ICR: Iterative Clarification and Rewriting for Conversational Search

Zhiyu Cao, Peifeng Li[*], Qiaoming Zhu
School of Computer Science and Technology, Soochow University, Suzhou, China
`zycao18@stu.suda.edu.cn`, `{pfli, qmzhu}@suda.edu.cn`

Abstract

Most previous work on Conversational Query Rewriting employs an end-to-end rewriting paradigm. However, this approach is hindered by the issue of multiple fuzzy expressions within the query, which complicates the simultaneous identification and rewriting of multiple positions. To address this issue, we propose a novel framework ICR (Iterative Clarification and Rewriting), an iterative rewriting scheme that pivots on clarification questions. Within this framework, the model alternates between generating clarification questions and rewritten queries. The experimental results show that our ICR can continuously improve retrieval performance in the clarification-rewriting iterative process, thereby achieving state-of-the-art performance on two popular datasets.

1 Introduction

In conversational question answering, users' questions often require the assistance of external knowledge. Conversational search is designed to provide users with external information needs in multi-turn conversation. However, users often omit some content in their queries for the sake of simplification during the conversation. Conversational Query Rewriting (CQR) is a key step in conversational search, aiming to rewrite vague queries in conversation into de-contextualized queries, thereby promoting conversational search. As shown in Figure 1, given the dialogue history (i.e., Q_1 to A_2) and the current query Q_3, the goal of CQR is to rewrite Q_3 into a more complete query (e.g., "What was the purpose of the space shuttle program operated from 1981 to 2011 by NASA?") and use this rewritten query to retrieve relevant passages.

Early studies on CQR used manually labeled rewritten queries as ground truth to train the models (Lin et al., 2020; Wu et al., 2022; Mao et al.,

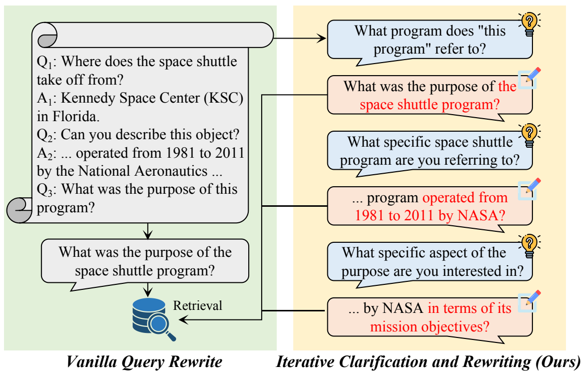

Q1: Where does the space shuttle take off from?
A1: Kennedy Space Center (KSC) in Florida.
Q2: Can you describe this object?
A2: ... operated from 1981 to 2011 by the National Aeronautics ...
Q3: What was the purpose of this program?
What was the purpose of the space shuttle program?
Retrieval
What program does "this program" refer to?
What was the purpose of the space shuttle program?
What specific space shuttle program are you referring to?
... program operated from 1981 to 2011 by NASA?
What specific aspect of the purpose are you interested in?
... by NASA in terms of its mission objectives?
Vanilla Query Rewrite
Iterative Clarification and Rewriting (Ours)

Figure 1: ICR differs from those traditional query rewriting methods in that we use clarification questions as the pivot to guide rewriting.

2023a; Mo et al., 2023). However, manually labeled rewritten queries often only conform to human readability and may not necessarily be conducive to conversational search. Therefore, some recent studies (Ye et al., 2023; Mao et al., 2023b; Jang et al., 2024; Mo et al., 2024; Lai et al., 2025; Yoon et al., 2025) have explored using LLMs to generate rewritten queries. Although these studies have achieved better retrieval performance, they still employed an end-to-end paradigm, directly generating rewritten query based on the dialogue history and current query. As with humans, LLMs do not always produce the best output on the first try (Madaan et al., 2023). Therefore, the limitation of the above paradigm lies in the fact that the user's query may have multiple potential ambiguities in expression (e.g., Q_3 in Figure 1 does not specify either what program it is or when the program occurs), and it is difficult to resolve all the ambiguities by rewriting the query in one step, making it challenging to deal with extremely ambiguous queries. This highlights the need for advanced rewrite mechanisms to handle these complex scenarios.

To address the above issues, we propose a novel framework ICR (Iterative Clarification and Rewriting). ICR first constructs iterative clarification-

[*] Corresponding author

rewriting data based on the retrieval performance of the query. The model then learns to engage in the process of clarification-rewriting, alternately generating clarification questions and rewritten queries. This process is critical to identify and rectify the ambiguites and omissions in the query. As shown in Figure 1, this strategy advances the traditional CQR methods by explicitly decomposing the query rewriting process into multiple iterations that include clarification and rewriting, each designed to progressively refine the rewritten queries. Since multiple queries are generated during the iteration process, we fused the retrieval results based on the contribution of the rewritten queries in the iterative process. The experimental results on TopiOCQA and QReCC show that our ICR can achieve SOTA performance.

2 Related Work

Previous work used manually labeled rewritten queries as supervisory signals. Lin et al. (2020) introduced pre-trained language models to relax the independence assumptions made when using MLE objective in the CQR task. CONQRR (Wu et al., 2022) used reinforcement learning to adjust and optimize against off-the-shelf retrievers for conversational retrieval. Since previous studies generated tokens one by one from scratch, EDIRCS (Mao et al., 2023a) simultaneously selected the tokens from dialogues and generated a portion of new tokens, and then proposed two conversational search-oriented learning objectives. ConvGQR (Mo et al., 2023) proposed a knowledge fusion mechanism combining query rewriting and expansion.

Since manually labeled rewritten queries are often not optimal, recent work has focused on using LLMs to generate rewritten queries. Ye et al. (2023) defined four important properties for rewritten queries and used LLMs as rewrite editors. LLM4CS (Mao et al., 2023b) explored three prompting methods to generate rewritten queries. By using the signal of information retrieval as a reward, IterCQR (Jang et al., 2024) did not rely on manually annotated rewritten queries. CHIQ (Mo et al., 2024) enhanced the conversation history and then generated rewritten queries based on the enhanced conversation history through three approaches. Considering the two perspectives of the term and semantics, AdaCQR (Lai et al., 2025) introduced the contrastive loss to optimize the reformulation model. RETPO (Yoon et al., 2025) optimized a language model to generate rewritten queries preferred by the retriever.

Unlike previous studies directly using manually labeled or LLM-labeled rewritten queries as supervised signals, our ICR decomposes CQR into several iterations of clarification-rewriting, dynamically rewriting the current query and continuously refine it during iterations.

3 Methodology

3.1 Problem Formulation

Conversational search is to find the top-K most relevant passages $\mathcal{P}_t = \{p_j\}_{j=1}^{K}$ from a large collection $\mathcal{C}$ ($p_j \in \mathcal{C}$) based on the current query q_t and the dialogue history $\mathcal{H}_{t-1} = \{q_i, a_i\}_{i=1}^{t-1}$ where q_i and a_i are the i-th query and response, respectively. These passages are then used to generate the t-th turn response a_t. The CQR task is commonly employed as an intermediate step to rewrite the ambiguous query q_t into a self-contained rewritten query r_t, which is then used for passage retrieval. The process can be formally represented as follows:

$$r_t \leftarrow CQR\left([\mathcal{H}_{t-1}; q_t]\right), \mathcal{P}_t \leftarrow \mathcal{R}(r_t, \mathcal{C}, K), \quad (1)$$

where $CQR(\cdot)$ represents the CQR model and $\mathcal{R}(\cdot)$ can be either a sparse or dense retriever. For brevity, we omit the subscripts later and CQR can be represented as $r \leftarrow CQR\left([\mathcal{H}; q]\right)$.

3.2 Overview

As shown in Figure 2, our ICR consists of four modules. We first construct iterative clarification-rewriting data offline based on the retrieval performance of rewritten queries. Secondly, we propose a progressive fine-tuning scheme, learning to ask clarification questions and rewrite queries based on those clarification questions. Thirdly, we construct preference data for preference alignment from three dimensions: overthinking, underthinking, and insufficient decomposition. Finally, to fully take into account the contribution of the different rewritten queries during the iteration process, we propose process-aware reciprocal rank fusion.

3.3 Clarification-Rewriting Data Generation

To construct the clarification-rewriting iterative data, we introduce LLMs to simulate the roles of asking clarification questions and rewriting based on the clarification questions. The specific process of Clarification-Rewriting Data Generation

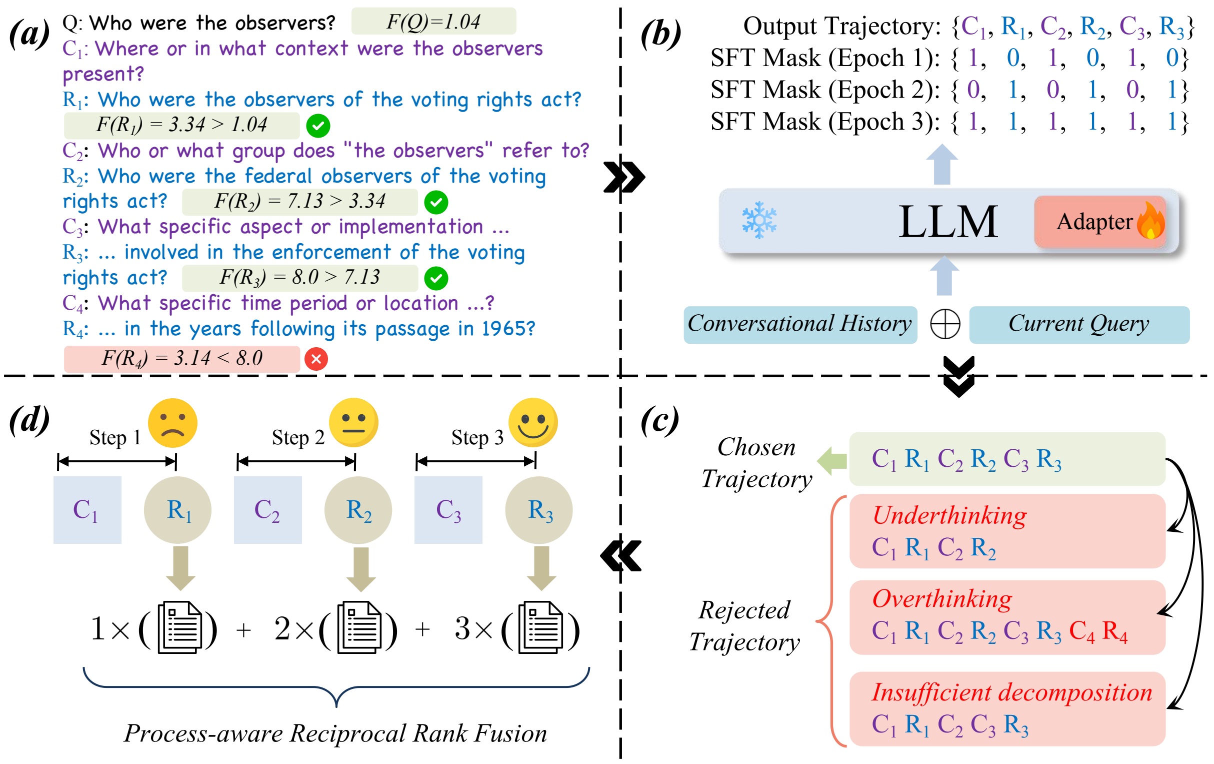

(a)
Q: Who were the observers?
F(Q)=1.04
C1: Where or in what context were the observers present?
R1: Who were the observers of the voting rights act?
F(R1) = 3.34 > 1.04
C2: Who or what group does "the observers" refer to?
R2: Who were the federal observers of the voting rights act?
F(R2) = 7.13 > 3.34
C3: What specific aspect or implementation ...
R3: ... involved in the enforcement of the voting rights act?
F(R3) = 8.0 > 7.13
C4: What specific time period or location ...?
R4: ... in the years following its passage in 1965?
F(R4) = 3.14 < 8.0
(b)
Output Trajectory: {C1, R1, C2, R2, C3, R3}
SFT Mask (Epoch 1): { 1, 0, 1, 0, 1, 0}
SFT Mask (Epoch 2): { 0, 1, 0, 1, 0, 1}
SFT Mask (Epoch 3): { 1, 1, 1, 1, 1, 1}
LLM
Adapter
Conversational History
Current Query
(d)
Step 1
Step 2
Step 3
C1 R1 C2 R2 C3 R3
1×( ) + 2×( ) + 3×( )
Process-aware Reciprocal Rank Fusion
(c)
Chosen Trajectory
C1 R1 C2 R2 C3 R3
Rejected Trajectory
Underthinking
C1 R1 C2 R2
Overthinking
C1 R1 C2 R2 C3 R3 C4 R4
Insufficient decomposition
C1 R1 C2 C3 R3

Figure 2: **Overview of the ICR framework.** ICR consists of four main modules: (a) CRDG: Clarification-Rewriting Data Generation (§ 3.3), (b) PSFT: Progressive Supervised Fine-Tuning from Clarification to Rewriting (§ 3.4), (c) DPO-CRP: Direct Preference Optimization for Clarification-Rewriting Process (§ 3.5), and (d) PRRF: Process-aware Reciprocal Rank Fusion (§ 3.6).

(CRDG) is shown in Algorithm 1. In each iteration, we first use an LLM to generate a clarification question based on the query from the previous iteration, and then generate a new query based on the conversational context and the current clarification question (the prompts for clarification and rewriting are shown in Appendix A) as follows:

$$c_i \sim \mathcal{M}(r_{i-1}), r_i \sim \mathcal{M}(c_i, \mathcal{H}, r_{i-1}), \quad (2)$$

where c_i and r_i represent the clarification question and rewritten query generated in the i-th turn, respectively. Note that in the first iteration, we use the original query (i.e., $r_0 = q$). In this process, we need to determine when to stop asking clarification questions, i.e., generate the final rewritten query. If the quality of the rewritten query does not improve continuously, it should stop asking clarification questions. Here we use the retrieval performance of rewritten query to measure the quality of rewritten query as follows:

$$F(r) = \sum_{m \in MT} m_s(r, d_r) + m_d(r, d_r), \quad (3)$$

where $MT = \{\text{MRR}, \text{NDCG}, \text{R@10}, \text{R@100}\}$ which is described in Section 4.1, m_s represents sparse retrieval, m_d represents dense retrieval, and d_r represents the gold passages corresponding to the query r. $F(r)$ denotes the sum of the four metrics in MT for both sparse and dense retrieval. A higher $F(r)$ indicates a better quality rewritten query. In the i-th iteration, if $F(r_i) \leq F(r_{i-1})$, the clarification and rewriting of this round fails and resampling is performed.

Algorithm 1 Data Construction

```
Require: Conversational query rewriting dataset $D_{CQR} = \{\mathcal{H}^i, q^i, \mathcal{P}^i\}$; Early stopping parameter $\mathcal{E}$; Maximum number of clarification iterations $\mathcal{I}$; Large language model $\mathcal{M}$.
1:  $D_{cr} \leftarrow \{\}$ ▷ Initialize the clarification-rewriting dataset
2:  for each $(\mathcal{H}^i, q^i, \mathcal{P}^i) \in D_{CQR}$ do
3:      $R_{max} \leftarrow 0, num_d \leftarrow 0, \hat{q} \leftarrow q^i, \tau \leftarrow$ “”
4:      for $e = 1$ to $\mathcal{I}$ do
5:          $c \leftarrow \mathcal{M}(\hat{q})$ ▷ Sampling clarification question
6:          $r \leftarrow \mathcal{M}(\mathcal{H}^i, \hat{q}, c)$ ▷ Generate rewritten query
7:          Calculate $F(r)$ according to $\mathcal{P}^i$ and Eq. 3.
8:          if $F(r) > R_{max}$ then
9:              $R_{max} \leftarrow F(r), num_d \leftarrow 0$
10:             $\hat{q} \leftarrow r$ ▷ Update rewritten query
11:         else
12:             $num_d \leftarrow num_d + 1$
13:             Break if $num_d \geq \mathcal{E}$
14:         end if
15:         Add “[Clarification] $c$ [Rewrite] $r$” to $\tau$
16:     end for
17:     Add $(H^i, q^i, \tau)$ to $D_{cr}$
18: end for
```

We set the early stopping parameters $\mathcal{E}$ and maximum iteration rounds $\mathcal{I}$. If the retrieval performance does not improve for $\mathcal{E}$ consecutive iterations or has iterated $\mathcal{I}$ rounds, the iteration will be terminated. Based on the above operation, we can collect an iterative clarification-rewriting dataset $D_{cr} = \{x^{(i)}, \tau^{(i)}\}$. x is the concatenation of dialogue history $\mathcal{H}$ and the current query q, $\tau = (c_1, r_1, \ldots, c_n, r_n)$ is the clarification-rewriting iterative trajectory, where $F(r_1) < F(r_2) < \cdots < F(r_n)$. In order to distinguish between clarification question and rewritten query, as well as to allow parsing of the model output during inference, we add `[Clarification]` and `[Rewrite]` respectively before the clarification question and rewritten query.

3.4 Progressive Supervised Fine-Tuning from Clarification to Rewriting

Once the iterative clarification-rewriting data D_{cr} is ready, we optimize the model π_θ to initialize the clarification-rewriting behavior. During the clarification-rewriting iterative process, the model needs to learn how to ask clarification questions as well as how to rewrite based on the clarification questions. Since asking clarification questions and rewriting are two different skills, training them together may confuse the model. How to decouple two skills is crucial for the learning of the model.

Therefore, we propose a Progressive Supervised Fine-Tuning (PSFT) approach, consisting of three epochs. In the first and second epochs, we mask out the loss corresponding to the rewritten queries and clarification questions respectively. In the third epoch, no tokens are masked out, i.e., training the model's ability to ask clarification questions and rewrite simultaneously. For the token t in the input x, its corresponding mask is $\delta_{mask}(t)$ as follows:

$$\delta_{mask}(t) = \begin{cases} 0, & \text{if Type}(t) = \texttt{Rewrite}\ \&\ \text{Epoch 1} \\ 0, & \text{if Type}(t) = \texttt{Clarification}\ \&\ \text{Epoch 2} \\ 1, & \text{otherwise} \end{cases}$$

where Type$(\cdot)$ indicates which type the token belongs to, Type(t) = `Rewrite` if it corresponds to a rewritten query, or Type(t) = `Clarification` if it corresponds to a clarification question. We optimize our policy π_θ by minimizing the following objective:

$$\mathcal{L} = -\mathbb{E}_{(x,\tau)\sim\mathcal{D}_{cr}} \sum_{t\in\tau} \delta_{mask}(t) \log \pi_\theta(t \mid x, \tau_{:t}). \quad (4)$$

3.5 Direct Preference Optimization for Clarification-Rewriting Process

Previous studies have shown that LLMs exhibit phenomena of overthinking (Chen et al., 2024) and underthinking (Wang et al., 2025). In the iterative clarification-rewriting framework we propose, the models need to avoid issues of overthinking and underthinking. On the one hand, the model should not ask overly detailed clarification questions that deviate from the original query's intent (i.e., overthinking). On the other hand, the model should not end asking clarification questions prematurely in cases of insufficient information (i.e. underthinking). In addition, in the previous data construction phase, we found that LLMs sometimes raise two clarification questions in one step. We expect the model to break down rewriting into individual subproblems.

To address the above issues, we propose Direct Preference Optimization for Clarification-Rewriting Process (DPO-CRP), which constructs preference data from three perspectives: overthinking, underthinking, and insufficient decomposition. This enables the model to ask appropriate clarification questions at the right time. First, we take the data constructed in Section 3.3 as chosen samples. Assuming the chosen sample's corresponding iterative trajectory is $\tau_w = (c_1, r_1, \ldots, c_n, r_n)$. Then, rejected trajectories are constructed from three perspectives: overthinking, underthinking, and insufficient decomposition as follows.

Overthinking To build data for overthinking, we need to ensure that the iterative process of the model has redundant steps. Therefore, we add an extra step (c_{n+1}, r_{n+1}) to the original iterative trajectory τ_w, thereby obtaining the iterative trajectory of overthinking: $\tau_{ot} = (c_1, r_1, \ldots, c_n, r_n, c_{n+1}, r_{n+1})$. To ensure that this step is redundant, we verify whether $F(r_{n+1}) \leq F(r_n)$ during the sampling process:

$$\begin{aligned} c_{n+1} \sim \mathcal{M}(r_n), r_{n+1} \sim \mathcal{M}(c_{n+1}, \mathcal{H}, r_n), \\ s.t. F(r_{n+1}) \leq F(r_n). \end{aligned} \quad (5)$$

In this way, overthinking preference dataset $D_{ot} = \{x^{(i)}, \tau_w^{(i)}, \tau_{ot}^{(i)}\}$ is constructed.

Underthinking We noticed that the underthinking iteration trajectory ends the iteration before reaching the optimal query. To construct such data, we randomly truncate the trajectory. Specifically, for the trajectory τ_w, we randomly sample a position e from $[1, n-1]$, then the corresponding underthinking trajectory is $\tau_{ut} = (c_1, r_1, \dots, c_e, r_e)$. Then the underthinking preference dataset is $D_{ut} = \{x^{(i)}, \tau_w^{(i)}, \tau_{ut}^{(i)}\}$.

Insufficient decomposition During the previous data construction process, we found that the model would continuously raise two clarification questions in one iteration. To alleviate this insufficient decomposition issue, we sample a position i from $[1, n-1]$, and merge (c_i, r_i) with (c_{i+1}, r_{i+1}) to form $([c_j; c_{j+1}], r_{j+1})$, where $[\cdot;\cdot]$ denotes the concatenation operation. As a result, we obtain the corresponding insufficiently decomposed trajectory $\tau_{id} = (c_1, r_1, \dots, [c_j; c_{j+1}], r_{j+1}, \dots)$ and the preference dataset $D_{id} = \{x^{(i)}, \tau_w^{(i)}, \tau_{id}^{(i)}\}$.

Direct Preference Optimization By constructing the rejected trajectories for the above three dimensions, we can obtain a process preference optimization dataset $D_{pref} = D_{ot} \cup D_{ut} \cup D_{id}$. Specific information about dataset D_{pref} can be found in Appendix B. After that, we can optimize the policy model π_θ by minimizing the following loss on dataset D_{pref} through DPO (Rafailov et al., 2023) training:

$$\mathcal{L}_{\mathrm{DPO}}(\pi_\theta; \pi_{ref}; \mathcal{D}_{pref}) = -\mathbb{E}_{(x,\tau_w,\tau_l)\sim\mathcal{D}_{pref}} \left[\log\sigma\left(\beta\log\frac{\pi_\theta(\tau_w|x)}{\pi_{\mathrm{ref}}(\tau_w|x)} - \beta\log\frac{\pi_\theta(\tau_l|x)}{\pi_{\mathrm{ref}}(\tau_l|x)}\right)\right], \tag{6}$$

where π_{ref} is the reference model initialized from the original model before DPO.

3.6 Process-aware Reciprocal Rank Fusion

During the inference phase, the model iteratively asks clarification questions and generates the rewritten queries based on the conversational context. We parse the model's output to obtain the set of rewritten queries $Q = \{r_1, r_2, \dots, r_{|Q|}\}$ in each iteration, where r_i represents the rewritten query output by the model in the i-th iteration. In this process, multiple rewritten queries may be generated, each exhibiting distinct retrieval efficacy. Directly employing the query from the final iteration for retrieval can introduce noise due to error propagation throughout the clarification-rewriting iterative process, which may cause deviation from the original query. The primary challenge is effectively leveraging all generated rewritten queries.

We can notice that the retrieval performance of the rewritten queries corresponding to the later iterations is often better, which is due to the model's continuous refinement of the query during the clarification-rewriting process. Inspired by Reciprocal Rank Fusion (Cormack et al., 2009), we propose Process-Aware Reciprocal Rank Fusion (PRRF). Specifically, we weight the retrieval results corresponding to each rewritten query in the iterative process, with the weight coefficient being the position of the rewritten query in the iterations. The score of the passage d after fusion is as follows:

$$\mathsf{PRRF}(d \in \mathcal{C}) = \sum_{i\in[1,|Q|]} \frac{i}{rank(r_i, d) + k}, \tag{7}$$

where $rank(r_i, d)$ denotes the ranking of passage d in the retrieval results of r_i and k is a constant that we set to 60. We multiply the weight of the rewritten query in the i-th iteration by i, so that the retrieval results of queries in later iterations have a greater impact on fusion.

4 Experiments

4.1 Experimental Settings

Datasets and Evaluation Metrics We conducted experiments on two widely used datasets, TopiOCQA (Adlakha et al., 2022) and QReCC (Anantha et al., 2021). To evaluate the zero-shot capability of ICR, we also conducted zero-shot analysis on CAsT-19 (Dalton et al., 2020), CAsT-20 (Dalton et al., 2021), and CAsT-21 (Dalton et al., 2022). Following previous work, we evaluated the performance of the models on four metrics: MRR, NDCG@3, Recall@10, and R@100. In Appendices B and C, we provide specific information on the datasets and evaluation metrics.

Implementation Details In order to compare fairly with previous work, we use `gpt-3.5-turbo-0125` to generate iterative clarification-rewriting data and `Llama-2-7b-hf` as the backbone π_θ. In data construction, the early stopping parameter $\mathcal{E}$ and the maximum number of iteration rounds $\mathcal{I}$ are set to 3 and 10, respectively. Following the previous work (Mo et al., 2024; Lai et al., 2025; Yoon et al., 2025), we also adopted query expansion. More implementation details can be found in Appendix D.

Baselines To verify the effectiveness of ICR, we compare it with the following baselines: EDIRCS (Mao et al., 2023a), LLM-Aided (Ye et al., 2023), LLM4CS (Mao et al., 2023b), IterCQR (Jang et al., 2024), CHIQ (Mo et al., 2024), AdaCQR (Lai et al., 2025) and RETPO (Yoon et al., 2025).

4.2 Main Results

As shown in Table 1, ICR significantly outperforms all baselines in both sparse retrieval and dense retrieval. For example, in the dense retrieval of

Type	System	Backbone	TopiOCQA				QReCC			
			MRR	NDCG	R@10	R@100	MRR	NDCG	R@10	R@100
Sparse (BM25)	EDIRCS	T5-base	-	-	-	-	41.2	-	62.7	90.2
	LLM-Aided	ChatGPT	-	-	-	-	49.4	46.5	67.1	88.2
	LLM4CS	ChatGPT	27.9	26.4	48.4	71.1	51.6	49.3	75.3	92.6
	CHIQ	LLaMA2-7B	25.6	23.5	44.7	-	54.3	51.9	**78.5**	-
	IterCQR	T5-base	16.5	14.9	29.3	54.1	46.7	44.1	64.4	85.5
	AdaCQR	T5-base	28.3	26.5	48.9	71.2	55.1	52.5	76.5	93.7
	RETPO	LLaMA2-7B	28.3	26.5	48.3	73.1	50.0	47.3	69.5	89.5
	ICR	LLaMA2-7B	**31.4**	**30.4**	**52.8**	**76.3**	**58.4**	**54.9**	78.3	**95.9**
Dense (ANCE)	EDIRCS	T5-base	-	-	-	-	42.1	-	65.6	85.3
	LLM-Aided	ChatGPT	-	-	-	-	43.5	41.3	65.6	82.3
	LLM4CS	ChatGPT	35.4	34.4	55.2	72.2	44.7	41.8	67.2	84.0
	CHIQ	LLaMA2-7B	38.0	37.0	61.6	-	47.2	44.6	70.8	-
	IterCQR	T5-base	26.3	25.1	42.6	62.0	42.9	40.2	65.5	84.1
	AdaCQR	T5-base	38.5	37.6	58.4	75.0	45.8	42.9	67.3	83.8
	RETPO	LLaMA2-7B	30.0	28.9	49.6	68.7	44.0	41.1	66.7	84.6
	ICR	LLaMA2-7B	**42.1**	**40.4**	**64.3**	**78.3**	**49.5**	**46.8**	**73.2**	**88.3**

Table 1: Evaluation results of various retrieval system types on the test sets of TopiOCQA and QReCC.

TopiOCQA, MRR, NDCG, R@10 and R@100 are improved by 3.6, 2.8, 5.9 and 3.3, respectively, compared to the best AdaCQR, indicating that ICR can not only recall more relevant passages but also rank them higher. In comparison with the aforementioned LLMs-based baselines (e.g., CHIQ, IterCQR, AdaCQR and RETPO), ICR has made significant improvements. This finding indicates that the iterative process of clarification and rewriting enables the model to progressively refine the query, in contrast to the alternative of the model generating the rewritten query directly.

It is worth noting that both ICR and CHIQ used retrieval result fusion. However, CHIQ requires two rounds of query generation, while ICR can generate all rewritten queries at once. In addition, ICR considers the contributions of different rewritten queries for fusion. Therefore, ICR significantly outperforms CHIQ on almost all metrics, except for the R@10 metric on QReCC.

We also provide the performance analysis of ICR on CAsT 19-21 in the Appendix E, which demonstrates that ICR has excellent zero-shot generalization capabilities.

4.3 Ablation Study

To analyze the contribution of each component, we performed ablation experiments shown in Table 2.

DPO-CRP To optimize the clarification-rewriting iteration process, we designed preference data from three perspectives: overthinking, underthinking, and insufficient decomposition. We have separately removed the preference data for the above three dimensions, as shown in "w/o OT" (OverThinking),

Variant	TopiOCQA			
	MRR	NDCG	R@10	R@100
ICR	42.1	40.4	64.3	78.3
w/o. OT	$41.2_{\downarrow 0.9}$	$39.0_{\downarrow 1.4}$	$63.1_{\downarrow 1.2}$	$77.1_{\downarrow 1.2}$
w/o. UT	$40.9_{\downarrow 1.2}$	$38.9_{\downarrow 1.5}$	$62.9_{\downarrow 1.4}$	$77.2_{\downarrow 1.1}$
w/o. ID	$41.6_{\downarrow 0.5}$	$39.5_{\downarrow 0.9}$	$63.5_{\downarrow 0.8}$	$77.6_{\downarrow 0.7}$
w/o. DPO-CRP	$39.4_{\downarrow 2.7}$	$37.5_{\downarrow 2.9}$	$61.8_{\downarrow 2.5}$	$76.2_{\downarrow 2.1}$
Vanilla SFT	$41.6_{\downarrow 0.5}$	$40.1_{\downarrow 0.3}$	$63.7_{\downarrow 0.6}$	$77.8_{\downarrow 0.5}$
Final Rewriting	$41.3_{\downarrow 0.8}$	$39.2_{\downarrow 1.2}$	$63.4_{\downarrow 0.9}$	$77.4_{\downarrow 0.9}$
RRF	$40.9_{\downarrow 1.2}$	$38.7_{\downarrow 1.7}$	$63.1_{\downarrow 1.2}$	$76.9_{\downarrow 1.4}$
$\text{ICR}_{\text{GPT-4.1}}$	$42.8_{\uparrow 0.7}$	$41.1_{\uparrow 0.7}$	$64.9_{\uparrow 0.6}$	$78.7_{\uparrow 0.4}$
$\text{ICR}_{\text{Qwen3-14B}}$	$42.6_{\uparrow 0.5}$	$40.9_{\uparrow 0.5}$	$64.7_{\uparrow 0.4}$	$78.5_{\uparrow 0.2}$

Table 2: Ablation study for each component of ICR and different LLMs.

"w/o UT" (UnderThinking), and "w/o ID" (Insufficient Decomposition) in Table 2, respectively. Removing any preference dimension reduces retrieval performance, showing that all three dimensions are crucial for effective clarification-rewriting iterations. Notably, eliminating the underthinking preference data causes the most significant drop in performance, potentially due to the fact that the model is more inclined to end iterations prematurely as opposed to iterating redundant steps. We also tried to remove the whole DPO-CRP ("w/o DPO-CRP") directly. If only supervised fine-tuning is performed without preference optimization, the model's performance has dropped significantly.

PSFT Recognizing the distinct skills of asking clarification questions and rewriting, we propose Progressive Supervised Fine-Tuning (PSFT). We also tried directly fine-tuning the model for 3 epochs without masking any tokens ("Vanilla SFT"), and the retrieval performance has declined across all metrics. This indicates that PSFT can guide the model from asking clarification questions to rewrit-

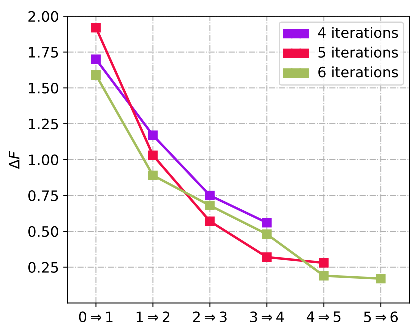


Figure 3: The magnitude of change ΔF in the retrieval performance of queries between adjacent iteration steps.

ing.

PRRF To analyze the effectiveness of our proposed Process-aware Reciprocal Rank Fusion (PRRF), we evaluated two additional variants, i.e., directly using the rewritten query from the last iteration ("Final rewriting") and reciprocal rank fusion without considering the contribution of rewritten queries ("RRF"). It can be observed that using the final rewritten query and using the reciprocal rank fusion both lead to performance degradation. Additionally, the retrieval performance of RRF is worse than that of directly using the final rewritten query. This indicates that some queries in the early iteration process are still incomplete, and their retrieval results may interfere with the fusion process.

Different LLMs To analyze whether ICR can be adapted to other models, we used GPT-4.1 for data construction and Qwen3-14B as the backbone for training, with results shown in Table 2 for ICR$_{\texttt{GPT-4.1}}$ and ICR$_{\texttt{Qwen3-14B}}$. It can be observed that using larger and more advanced models can achieve better performance, which indicates that ICR can be adapted to other models. Notably, ICR$_{\texttt{GPT-4.1}}$ exhibits a more pronounced performance enhancement relative to ICR$_{\texttt{Qwen3-14B}}$, potentially attributable to the critical role of high-quality iterative trajectory data over employing a robust backbone model.

4.4 Analysis

Evolution of Queries between Adjacent Iterative Steps We analyze the variation of retrieval performance of queries between adjacent iteration steps for samples with the iteration steps 4, 5, and 6, as shown in Figure 3. We can observe that in the initial iterations, the growth of retrieval performance (i.e. $F(\cdot)$) is very rapid. As the iterations increase, the growth rate gradually slows down. This phenomenon arises from the saturation of query information, whereby additional clarification becomes redundant and potentially detrimental to retrieval performance.

	TopiOCQA		QReCC	
	Sparse	Dense	Sparse	Dense
Before DPO-CRP				
LSR	77.8	78.4	79.8	79.1
GSR	60.2	60.4	62.4	61.2
After DPO-CRP				
LSR	80.2$_{\uparrow 2.4}$	81.0$_{\uparrow 2.6}$	83.2$_{\uparrow 3.4}$	82.8$_{\uparrow 3.7}$
GSR	63.2$_{\uparrow 3.0}$	64.2$_{\uparrow 3.8}$	66.4$_{\uparrow 4.0}$	65.7$_{\uparrow 4.5}$

Table 3: LSR and GSR on TopiOCQA and QReCC.

Analysis of Iterative Process The iterative nature of ICR can lead to error propagation, where inappropriate clarification questions in earlier iterations result in a rewritten query that contradicts the original intent. To quantitatively assess the iterative improvement of query quality, we define the local success rate LSR and the global success rate GSR:

$$\texttt{LSR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathcal{D}^i_{test}|}\sum_{r_i^j\in\mathcal{D}^i_{test}}\mathbb{1}[(F(r_i^{j-1})<F(r_i^j))],$$
$$\texttt{GSR} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{1}\left[\bigwedge_{r_i^j\in\mathcal{D}^i_{test}}(F(r_i^{j-1})<F(r_i^j))\right],$$

where $\mathbb{1}(\cdot)$ is the indicator function, and N is the number of data entries. r_i^j represents the rewritten query corresponding to the j-th iterative step of the i-th test sample $\mathcal{D}^i_{test}$ (specifically, r_i^0 represents the original query). $F(\cdot)$ is the metric used to calculate the quality of the rewritten query in Equation 3. LSR and GSR measure the proportion of effective iterative steps, i.e., the clarification questions posed by the model during the iteration process enhance the retrieval performance of the rewritten query. LSR and GSR are measured from both step-level and sample-level respectively. As shown in Table 3, both LSR and GSR are significantly improved after using DPO-CRP. This indicates that the preference samples designed in three dimensions can effectively guide the clarification and rewriting process. Moreover, we observed that the improvement in GSR is greater than LSR after using DPO-CRP. This is because training with DPO-CRP corrected minor errors caused by only a few steps of the iteration trajectory in certain samples, thereby increasing the proportion of effective trajectories.

We also analyze the number of redundant steps in the overthinking preference data in Appendix F, and provide a latency analysis of ICR in Appendix G.

F($\cdot$)	Type	TopiOCQA			
		MRR	**NDCG**	**R@10**	**R@100**
$m_s(\cdot)+m_d(\cdot)$	Sparse	31.4	30.4	52.8	76.3
	Dense	42.1	40.4	64.3	78.3
$m_s(\cdot)$	Sparse	29.3	28.5	50.8	74.2
	Dense	39.8	37.8	61.7	75.9
$m_d(\cdot)$	Sparse	28.7	27.8	49.9	73.7
	Dense	40.1	38.2	62.2	76.6

Table 4: Impact on model performance when different retrieval types are used as query quality indicators.

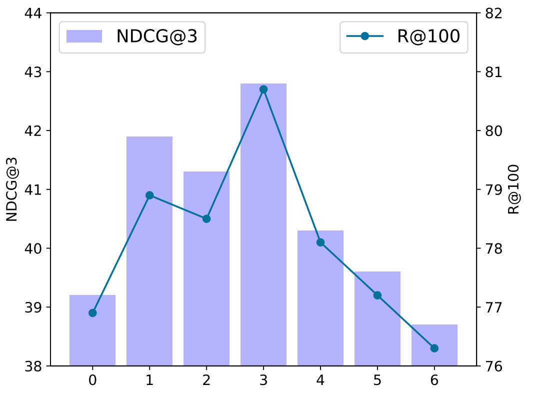


Figure 4: The correlation between iterative steps and model performance.

4.5 Quality Measurement of Rewritten Queries

During the process of constructing iterative data, we measure the quality of queries by combining the performance of sparse retrieval and dense retrieval. To verify the impact of different types of retrieval performance on measuring query quality, we attempted using only sparse retrieval and only dense retrieval separately, as shown in Table 4. It can be observed that using only sparse retrieval (i.e., $m_s(\cdot)$) or only dense retrieval (i.e., $m_d(\cdot)$) to measure query quality leads to degradation of model performance. In addition, when using sparse retrieval to measure query quality, the performance of dense retrieval decreases more significantly, while when using dense retrieval to measure query quality, the performance of sparse retrieval decreases more significantly. This is due to the fact that sparse and dense retrieval take into account lexical and semantic level information respectively, and measuring performance using only one type of retrieval will lead to bias.

4.6 Correlation between Iterations and Performance

The number of iterations in the clarification and rewriting process varies depending on the query. As shown in Figure 4, we analyzed the correlation between the number of iterations and model performance on the TopiOCQA dataset. It can be observed that the performance is best when the number of iteration steps is 3. This could be because of insufficient rewriting with few iterations. On the other hand, excessive iterations can lead to unnecessary clarification questions that introduce irrelevant noise in the query, which hinders retrieval performance. This validates the rationality of the overthinking and underthinking data we designed in the DPO-CRP process.

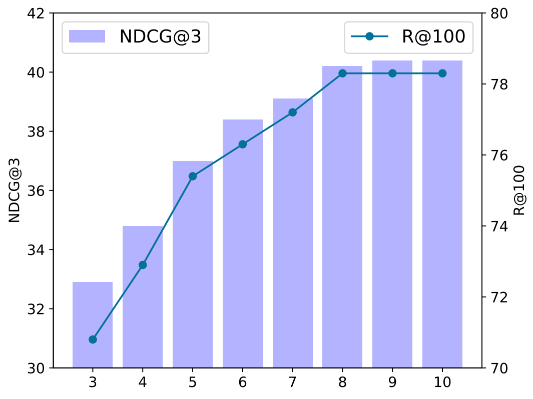


Figure 5: Impact of maximum number of iteration rounds $\mathcal{I}$ on model performance.

4.7 Maximum Number of Iterations $\mathcal{I}$ in Data Construction

During the iterative data construction phase, we default to a maximum of 10 iterations, exiting when the iteration number reaches 10. We analyzed the impact of the maximum number of iterations $\mathcal{I}$ on the model performance shown in Table 5. When the number of iterations is less than 8, increasing the number of iterations continuously improves the model performance. This is due to the fact that when the maximum number of iterations is small, the model may exit the iteration prematurely, resulting in insufficient information in the rewritten query. This situation leads to bias in the generated trajectory data, thereby preventing the model from correctly iterating.

4.8 Qualitative Analysis and Human Evaluation

To qualitatively assess the evolution of queries during the clarification-rewriting iteration process, we define two metrics: *Completeness* and *Factuality*. *Completeness* assesses whether a query encapsulates all necessary information, while *Factuality* evaluates the alignment of query content with the

Comp.	Fact.	Proportion (%)	Avg. ΔF
✗	✗	2.0	-3.61
✓	✗	9.0	-1.32
✗	✓	28.0	0
✓	✓	61.0	4.10

Table 5: Analysis of completeness and factuality changes during the iterative process. ✗ indicates unchanged completeness (Comp.) or decreased factuality (Fact.), while ✓ indicates increased completeness or unchanged factuality.

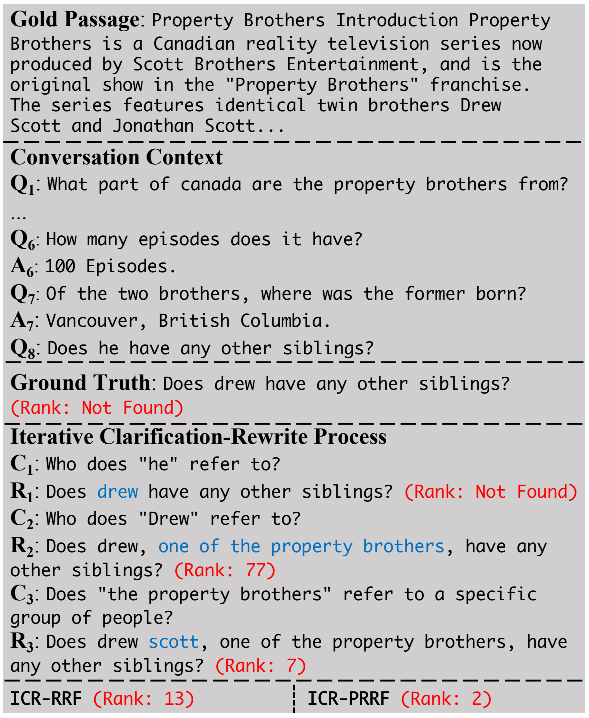
dialogue context. It is obvious that during the iteration process, the *Completeness* of the query does not decrease, while its *Factuality* does not increase. This phenomenon occurs because while information is continuously refined, there is also potential for noise introduction.

We sampled 200 instances with more than one iteration step and distributed these samples to three graduate students in NLP. They annotated the changes in *Completeness* and *Factuality* of the queries from the initial to the final iteration, as shown in Table 5. It can be observed that the *Completeness* and *Factuality* of most queries improved in the final iteration, accounting for 61.0%. Additionally, the retrieval performance of these queries has significantly improved (↑4.10). For 28.0% of the queries, *Completeness* and *Factuality* remained constant due to either no alterations in subsequent iterations or modifications limited to stop words, resulting in unchanged retrieval performance.

During the iterations, although in some cases the model rewrite the query to be more complete based on the clarification questions, it also introduces noise, which account for 9.0% and result in an average performance degradation of 1.32. We note that 2.0% of the queries have unchanged *Completeness* but degraded *Factuality*, which is due to the rewriting of correct information into incorrect information in later iterations, resulting in degraded *Factuality* and reduced retrieval performance.

4.9 Case Study

In Figure 6, we provide a case study where the blue font indicates the newly added key information for retrieval in each iteration. In this example, the model undergoes three clarification-rewriting iterations. The rewritten query generated in each iteration introduced key information that was not present in the previous query. In the first iteration, the model rewrites “he” as “drew”. In the second iteration, the model provided an explanation of the



Gold Passage: Property Brothers Introduction Property Brothers is a Canadian reality television series now produced by Scott Brothers Entertainment, and is the original show in the "Property Brothers" franchise. The series features identical twin brothers Drew Scott and Jonathan Scott...

Conversation Context
Q_1: What part of canada are the property brothers from?
...
Q_6: How many episodes does it have?
A_6: 100 Episodes.
Q_7: Of the two brothers, where was the former born?
A_7: Vancouver, British Columbia.
Q_8: Does he have any other siblings?

Ground Truth: Does drew have any other siblings?
(Rank: Not Found)

Iterative Clarification-Rewrite Process
C_1: Who does "he" refer to?
R_1: Does drew have any other siblings? (Rank: Not Found)
C_2: Who does "Drew" refer to?
R_2: Does drew, one of the property brothers, have any other siblings? (Rank: 77)
C_3: Does "the property brothers" refer to a specific group of people?
R_3: Does drew scott, one of the property brothers, have any other siblings? (Rank: 7)

ICR-RRF (Rank: 13) | ICR-PRRF (Rank: 2)

Figure 6: Case study on TopiOCQA. The red font indicates the ranking of the gold passage in the top-100 retrieval results.

background of “drew” (i.e., “one of the property brothers”). By inferring from “one of the property brothers”, the model introduced “scott” in the third iteration, making the name more specific. During the retrieval process, we found that the effect of Reciprocal Rank Fusion (ICR-RRF) was worse than directly using the final rewritten query (13 vs. 7), which is attributed to the interference of rewritten queries in the first two iterations on the fusion. By using Process-aware Reciprocal Rank Fusion (ICR-PRRF), the weights of the queries in the subsequent iterations can be improved, ultimately ranking the gold passage second in the top-100 retrieval results. In addition, ICR makes query rewriting more transparent by pivoting on clarification questions. We provide more examples in Appendix H.

5 Conclusion

In this paper, we incorporate a novel framework ICR into CQR. ICR involves progressive fine-tuning and constructing preference data for preference alignment from three perspectives: overthinking, underthinking, and insufficient decomposition. Through process-aware fusion, all rewritten queries in the iterative process are fully utilized. On both TopiOCQA and QReCC datasets, our ICR achieves state-of-the-art performance.

Limitations

Although our proposed ICR can advance the research on conversational search, it still suffers from the following two drawbacks. First, we used rewritten queries for the final retrieval, but did not utilize the clarification questions. However, clarification questions often contain some key information. Future research can focus on achieving more efficient retrieval by combining clarification questions with rewritten queries. Second, in the retrieval result fusion module, we fused all the rewritten queries. Even if we consider the contributions of different rewritten queries, some rewritten queries may deviate significantly from the original query and can be directly discarded.

Acknowledgements

The authors would like to thank the three anonymous reviewers for their comments on this paper. This research was supported by the National Natural Science Foundation of China (Nos. 62376181 and 62276177), and Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

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, Online. Association for Computational Linguistics.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, and 1 others. 2024. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.

Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. Trec cast 2019: The conversational assistance track overview. *arXiv preprint arXiv:2003.13624*.

Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2021. Cast 2020: The conversational assistance track overview. Technical report.

Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2022. Trec cast 2021: The conversational assistance track overview. In *In Proceedings of TREC*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1762–1777. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Yunah Jang, Kang-il Lee, Hyunkyung Bae, Hwanhee Lee, and Kyomin Jung. 2024. IterCQR: Iterative conversational query reformulation with retrieval guidance. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8121–8138, Mexico City, Mexico. Association for Computational Linguistics.

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, Singapore, December 6-10, 2023*, pages 6649–6675. 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.

Yilong Lai, Jialong Wu, Congzhi Zhang, Haowen Sun, and Deyu Zhou. 2025. AdaCQR: Enhancing query reformulation for conversational search via sparse and dense retrieval alignment. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7698–7720, Abu Dhabi, UAE. Association for Computational Linguistics.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. 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*, pages 2356–2362.

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. *Preprint*, arXiv:2004.01909.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pages 2421–2425. ACM.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

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, Toronto, Canada. Association for Computational Linguistics.

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.

Fengran Mo, Abbas Ghaddar, Kelong Mao, Mehdi Rezagholizadeh, Boxing Chen, Qun Liu, and Jian-Yun Nie. 2024. CHIQ: contextual history enhancement for improving query rewriting in conversational search. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 2253–2268. Association for Computational Linguistics.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 11897–11916. Association for Computational Linguistics.

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

Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, and 1 others. 2025. Thoughts are all over the place: On the underthinking of o1-like llms. *arXiv preprint arXiv:2501.18585*.

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

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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

Chanwoong Yoon, Gangwoo Kim, Byeongguk Jeon, Sungdong Kim, Yohan Jo, and Jaewoo Kang. 2025. Ask optimal questions: Aligning large language models with retriever‘s preference in conversation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5899–5921, Albuquerque, New Mexico. Association for Computational Linguistics.

Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023. Retrieve anything to augment large language models. *Preprint*, arXiv:2310.07554.

Dataset	Split	#Conv.	#Turns(Qry.)	#Collection
TopiOCQA	Train	3,509	45,450	25M
	Test	205	2,514	
QReCC	Train	10,823	63,501	54M
	Test	2,775	16,451	
CAsT-19	Test	50	479	38M
CAsT-20	Test	25	208	
CAsT-21	Test	26	239	40M

Table 6: Statistics of conversational search datasets.

Dataset	#D_{ot}	#D_{ut}	#D_{id}	#D_{pref}
TopiOCQA	45,450	39,112	39,112	123,674
QReCC	29,596	26,427	26,427	82,450

Table 7: Statistics of the process preference optimization dataset.

A Prompts

In Tables 11 and 12, we provide prompts for generating clarification questions and rewriting queries based on clarification questions, respectively.

B Details of Datasets

We present statistical information on the datasets used in the experiments in Table 6. It is worth noting that for the QReCC dataset, following previous work (Mo et al., 2024; Lai et al., 2025; Yoon et al., 2025), some samples without gold passage label were excluded, resulting in 29,596 and 8,209 turns being retained in the final training set and test set, respectively.

For the construction of the clarification-rewriting process preference optimization dataset D_{pref}, we devised three dimensions: overthinking, underthinking, and insufficient decomposition. Among them, for underthinking and insufficient decomposition, we filter out iterative trajectories with rounds less than 2 because these trajectories cannot be further truncated or merge the intermediate steps. The statistics of the final process preference optimization dataset D_{pref} are shown in Table 7.

In order to analyze the distribution of types of clarification questions, we counted the number of interrogative words in the clarification questions, as shown in Figure 7. The interrogative words with a percentage greater than 1% are “What”, “Which”, and “Who”. The interrogative words included “Others” are “How”, “Where”, “When”, and so on. The interrogative word "What" has the highest proportion in clarification questions, followed by “Who”.

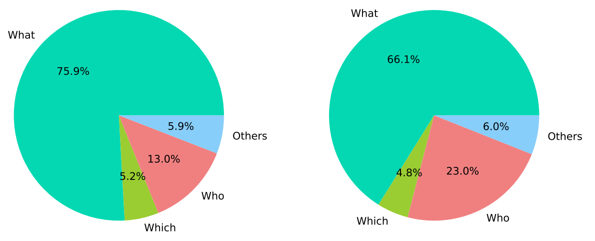


Figure 7: Distribution of number of interrogative words in clarification questions on TopiOCQA (left) and QReCC (right).

These two interrogative words usually involve referring to some entities.

C Details of Evaluation Metrics

MRR focuses on the position at which relevant passages are ranked, with a higher MRR indicating a higher position. NDCG@3 considers the relevance and ranking position of the top 3 retrieval results. Recall@K evaluates the ability to retrieve relevant passages in the top-K results.

D Implementation Details

We used the LoRA (Hu et al., 2022) fine-tuning method, with the LoRA rank set to 8. The model performs DPO training of 3 epochs on the constructed process preference dataset. We set the following hyperparameters for optimization: batch size is 8, gradient accumulation steps are 4, and learning rate is 1e-5. For the retrieval system, we use Faiss (Johnson et al., 2019) for dense retrieval and Pyserini (Lin et al., 2021) for sparse retrieval. In BM25, k_1 and b are set to 0.9 and 0.4 on TopiOCQA, and 0.82 and 0.68 on QReCC, where k_1 controls the non-linear term frequency normalization, while b is the scale of the inverse document frequency. The retriever used in dense retrieval is ANCE (Xiong et al., 2021). We use `pytrec_eval` to calculate the evaluation metrics MRR, NDCG@3, Recall@10, and R@100.

E Zero-shot Analysis

To analyze the zero-shot generalization performance of ICR, we additionally compared ICR with the following methods: LLM-Embedder (Zhang et al., 2023), HyDE (Gao et al., 2023), Query2doc (Wang et al., 2023), InstructorR (Jin et al., 2023), RepLLaMA (Ma et al., 2024) and E5-Mistral (Wang et al., 2024).

We report in Table 8 the generalization performance of ICR on three datasets: CAST-19, CAST-

System	Backbone	CAsT-19			CAsT-20			CAsT-21		
		MRR	N@3	R@10	MRR	N@3	R@10	MRR	N@3	R@10
E5-Mistral	Mistral-7B	62.2	31.3	9.5	22.0	15.4	8.4	48.2	32.5	20.5
HyDE	ChatGPT-3.5	55.6	39.2	10.0	44.8	29.3	16.9	-	-	-
Query2doc	ChatGPT-3.5	58.8	42.4	11.6	48.6	32.5	17.3	-	-	-
InstructorR	ChatGPT-3.5	61.2	46.6	10.4	43.7	29.6	8.3	46.7	32.5	18.4
LLM4CS	ChatGPT-3.5	70.4	46.8	11.7	58.6	41.5	19.3	66.1	46.9	24.4
RepLLaMA	LLaMA2-7B	62.4	31.6	10.6	26.8	18.3	10.4	47.4	32.7	19.6
LLM-Embedder	LLaMA2-7B	63.3	36.6	11.4	25.2	15.4	8.7	46.8	31.2	17.3
CHIQ	LLaMA2-7B	73.3	50.5	12.9	54.0	38.0	19.3	62.9	46.5	25.2
AdaCQR	T5-base	74.5	-	13.8	56.6	-	19.2	64.2	-	25.0
ICR	LLaMA2-7B	**76.9**	**53.7**	**15.3**	**58.9**	**43.8**	**20.3**	**67.9**	**47.8**	**27.1**

Table 8: Zero-shot retrieval performances under the dense retrieval (ANCE).

Variant	TopiOCQA			
	MRR	**NDCG**	**R@10**	**R@100**
ICR	42.1	40.4	64.3	78.3
Multi-Redundant	42.3	40.6	63.8	77.8

Table 9: Analysis of the number of redundant iterative steps in the process of constructing preference data for overthinking.

20, and CAST-21. ICR significantly outperforms previous methods in zero-shot setting, with improvements of 2.4, 2.3, and 3.7 on the MRR of three datasets compared to previous methods. This is attributed to the fact that ICR's iterative rewriting framework can continuously refine the query, propose clarification questions on the ambiguous expressions of the query, and perform rewriting.

F Analysis of Overthinking Preference Data Construction

When constructing overthinking data, we build the trajectory of overthinking by adding a redundant step after the iterative trajectory of the chosen sample. We can also choose to add more redundant steps. Specifically, we randomly sample a number k from $[1, 2, 3, 4]$, and add k redundant steps after $\tau_w = (c_1, r_1, \ldots, c_n, r_n)$, ensuring $F(r_n) \geq F(r_{n+1}) \geq \cdots \geq F(r_{n+k})$. As shown in "Multi-Redundant" of Table 9, increasing the number of redundant iterative steps does not significantly improve performance, and even results in a decrease in performance at R@10 and R@100. One potential reason is that the iterative trajectory after adding more redundant steps differs too much from the original iterative trajectory, widening the gap between chosen samples and rejected samples.

	TopiOCQA	**QReCC**
Latency	2.13	2.17

Table 10: Per-sample latency (in seconds) for iterative trajectory generation on TopiOCQA and QReCC.

The excessive distinction may cause the model to overfit the training data during the preference learning process, unable to grasp the timing to exit the iteration.

G Latency Analysis of ICR

Since ICR requires the generation of complete iterative trajectories during inference, it may incur additional computational overhead compared to directly generating rewritten query. We calculated the latency of ICR from iterative trajectory generation, as shown in Table 10. The inference latency on TopiOCQA and QReCC is 2.13 seconds and 2.17 seconds respectively, which is tolerable for humans and can be deployed in the real world.

H Examples of ICR

In addition to the example in Figure 6, we provide two examples in Figures 8 and 9.

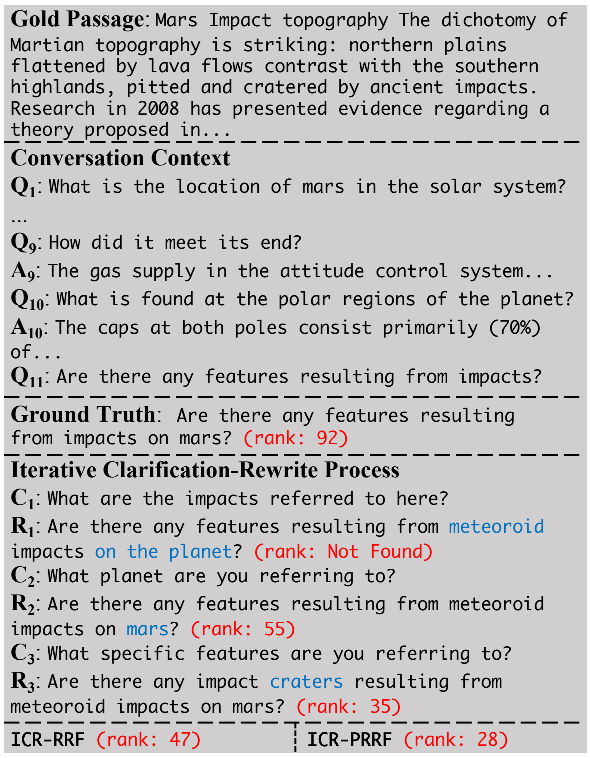
Given a query, this query may be ambiguous. For example, in this query, pronouns may be used to refer to entities or some components may be omitted, so you need to perform coreference resolution and ellipsis resolution. Please ask a question to clarify any unclear points in the query. You only need to output the clarification question, no need to output extra content. Here are some examples.
Examples:
#Query#: Has she produced anything else?
#Clarification Question#: Who does "she" refer to?

#Query#: Has she produced anything else?
#Clarification Question#: What does "anything else" exclude here?

#Query#: Who were the first settlers?
#Clarification Question#: Where are the settlers referred to here?

Please ask a clarification question about the following query.
#Query#: `{Current Query}`
#Clarification Question#:

Table 11: Prompt used in clarification question generation.



Gold Passage: Mars Impact topography The dichotomy of Martian topography is striking: northern plains flattened by lava flows contrast with the southern highlands, pitted and cratered by ancient impacts. Research in 2008 has presented evidence regarding a theory proposed in...

Conversation Context

Q$_1$: What is the location of mars in the solar system?

...

Q$_9$: How did it meet its end?

A$_9$: The gas supply in the attitude control system...

Q$_{10}$: What is found at the polar regions of the planet?

A$_{10}$: The caps at both poles consist primarily (70%) of...

Q$_{11}$: Are there any features resulting from impacts?

Ground Truth: Are there any features resulting from impacts on mars? (rank: 92)

Iterative Clarification-Rewrite Process

C$_1$: What are the impacts referred to here?

R$_1$: Are there any features resulting from meteoroid impacts on the planet? (rank: Not Found)

C$_2$: What planet are you referring to?

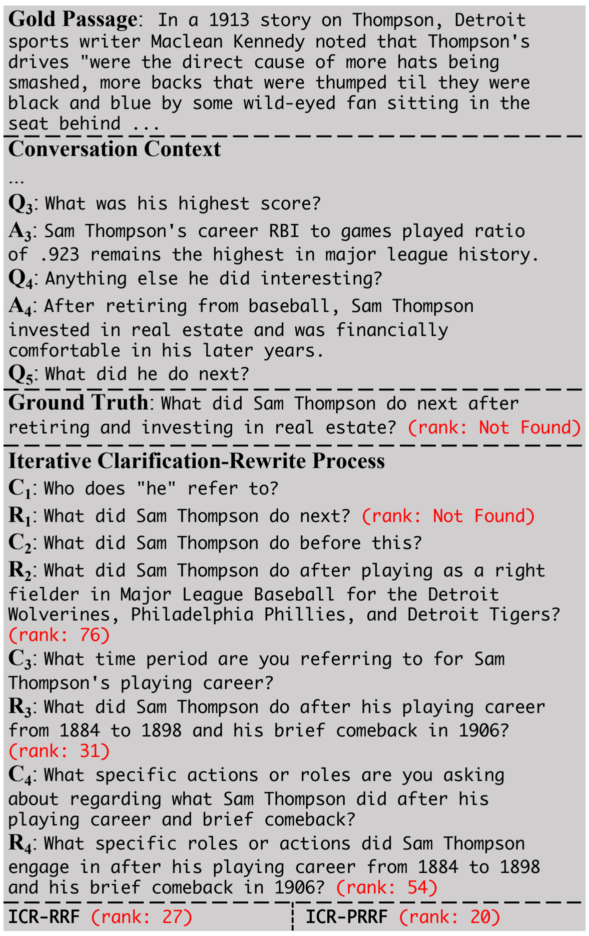
R$_2$: Are there any features resulting from meteoroid impacts on mars? (rank: 55)

C$_3$: What specific features are you referring to?

R$_3$: Are there any impact craters resulting from meteoroid impacts on mars? (rank: 35)

ICR-RRF (rank: 47) | ICR-PRRF (rank: 28)

Figure 8: Example on TopiOCQA.



Gold Passage: In a 1913 story on Thompson, Detroit sports writer Maclean Kennedy noted that Thompson's drives "were the direct cause of more hats being smashed, more backs that were thumped til they were black and blue by some wild-eyed fan sitting in the seat behind ...

Conversation Context

...

Q$_3$: What was his highest score?

A$_3$: Sam Thompson's career RBI to games played ratio of .923 remains the highest in major league history.

Q$_4$: Anything else he did interesting?

A$_4$: After retiring from baseball, Sam Thompson invested in real estate and was financially comfortable in his later years.

Q$_5$: What did he do next?

Ground Truth: What did Sam Thompson do next after retiring and investing in real estate? (rank: Not Found)

Iterative Clarification-Rewrite Process

C$_1$: Who does "he" refer to?

R$_1$: What did Sam Thompson do next? (rank: Not Found)

C$_2$: What did Sam Thompson do before this?

R$_2$: What did Sam Thompson do after playing as a right fielder in Major League Baseball for the Detroit Wolverines, Philadelphia Phillies, and Detroit Tigers? (rank: 76)

C$_3$: What time period are you referring to for Sam Thompson's playing career?

R$_3$: What did Sam Thompson do after his playing career from 1884 to 1898 and his brief comeback in 1906? (rank: 31)

C$_4$: What specific actions or roles are you asking about regarding what Sam Thompson did after his playing career and brief comeback?

R$_4$: What specific roles or actions did Sam Thompson engage in after his playing career from 1884 to 1898 and his brief comeback in 1906? (rank: 54)

ICR-RRF (rank: 27) | ICR-PRRF (rank: 20)

Figure 9: Example on QReCC.

Given a conversation and a clarification question, the final query in the conversation may be ambiguous. Please rephrase the final query based on the clarification question, address the issue raised, and do not change the original meaning. You only need to output the rephrased query without any extra content. Here are some examples.
Examples:
#Clarification Question#:
Who does "she" refer to?
#Conversation#:
Q: Who produced the original show one foot in the grave?
A: Susan Belbin.
Q: Has she produced anything else?
#Rewritten Query#:
Has susan belbin produced anything else?

#Clarification Question#:
What does "anything else" exclude here?
#Conversation#:
Q: Who produced the original show one foot in the grave?
A: Susan Belbin.
Q: Has she produced anything else?
#Rewritten Query#:
Has she produced anything else besides one foot in the grave?

#Clarification Question#:
Where are the settlers referred to here?
#Conversation#:
Q: Where was the indian ocean mentioned above located?
A: Indian Ocean is the third-largest of the world's oceanic divisions, it is bounded by Asia to the north, Africa to the west and Australia to the east. To the south it is bounded by the Southern Ocean or Antarctica, depending on the definition in use. Along its core, the Indian Ocean has some large marginal or regional seas such as the Arabian Sea, the Laccadive Sea, the Somali Sea, Bay of Bengal, and the Andaman Sea.
Q: Who were the first settlers?
#Rewritten Query#:
Who were the first settlers of the indian ocean?

Please rephrase the last query in the conversation based on the clarification question below.
#Clarification Question#:
`{Clarification Question}`
#Conversation#:
`{Conversation}`
#Rewritten Query#:

Table 12: Prompt used in rewriting.