming web search sessions for conversational dense retrieval. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2935–2946.

Kelong Mao, Hongjin Qian, Fengran Mo, Zhicheng Dou, Bang Liu, Xiaohua Cheng, and Zhao Cao. 2023c. Learning denoised and interpretable session representation for conversational search. In *Proceedings of the ACM Web Conference 2023*, pages 3193–3202.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4089–4100.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.

Fengran Mo, Abbas Ghaddar, Kelong Mao, Mehdi Rezagholizadeh, Boxing Chen, Qun Liu, and Jian-Yun Nie. 2024a. Chiq: Contextual history enhancement for improving query rewriting in conversational search. *arXiv preprint arXiv:2406.05013*.

Fengran Mo, Kelong Mao, Yutao Zhu, Yihong Wu, Kaiyu Huang, and Jian-Yun Nie. 2023. Convgqr: Generative query reformulation for conversational search. *arXiv preprint arXiv:2305.15645*.

Fengran Mo, Chen Qu, Kelong Mao, Tianyu Zhu, Zhan Su, Kaiyu Huang, and Jian-Yun Nie. 2024b. History-aware conversational dense retrieval. *arXiv preprint arXiv:2401.16659*.

Fengran Mo, Bole Yi, Kelong Mao, Chen Qu, Kaiyu Huang, and Jian-Yun Nie. 2024c. Convsdg: Session data generation for conversational search. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 1634–1642.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. Scispacy: fast and robust models for biomedical natural language processing. *arXiv preprint arXiv:1902.07669*.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W Bruce Croft, and Mohit Iyyer. 2020. Open-retrieval conversational question answering. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 539–548.

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. *arXiv preprint arXiv:2305.18290*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Zhiqing Sun, Yikang Shen, Hongxin Zhang, Qinhong Zhou, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. 2023. Salmon: Self-alignment with principle-following reward models. *arXiv preprint arXiv:2310.05910*.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. 2023. Fine-tuning language models for factuality. *arXiv preprint arXiv:2311.08401*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2021. Question rewriting for conversational question answering. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 355–363.

Christophe Van Gysel and Maarten de Rijke. 2018. Pytrec_eval: An extremely fast python interface to trec_eval. In *SIGIR*. ACM.

Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020. Query resolution for conversational search with limited supervision. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval (SIGIR)*, pages 921–930.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. *arXiv preprint arXiv:2303.07678*.

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

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*, pages 10000–10014.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

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

In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-shot conversational dense retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on research and development in information retrieval*, pages 829–838.

Tianhua Zhang, Kun Li, Hongyin Luo, Xixin Wu, James Glass, and Helen Meng. 2024. Adaptive query rewriting: Aligning rewriters through marginal probability of conversational answers. *arXiv preprint arXiv:2406.10991*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Dataset	Train	RF COLLECTION		
QReCC		QR	Plan	QE
# Dialogues	10,823	8,987	5,519	8,987
# Turns	63,501	29,596	8,817	29,596
TopiOCQA		QR	Plan	QE
# Dialogues	3,509	3,508	3,429	3,508
# Turns	45,450	24,283	13,845	24,283

Table 7: Statistics of RF COLLECTION, QReCC, and TopiOCQA.

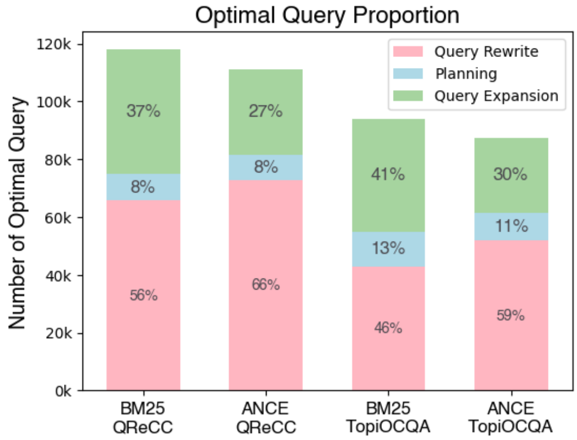

Figure 5: Proportion of optimal queries generated by each prompting method.

A Datasets

The training dataset of QReCC comprises 10,823 conversations encompassing 63,501 turns. For evaluating queries and gathering feedback from retrieval systems, we exclude turns with no gold passage label, yielding a dataset with 8,987 conversations and 29,596 turns.

TopiOCQA consists of 3,509 conversations with 45,450 turns. Unlike QReCC where we fully utilize the dataset, we conduct our method on a subset of TopiOCQA to manage costs associated with API requests, resulting in 3,429 conversations with 13,845 turns. Specifically, for the QR with planning prompting method, we only apply the method to turns where the number of optimal queries generated from the QR method is less than three.

B RF COLLECTION Details

When constructing the collection of optimal queries $\mathcal{C}_*$, we only choose rewrites whose rank is higher than 30. For the collection of binarized comparisons, we only consider the query with a rank higher than 50 as the preferred query. We do not pair the queries with the same rank.

B.1 Proportion of Question Types

To obtain statistics in Sec. E.1, we use the following process. Employing the NLTK (Loper and Bird, 2002) module for query processing, part-of-speech tagging was executed, and unseen nouns and adjectives were identified through the comparison of words in the conversational history by string matching. Queries commencing with 'what,' 'why,' 'where,' 'when,' and 'who' were categorized as Start with "*Wh*" queries Furthermore, for the categorization of queries into the query expansion style, the proportion of queries containing multiple sentences was calculated by Spacy (Neumann et al., 2019) library.

In Figure 5, we show the proportion of query rewrite method preferences exhibited by a sparse retriever and a dense retriever on QReCC and TopiOCQA. In the case of RF COLLECTION made with feedback from BM25, It is observable that the proportion integrating the query expansion surpasses that derived from feedback by ANCE. Moreover, within the RF-COLLECTION tailored for TopiOCQA, there is an observed elevation in the number of queries generated through the query expansion and planning method in comparison to those generated from QReCC. This tendency implies the elevated complexity inherent in TopiOCQA compared to QReCC-like topic-shifting. The rationale behind the relatively diminished overall proportion of planning lies in its role as an auxiliary method for Query Rewrite, as previously mentioned.

C Experimental Details

Implementation Detail For BM25, we set $k_1 = 0.82$, $b = 0.68$ in QReCC, and $k_1 = 0.9$, $b = 0.4$ in TopiOCQA, respectively, where k_1 controls the non-linear term frequency normalization and b is the scale of the inverse document frequency. We utilize GPT4-Turbo (gpt-4-1106-preview) via the OpenAI API to produce query candidates from contextualized questions. We use default hyperparameters of chat completion of API except for setting a temperature of 0.7 and maximum tokens as 1000. For each prompting method (Question Rewriting, Planning, Query Expansion), we generate 10, 10, and 5 candidates respectively. We use Faiss (Johnson et al., 2019) and Pyserini (Lin et al., 2021a) for efficient search across large passage indices. We retrieve top-100 relevant passages for each query can-

Query Reform.		Count		TopiOCQA			QReCC		
		#{Calls}	#{Queries}	MRR	NDCG	R@10	MRR	NDCG	R@10
REW	MaxProb	**1**	**1**	22.6	21.2	40.1	33.2	30.8	50.3
	Mean	5	5	23.7	22.4	42.1	34.1	31.4	51.6
	SC	5	5	23.0	21.9	40.5	33.5	31.1	51.6
RTR	MaxProb	2	2	28.9	27.8	46.4	38.2	35.7	57.3
	Mean	6	6	30.9	29.8	49.0	39.3	36.6	60.2
	SC	6	6	29.0	27.9	47.4	38.1	35.7	57.6
RAR	MaxProb	**1**	2	30.0	30.9	49.3	40.5	37.8	60.8
	Mean	5	10	32.0	**31.1**	50.5	41.8	39.1	62.3
	SC	5	10	31.1	29.9	50.0	40.6	38.0	60.8
RetPO (*Ours*)	-	**1**	**1**	**32.2**	**31.1**	**51.6**	**44.0**	**41.1**	**66.7**

Table 8: Performance comparison against LLM4CS with different aggregation methods across TopiOCQA and QReCC datasets. We evaluate it on the ANCE retriever.

didate and obtain rank using pytrec_eval (Van Gysel and de Rijke, 2018). Following (Kim and Kim, 2022), the maximum token length is constrained to 128 tokens for query representations and 384 tokens for passage representations.

We largely follow the Huggingface repository, Alignment Handbook.[14] We use Llama2-7b-hf as our backbone. We use eight A100 GPUs (80GB) to train the Llama2-7b. It is trained in one epoch for supervised fine-tuning. We set the learning rate as 2e-5, and the batch size as 20 per GPU. The warmup ratio is set to 0.1 and we use torch data type `bfloat16`. For the training of DPO, we set the beta as 0.1, and the maximum length as 1024. We train our model in three epochs with a batch size of 8 per GPU. We set the maximum input context length as 2048 and the output length as 200.

D Comparison with LLM4CS

Table 8 presents a detailed performance comparison of LLM4CS (Mao et al., 2023b). We utilize `gpt-3.5-turbo-16k` as the base model, maintaining the same few-shot demonstrations as used by Mao et al. (2023b). The results indicate that RETPO consistently outperforms all combinations of prompting strategies (REW, RTR, RAR) and aggregation methods (MaxProb, Mean, SC) employed by LLM4CS, despite using only a single inference to generate high-quality queries. This demonstrates that our approach effectively tailors query rewrites to the retriever, achieving better performance without the need for multiple inferences.

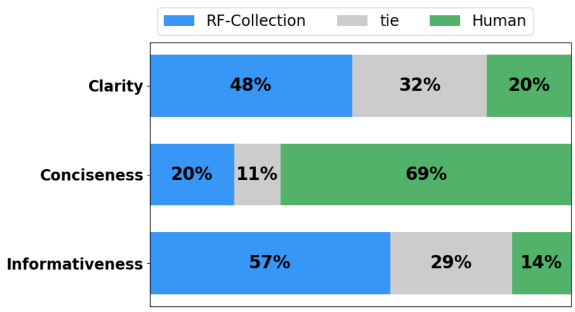

Figure 6: Pairwise evaluation with GPT-4. Rewrites from RF COLLECTION are compared with the human rewrites.

E Analysis Details

E.1 Comparison of Question Distributions

Table 10 presents a statistical analysis of query distributions of optimal queries from RF COLLECTION $\mathcal{C}_*$ and predicted rewrites from RetPO methods. It shows the number of words, frequency of unseen words from the held-out conversation, questions starting with '*Wh-*' words, and those composed of multiple sentences.[15] RF COLLECTION and RETPO tend to create longer queries, often extending to 2-5 times the length of the original one, which includes a number of words unseen within the utterances so far. The query expansion (QE) notably alters the question structure, frequently constructing them as multi-sentence entities (high % of Multiple Sents.). This method tends to prepend a pseudo-answer to the question (low % of Starting with '*Wh-*'). RETPO, in contrast, strikes a balance between QR and QE, achieving a midpoint depending on the retriever type.

[14] https://github.com/huggingface/alignment-handbook
[15] See Appendix E for details about the measurements.

Ret.	Trained on	Preference	QReCC MRR	QReCC R@10	QReCC R@100	TopiOCQA MRR	TopiOCQA R@10	TopiOCQA R@100
BM25	OQF-QReCC	BM25	50.0	69.5	89.5	18.1	31.9	58.7
		ANCE	44.4	66.7	90.0	17.2	32.0	59.1
	OQF-TopiOCQA	BM25	44.7	66.8	89.3	28.3	48.3	73.1
		ANCE	40.1	62.2	86.5	23.1	41.3	69.4
ANCE	OQF-QReCC	BM25	43.3	65.0	82.5	23.1	39.3	58.3
		ANCE	44.0	66.7	84.6	23.2	40.0	59.4
	OQF-TopiOCQA	BM25	42.5	63.5	81.6	32.2	51.6	69.5
		ANCE	40.9	61.9	79.9	30.0	49.6	68.7

Table 9: Retrieval performance when generalizing toward different setups.

	Orig.	RF $\mathcal{C}_*$ QR	RF $\mathcal{C}_*$ QE	RETPO Spar.	RETPO Den.
# Words	6.9	11.3	30.0	22.4	15.9
# Unseen Words	0.0	2.2	7.7	4.9	3.0
% Start with '*Wh*'	62.1	63.5	0.03	12.8	28.6
% Multiple Sents.	0.08	0.2	99.4	59.8	27.7

Table 10: Statistics for question distributions from RF COLLECTION and RETPO. We compare the number of words and the structure of questions.

E.2 GPT-4 Evaluation Details

Prompts used in GPT-4 evaluation are shown in Table 12, 13, and 14. Considering the position bias in GPT-4 evaluation (Zheng et al., 2023), we assess the same instance twice, reversing the order of the two rewritten questions. Also, we regard the comparison as a 'Tie' if the two evaluation results conflict with each other.

E.3 Evaluating RF COLLECTION

In Table 6, we present a comprehensive comparison of various query reformulation strategies. We assess the performance of rewrites generated from our RF COLLECTION against baselines including oracle setups. We report the best retrieval performances of each set. All of our prompting methods significantly outperform `Human Rewrite` with a huge gap in most metrics. Query expansion shows the best performance among the prompting methods, showing its efficacy in adding keywords. The combined set `Union` of all strategies yields the best results, indicating these methods are mutually beneficial.

F Case Study

In Table 15, we demonstrate the effectiveness of RETPO in enhancing retrieval performance by providing additional specific information. While the information generated by RETPO does not seemingly overlap with the actual answer, they nevertheless contribute by offering supplementary cues that guide the retriever toward the most pertinent passages.

Query Reform.	#(Q)	MRR	R@10	R@100
Sparse (BM25)				
`Original`	1	6.5	11.1	21.5
`Concat` $(H_{<t}, q_t)$	1	47.0	65.1	82.8
`Human Rewrite`	1	40.0	62.7	98.5
`+ Gold Answer`	1	92.4	97.2	99.7
RF COLLECTION				
Query Rewriting	10	64.5	81.1	94.5
w/ Planning	10	68.2	83.6	95.2
Query Expansion	5	75.0	91.3	99.1
`Union`	25	85.1	93.7	98.6

Table 11: Comparison of effectiveness with BM25 over different query reformulation strategies. We evaluate the performance of our generated rewrites from RF COLLECTION against simple baselines and oracle setups.

F.1 Over-specification Issue

In Table 16, we present a failure case where RETPO fails to accurately align with the original search intent, resulting in a misjudgment during retrieval. The deviation from the original question scope is highlighted, indicating an over-specification in the output query. This over-specification leads to a mismatch with the intended search query, thereby hindering successful retrieval.

G Prompts

Table 17, 18, 19 illustrate examples of our prompting methods: question rewriting (QR), QR with planning, and query expansion. Following (Khattab et al., 2022), each prompt comprises four components: an instruction, a format specification, a few-shot example, and a test instance. In question

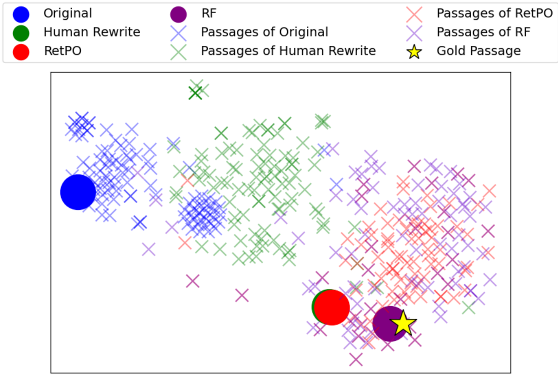

Figure 7: T-SNE visualization of ANCE embeddings from RETPO and RF COLLECTION. Queries and passages from the same method are colored identically.

rewriting, we instruct LLM to generate a series of decontextualized questions adhering to the predefined criteria proposed by (Ye et al., 2023). In QR with planning, the LLM is guided to elicit relevant information that might help reformulate a question, before generating each rewritten question. In query expansion, LLM produces a set of pseudo-answer candidates expected to align closely with the potential response of the question. We use a one-shot example for each prompting method to demonstrate the desired action and output.

[Instruction] Please act as an impartial judge and evaluate the quality of the query-rewriting system displayed below. The system tries to rewrite the conversational input to a stand-alone question, eliminating dependency on the conversational context. Your job is to compare the **clarity** of the two rewritten stand-alone questions. That is, You should check which question is **less open to multiple interpretations and has a more clear intention.** Please choose either 'A' or 'B'. If the two questions show the same clarity, answer it by 'Tie'. For example, Judge: (A|B|Tie) [Conversation] {*conversation*} [The Start of stand-alone question A] {*query*_1} [The End of stand-alone question A] [The Start of stand-alone question B] {*query_2*} [The End of stand-alone question B] Judge:

Table 12: GPT4 prompt for evaluating clarity

[Instruction] Please act as an impartial judge and evaluate the quality of the query-rewriting system displayed below. The system tries to rewrite the conversational input to a stand-alone question, eliminating dependency on the conversational context. Your job is to compare the **conciseness** of the two rewritten stand-alone questions. That is, You should check which question is **more brief and directly states the search intent without additional elaboration.** Please choose either 'A' or 'B'. If the two questions show the same conciseness, answer it by 'Tie'. For example, Judge: (A|B|Tie) [Conversation] {*conversation*} [The Start of stand-alone question A] {*query*_1} [The End of stand-alone question A] [The Start of stand-alone question B] {*query_2*} [The End of stand-alone question B] Judge:

Table 13: GPT4 prompt for evaluating conciseness

[Instruction]
Please act as an impartial judge and evaluate the quality of the query-rewriting system displayed below. The system tries to rewrite the conversational input to a stand-alone question, eliminating dependency on the conversational context.

Your job is to compare the **informativeness** of the two rewritten stand-alone questions.
That is, You should check **which question provides more useful and relevant information.**
Please choose either 'A' or 'B'. If the two questions show the same informativeness, answer it by 'Tie'.
For example, Judge: (A|B|Tie)

[Conversation]
{*conversation*}

[The Start of stand-alone question A]
{*query*_1}
[The End of stand-alone question A]

[The Start of stand-alone question B]
{*query_2*}
[The End of stand-alone question B]

Judge:

Table 14: GPT4 prompt for evaluating informativeness

Conversation:
Q1: where are we now video who is the girl A1: The Where Are We Now? music video, directed by Tony Oursler, shows Bowie and an unnamed female companion. The woman was later confirmed as artist Jacqueline Humphries, Oursler's wife.
Q2: why was the wife in the music video A2: David Bowie and Oursler reportedly wanted someone who looked like Bowie's PA, Corinne Coco Schwab, as she did in the 1970s in Berlin.
Q3: where is the setting of the music video A3: The video is set in what could be an artists' studio in Berlin, where Bowie lived from 1976, showing moving black-and-white footage of the city from the 1970s.
Q4: any references in the music video A4: The footage on the screen and references in the lyrics include the Berlin Wall and mention of the Bosebrucke, the first border crossing that opened when the Wall fell.

Original: any other references (**rank: Not Found**)

Human Rewrite: any other references in the where are we now music video besides the berlin wall and bosebrucke (**rank: Not Found**)

RETPO: The video may also include references to Bowie's time in Berlin during the 1970s, such as the black-and-white footage of the city and the mention of the Bosebrucke.\nAny other references in the Where Are We Now music video besides the Berlin Wall and mention of the Bosebrucke (**rank: 1**)

Answer: Other reference points in the video and song are the Brandenburg Gate the Dschungel nightclub; the Fernsehturm, or television tower KaDeWe, the department store and graffiti from Kunsthaus Tacheles.

Table 15: Success case on QReCC (2396_5) when using BM25.

Conversation:
Q1: what is mechanical animals? A1: Mechanical Animals is the third studio album by American rock band Marilyn Manson.
Q2: what kind of music did this album have? A2: A rock opera and concept album, Mechanical Animals is the second installment in a trilogy
Q3: who producedthe album? A3: Marilyn Manson subsequently employed Michael Beinhorn as principal producer, co-producing the record with Marilyn Manson. Sean Beavan was also brought in to supply additional production work.
Q4: was this album a hit on the charts? A4: Mechanical Animals debuted at number one on the Billboard 200 with first-week sales of 223,000 units, becoming Marilyn Manson's first number-one album on the chart.
Q5: did it win any awards? A5: Mechanical Animals was certified platinum by the Recording Industry Association of America on February 25, 1999,
Q6: did they go on tour? A6: The Beautiful Monsters Tour was a North American concert tour co-headlined by American rock bands Hole and Marilyn Manson.

Original: who was hole? (**rank: Not Found**)

Human Rewrite: Who was American rock band Hole? (**rank: 3**)

RETPO: Who were the members of the band Hole that co-headlined the Beautiful Monsters Tour with Marilyn Manson? (**rank: Not Found**)

Answer: Hole was an American alternative rock band formed in Los Angeles, California in 1989.

Table 16: Failure case in QReCC (1321_7) when using BM25. The red text indicates the deviation from the original question scope. The resulting query from RETPO over-specifies irrelevant details, asking about members of the band Hole, rather than the band as a whole. It leads to misalignment with the original search intent.

Given a question and its context, decontextualize the question by addressing coreference and omission issues. The resulting question should retain its original meaning and be as informative as possible, and should not duplicate any previously asked questions in the context. Please give me a list of 10 candidates for the rewrite. Here are some examples.

- - -

Follow the following format.

Conversation:
${conversational context for the question}

Question: ${follow-up question to be rewritten}

Rewrite: ${list of 10 rewritten question candidates, each on a new line.}
Rewrite i: ${(i)-th rewritten question that address coreference and omission issues}

- - -

Conversation:
Q1: How did religion effect their society? A1: Religion held ancient Hawaiian society together, affecting habits, lifestyles, work methods, social policy and law. The legal system was based on religious kapu, or taboos.
Q2: What is Kapu? A2: Kapu is the ancient Hawaiian code of conduct of laws and regulations.
...
Q4: What are the beginnings of the kapu system like? A4: The rigidity of the kapu system might have come from a second wave of migrations in 1000–1300 from which different religions and systems were shared

Question: How did this wave effect society or the system?

Rewrite:
```
Rewrite 1: How did the second wave of migrations between 1000-1300 impact ancient
Hawaiian society or the kapu system?
Rewrite 2: In what ways were the social structure or kapu system of ancient Hawaii
influenced by migrations from 1000 to 1300?
Rewrite 3: ...
```

Table 17: Prompt for the question rewriting method

I am working on finding information to rewrite the question. Given a question and its context, Please provide 10 information-Rewrite pairs, where each pair consists of information that might be needed to answer the question and a rewritten question. the rewritten question is a decontextualized version of the question by addressing coreference and omission issues with respect to each information. the resulting question should retain its original search intent. Here are some examples.

- - -

Follow the following format.

Conversation:
${conversational context for the question}

Question: ${follow-up question to be rewritten}

Information-Rewrite: ${list of 10 Information-Rewrite pairs, each on a new line}
Info i: ${(i)-th information that is needed to answer the question. it should not be too specific}
Rewrite i: ${(i)-th rewritten question that address coreference and omission issues with respect to (i)-th information.}

- - -

Conversation:
Q1: How did religion effect their society? A1: Religion held ancient Hawaiian society together, affecting habits, lifestyles, work methods, social policy and law. The legal system was based on religious kapu, or taboos.
Q2: What is Kapu? A2: Kapu is the ancient Hawaiian code of conduct of laws and regulations.
...
Q4: What are the beginnings of the kapu system like? A4: The rigidity of the kapu system might have come from a second wave of migrations in 1000–1300 from which different religions and systems were shared

Question: How did this wave effect society or the system?

Information-Rewrite:
`Info 1: Migration Impact - Information about how the second wave of migrations influenced the existing societal structures or introduced changes in ancient Hawaiian society.`
`Rewrite 1: How did the second wave of migrations around 1000-1300 AD affect ancient Hawaiian society and its structures?`

`Info 2: Changes to Kapu System - Details regarding any modifications or introductions to the kapu system as a result of the second wave of migrations.`
`Rewrite 2: What changes were made to the ancient Hawaiian kapu system due to the second wave of migrations?`

`Info 3: ...`

Table 18: Prompt for the planning method.

Please give me a list of 5 answer candidates based on the given conversation context and question. Here are some examples.

- - -

Follow the following format.

Conversation:
${conversational context for the question}

Question: ${follow-up question to be rewritten}

Answer: ${list of 5 answer candidates, each on a new line.}
Answer i: ${(i)-th answer for the current question}

- - -

Conversation:
Q1: How did religion effect their society? A1: Religion held ancient Hawaiian society together, affecting habits, lifestyles, work methods, social policy and law. The legal system was based on religious kapu, or taboos.
Q2: What is Kapu? A2: Kapu is the ancient Hawaiian code of conduct of laws and regulations.
...
Q4: What are the beginnings of the kapu system like? A4: The rigidity of the kapu system might have come from a second wave of migrations in 1000–1300 from which different religions and systems were shared

Question: How did this wave effect society or the system?

Answer:
```
Answer 1: The second wave of migrations brought new religious beliefs and practices, which likely intensified the existing kapu system and introduced additional taboos.
Answer 2: The influx of migrants during this period could have led to the formalization and expansion of the kapu system, as new ideas were integrated and enforced.
Answer 3: ...
```

Table 19: Prompt for the query expansion method. We concatenate the pseudo-answers with a self-contained query.