IterCQR: Iterative Conversational Query Reformulation with Retrieval Guidance

Anonymous ACL submission

Abstract

Conversational search aims to retrieve passages containing essential information to answer queries in a multi-turn conversation. In conversational search, reformulating context-dependent conversational queries into stand-alone forms is imperative to effectively utilize off-the-shelf retrievers. Previous methodologies for conversational query reformulation frequently depend on human-annotated rewrites. However, these manually crafted queries often result in sub-optimal retrieval performance and require high collection costs. To address these challenges, we propose **Iter**ative **C**onversational **Q**uery **R**eformulation **(IterCQR)**, a methodology that conducts query reformulation without relying on human rewrites. IterCQR iteratively trains the conversational query reformulation (CQR) model by directly leveraging information retrieval (IR) signals as a reward. Our IterCQR training guides the CQR model such that generated queries contain necessary information from the previous dialogue context. Our proposed method shows state-of-the-art performance on two widely-used datasets, demonstrating its effectiveness on both sparse and dense retrievers. Moreover, IterCQR exhibits superior performance in challenging settings such as generalization on unseen datasets and low-resource scenarios.

1 Introduction

In the conversational question answering (CQA) task, questions and answers are exchanged in a multi-turn conversation. As a component of CQA task, conversational search aims to retrieve passages that contain the necessary information to answer the current query within the conversation (Anantha et al., 2021; Adlakha et al., 2021).

Owing to the conversational setting, queries in CQA suffer from a high dependency on the previous conversation context, as shown in Figure 1, introducing challenges such as omissions, ambiguity, and coreference (Mao et al., 2023; Wang et al., 2023). Therefore, in conversational search, conversational queries cannot be directly used as inputs for off-the-shelf retrievers trained on non-contextual queries.

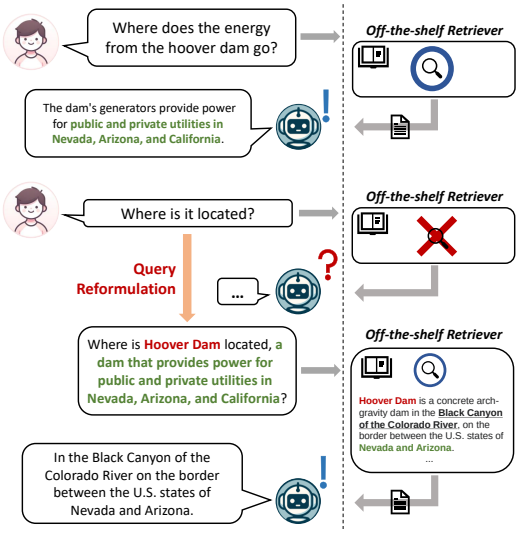

Figure 1: In the CQA task, the user's queries are dependent on the previous dialogue context. CQR task reformulates conversational queries into stand-alone queries, which are then fed into the off-the-shelf retrievers.

One possible strategy for mitigating this challenge is to train retrievers to comprehend long dialogue context (Yu et al., 2021; Kim and Kim, 2022; Lin et al., 2021a). However, this method results in substantial cost in retraining retrievers to handle long inputs. As an alternative method, researchers have explored conversational query reformulation (CQR) that reformulates conversational queries into stand-alone questions, which enables the utility of off-the-shelf retrievers (Mo et al., 2023; Voskarides et al., 2020a).

CQR methods can be categorized into rewriting and expansion. Most prior research trains models for query rewriting using human-annotated gold labels (Voskarides et al., 2020a; Del Tredici et al., 2022). However, these manually crafted queries often yield sub-optimal performance (Lin et al., 2021b; Wu et al., 2022) in addition to

requiring costly and time-consuming collection process. Human labels tend to focus on decontextualizing queries based on human subjective judgment, which does not always align with passage retrieval performance (Wu et al., 2022).

To address human rewrite's sub-optimality, ongoing research explores various expansion methods, such as potential answer expansion (Mo et al., 2023), and classifying previously mentioned entities for expansion (Del Tredici et al., 2022; Voskarides et al., 2020a). However, these expansions are not directly optimized for retrieval signals. Also, the existence of separate expansion and rewriting models requires additional training steps, storage, and two rounds of inference for a single query.

In this paper, we propose an **Iter**ative **C**onversational **Q**uery **R**eformulation (IterCQR) model, that iteratively performs query reformulation without using human rewrites. Since ground truth labels are unavailable, our approach employs an iterative framework to alternate between generating candidate queries and optimizing CQR model with their IR signals as a reward.

For the initialization of IterCQR, we leverage LLMs' ability to create an initial rewritten query dataset for training CQR model. After training IterCQR with the initial dataset, we iteratively train the model on generated query candidates through the Minimum Bayes Risk (MBR) (Smith and Eisner, 2006) training method and Top-1 candidate selection. In the training process. we integrate IR signal by defining the reward value as the cosine similarity between reformulated queries and ground-truth passages. After the iterative training process for IterCQR, we employ the final iteration model to reformulate queries, which are then utilized as inputs for off-the-shelf retrievers.

IterCQR achieves state-of-the-art performance on two widely used CQA datasets, the TopiOCQA (Adlakha et al., 2021) and QReCC (Anantha et al., 2021) datasets. We also show that IterCQR exhibits superior performance in various scenarios, such as generalization on unseen datasets and low-resource settings. Through a quantitative analysis of iterative reformulation, we experimentally demonstrate that IterCQR generates summary expansion from the preceding context as the iteration progresses. This expansion contributes to an improvement in retrieval performance, demonstrating the ability of IterCQR to generate retriever-friendly queries.

The main contributions of this work are as follows:

- We propose IterCQR which iteratively trains a conversational query reformulation model without human label while utilizing off-the-shelf retrievers.
- IterCQR exhibits state-of-the-art performance on both the TopiOCQA and QReCC datsets.
- IterCQR shows outstanding performance in challenging scenarios including generalization on unseen datasets and low-resource settings.

2 Related Works

2.1 Conversational Query Reformulation

CQR focuses on improving conversational search performance by rewriting and expanding user queries in a conversational context. In contrast to other conversational search methods, the reformulated queries in CQR can be directly utilized as input to off-the-shelf retrievers without fine-tuning.

Previous studies have addressed CQR by using human-rewritten queries or query expansion methods (Lin et al., 2020; Voskarides et al., 2020b; Yu et al., 2020). However, human rewrites have been reported to be sub-optimal (Lin et al., 2021b; Wu et al., 2022), and expansion methods require a separate model for term selection (Voskarides et al., 2020b; Kumar and Callan, 2020) or potential answer generation (Mo et al., 2023).

To address these shortcomings, ConQRR (Wu et al., 2022) employs Self-Critical Sequence Training (SCST) (Rennie et al., 2017) to directly optimize the query rewriting model to the retriever. More recently, ConvGQR (Mo et al., 2023) integrates query rewriting and expansion to further enhance retrieval performance; however, it requires two separate models for rewriting and expansion, which hampers efficiency in both the training and inference processes. Furthermore, these approaches require expensive human-rewritten queries. Recent works show that LLMs are capable of reformulating queries (Ma et al., 2023), including conversational queries (Mao et al., 2023; Wang et al., 2023; Dai et al., 2022). But LLM-generated queries also require further optimization for retrievers similar to human-rewritten queries.

Our work proposes an alternative approach of directly optimizing the CQR model to the retriever, using only gold passage annotation. We employ Minimum Bayes Risk (MBR) training (Smith and Eisner, 2006; Dasigi et al., 2019) and maximum

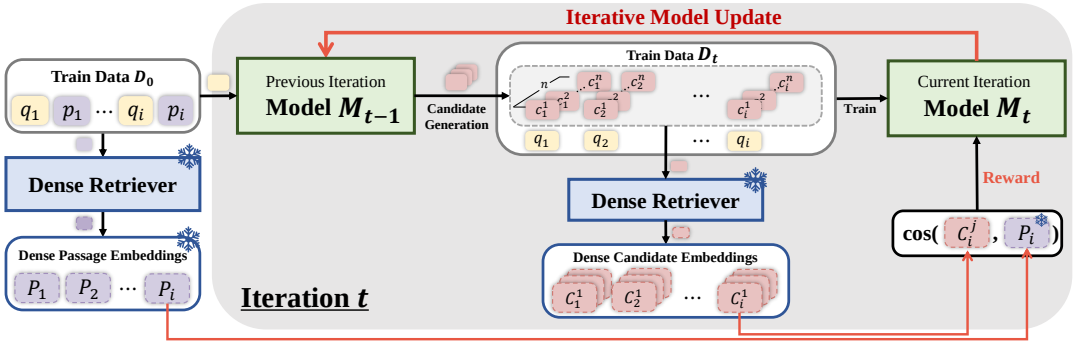

Figure 2: Overview of IterCQR. IterCQR trains on the candidates generated by the previous iteration model. We define reward as the cosine similarity between the frozen dense passage embeddings and dense candidate embeddings.

likelihood training based on the top-1 candidate to effectively learn without human annotated queries.

2.2 Iterative Learning in NLG tasks

The conventional method for training natural language generation (NLG) models uses human oracles, which are costly and time-consuming to collect; moreover, quality control during the collection is challenging. Therefore, research has been focused on learning without human supervision through iteratively enhancing the quality of the training dataset. Many works on weakly supervised QA and semantic parsing revolve around such iterative refinement paradigms, such as iterative search (Dasigi et al., 2019), ambiguous learning (Wu et al., 2023), and hard EM approach (Min et al., 2019). In task-oriented dialogue, Jang et al. (2022) propose to iteratively update the training set with self-generated samples. In this paper, we apply an iterative framework on the CQR task, which iteratively optimizes updated training samples with retrieval guidance.

3 Method

3.1 Problem Definition

Conversational search aims to retrieve relevant passages containing rich information that can answer the current conversational query q within a CQA system. To achieve this goal, conversational queries are reformulated so that we can utilize an off-the-shelf retriever R, which has been trained on non-conversational question-answering data.

We train a query reformulation model M to rewrite and expand the original query q based on the previous history context H to generate a de-contextualized query q^*. Training input for the query reformulation model M on turn k is $\{q_k, H_{k-1}\}$, concatenation of current query and the history context, where history context H_{k-1} is a consecutive sequence of previous queries and answers in reversed order. The reformulated query q_k^* from M is then used as input to the off-the-shelf retriever R, which retrieves a ranked list of the top-k relevant passages.

3.2 IterCQR

IterCQR utilizes the iterative setting to train CQR model without relying on human-rewritten queries. Specifically, our CQR model utilizes IR signals to generate the optimal query for retrieval tasks.

Algorithm 1: IterCQR

Input: Conversational query for k^{th} turn q_k, Previous history context H_{k-1}
Data: Train Data D without human-rewritten query
Model: CQR Model M_t for iteration t
Result: Reformulated query for k^{th} turn q_k^*, Query candidate for k^{th} turn c_k^j

```
for iteration t = 0..T do
    for q^k, H_{k-1} ∈ D do
        if t = 0 then                          // Initialize D_0
            q_k^* = LLM(q_k, H_{k-1})
            D_0 ← D_0 ∪ q_k^*
        else                                   // Generate D_t
            for j = 0..n do
                c_k^j = M_{t-1}(q_k, H_{k-1})
                D_t ← D_t ∪ c_k^j
            end
        end
    end
    if t = 0 then                              // Train
        L ← L_NLL with target q_k^*
    else if t ≤ τ (exploration factor) then
        L ← L_MBR with n candidates c_k^j
    else
        Select Top-1 Candidate c_i^top
        L ← L_NLL with target c_i^top
    end
    Train M_t with D_t to minimize L
end
```

We describe the training process of IterCQR in Algorithm 1. We first initialize IterCQR by training on the initial dataset D_0, which contains queries rewritten by LLM. After initializing the model, we go through the iterative process shown in Figure 2. During iteration t, we first utilize the previous iteration $t-1$ model M_{t-1} to generate n candidate queries for each instance in the training set D. Subsequently, this newly created training dataset, D_t, becomes the training data for the M_t model. In this iterative process, starting from M_1, candidates generated by the previous iteration model become the targets for training the current iteration model.

IterCQR leverages the cosine similarity between the dense embedding of the candidate query and gold passage to guide the CQR model to generate retriever-friendly queries. This reward prioritizes candidates with the most relevant semantic representation to the gold passage. Furthermore, for enhanced learning efficiency, we define exploration factor τ to balance the exploration and exploitation in the training phase. At the iterations less than or equal to τ, we employ the MBR training algorithm to facilitate exploration, followed by an exploitation phase using Top-1 candidate selection approach.

3.2.1 Data Initialization with LLM

We construct the initial dataset D_0 by utilizing LLM, `gpt-3.5-turbo`, to rewrite the queries.* The initial model M_0 is trained on D_0 with negative log-likelihood loss. Although LLMs show great abilities in various tasks, they still exhibit limitations in conversational query reformulation, considering that LLMs are not optimized for retrievers. Hence, starting from Iteration 1, we optimize IterCQR using IR signals.

3.2.2 MBR Training

In the early stages of the training, iterations less than or equal to τ, D_t contains diverse candidates that are suitable for exploration through the application of the Minimum Bayes Risk (MBR) training method. By employing all n candidates, the model can learn not only from the high-probability candidates but also from candidates with lower probability values. The MBR training algorithm seeks to minimize the expected value of a cost function $\mathcal{C}$ between input x and candidate y.

$$\min_{\theta} \sum_{i=1}^{N} \mathbb{E}_{\tilde{p}(y_i|x_i;\theta)} \mathcal{C}(x_i, y_i) \tag{1}$$

Here, we approximate the expectation using the re-normalized probabilities of the candidates obtained through beam search, denoted as $\tilde{p}$. To apply the MBR training algorithm for CQR model, we define a reward function $\mathcal{R}$ instead of the cost function $\mathcal{C}$. The MBR training loss is formulated to minimize the negative MBR term, as expressed in Equation 2:

$$\begin{aligned} \mathcal{L}_{MBR} &= -\sum_{i=1}^{N} \mathbb{E}_{\tilde{p}(c_i|q_i,H_{k-1};\theta)} \mathcal{R}(C_i, P_i) \\ &= -\sum_{i=1}^{N} \sum_{j=1}^{n} \tilde{\mathbb{P}}(c_i^j|q_i, H_{k-1}; \theta) \cdot \mathcal{R}(C_i^j, P_i). \end{aligned} \tag{2}$$

To employ the IR signal in the reward value, we set the reward function as the cosine similarity between the dense embedding of candidate query C_i and the dense embedding of the ground-truth passage P_i, as shown in Equation 3. We generate both passage and candidate query embeddings using the frozen encoder of the dense retriever.

$$\mathcal{R}(C_i, P_i) = \cos(C_i, P_i) \tag{3}$$

We observe that most reward values fall within a limited range, which could hamper providing fine-grained learning signals. To address this issue, we apply min-max normalization to scale the reward distribution into a range of 0 to 1.

By utilizing reward signals for all n candidate queries, the MBR approach enables learning across a diverse set of queries, ultimately resulting in significant improvements in the retrieval performance. After conducting query exploration through MBR iteration, the model proceeds to perform query exploitation by Top-1 candidate selection to effectively utilize the acquired knowledge.

3.2.3 Top-1 Candidate Selection

Following the standardization and enhancement of candidate quality through MBR training, we perform exploitation through Top-1 candidate selection. The exploitation objective aligns with exploration, reformulating conversational queries with retriever guidance. In this step, we select the top-1 candidate c^{top} among n candidates as the target for the training.

*The prompts used for initializing D_0 is shown the Appendix H.

In this process, the criteria for Top-1 candidate selection also rely on the cosine similarity between candidate embedding and the dense embedding of the gold passage. This selection criterion ensures alignment with the IR signal. We use the negative log-likelihood loss in the exploitation step to maximize the likelihood of generating top-1 candidate c_i^{top} as Equation 4:

$$\mathcal{L}_{NLL} = -\sum_{i=1}^{N} \log(\mathbb{P}(c_i^{top}|q_i, H_{k-1})). \quad (4)$$

Through this two-step approach, we aim to prevent queries from diverging too far from the semantic space of existing queries compared to training only with the MBR training algorithm for exploration. It also reduces computational complexity by eliminating the step of recalculating probabilities for all n candidates.

3.3 Retriever Models

We test IterCQR on both dense and sparse retrievers. Following the previous works (Mao et al., 2023, 2022; Mo et al., 2023; Yu et al., 2021), we use BM25 for sparse retriever and ANCE (Xiong et al., 2020) for the dense retriever. We generate dense embedding of gold passages and candidates using the ANCE dense retriever finetuned on MS MARCO (Nguyen et al., 2016). We store these embeddings and re-use them for the entire training and inference steps because the dense retriever is frozen from the beginning. We also use the ANCE model for encoding candidate queries' dense embeddings to calculate reward.

4 Experiments

Dataset We train and evaluate our model on QReCC (Anantha et al., 2021) and TopiOCQA (Adlakha et al., 2021). Both datasets consist of conversational queries and corresponding gold answers paired for each turn. Notably, TopiOCQA includes topic labels determined based on Wikipedia documents.

Evaluation Metrics Our model evaluates retrieval performance on commonly used metrics, such as mean reciprocal rank (MRR), NDCG@3, Recall@10, and Recall@100, following previous works (Wu et al., 2022; Mo et al., 2023; Anantha et al., 2021). We utilized the pytrec_eval tool (Gysel and de Rijke, 2018), as ConvGQR, to calculate these metrics in the subsequent experiments.

Baselines We compare our IterCQR model with seven baseline models: (1) **Raw**: This baseline represents the results obtained when using the original query of the data as input.[†] (2) **Initial**: The model M_0 trained by the initial dataset D_0 generated by `gpt-3.5-turbo`. (3) **GPT2QR (Transformer++)** (Anantha et al., 2021): GPT-2 (medium) (Radford et al., 2019) based QR model introduced in QReCC as a powerful baseline. (4) **QuReTeC** (Voskarides et al., 2020a): In this approach, query resolution is treated as a binary classification problem and trained to determine whether to include terms from the previous turn in the current query. (5) **T5QR** (Lin et al., 2020): Query reformulation model built upon the T5-base model (Raffel et al., 2020). (6) **CONQRR** (Wu et al., 2022): Leveraging reinforcement learning, CONQRR use retriever signals to generate queries optimized for the retriever. (7) **ConvGQR** (Mo et al., 2023): ConvGQR achieves strong performance through the integration of rewrite and potential answer expansion. The potential answer expansion model is trained using the gold answers from the dataset.

We directly report the results from ConvGQR paper for the baselines (RAW, GPT2QR, T5QR, and ConvGQR) and CONQRR results from the CONQRR paper.

Implementation Details We use T5-base as a backbone of the CQR model, and use ANCE dense retriever for the passage encoder which is kept frozen throughout all training iterations. For all iterations, the number of candidates is set to 10. We set τ as 1 for both the TopiOCQA and QReCC datasets. We use Adam optimizer with a learning rate of 1e-5, a batch size of 8, and a maximum query length of 32. See more details in Appendix A.

4.1 Main Results

We present the results of the IterCQR model trained on TopiOCQA and QReCC, respectively, in Table 1. Although other baseline models were trained or utilized human rewrites for training, IterCQR demonstrates superior performance even without using a human rewrite.

Notably, IterCQR outperforms the second-best performing model, ConvGQR, with a significant

[†]Raw results for TopiOCQA are the result of the model trained on QReCC since TopiOCQA does not have a human rewrite.

Type	Method	TopiOCQA				QReCC			
		MRR	NDCG@3	R@10	R@100	MRR	NDCG@3	R@10	R@100
Dense (ANCE)	Raw	0.041	0.038	7.5	13.8	0.102	0.093	15.7	22.7
	Initial	0.178	0.168	32.6	47.7	0.358	0.330	55.7	74.1
	GPT2QR	0.126	0.120	22.0	33.1	0.339	0.309	53.1	72.9
	QuReTeC	0.112	0.105	20.2	34.4	0.350	0.326	55.0	70.9
	T5QR	0.230	0.222	37.6	54.4	0.345	0.318	53.1	72.8
	CONQRR	-	-	-	-	0.418	-	65.1	**84.7**
	ConvGQR	0.256	0.243	41.8	58.8	0.420	0.391	63.5	81.8
	IterCQR	**0.263**	**0.251**	**42.6**	**62.0**	**0.429**	**0.402**	**65.5**	84.1
	Human-Rewrite	-	-	-	-	0.384	0.356	58.6	78.1
Sparse (BM25)	Raw	0.021	0.018	4.0	9.2	0.065	0.055	11.1	21.5
	Initial	0.132	0.115	25.2	47.3	0.322	0.290	51.8	81.2
	GPT2QR	0.062	0.053	12.4	26.4	0.304	0.279	50.5	82.3
	QuReTeC	0.085	0.073	16.0	31.3	0.340	0.305	55.5	86.0
	T5QR	0.113	0.098	22.1	44.7	0.334	0.302	53.8	86.1
	CONQRR	-	-	-	-	0.383	-	60.1	**88.9**
	ConvGQR	0.124	0.107	23.8	45.6	0.441	0.410	**64.4**	88.0
	IterCQR	**0.165**	**0.149**	**29.3**	**54.1**	**0.467**	**0.441**	**64.4**	85.5
	Human-Rewrite	-	-	-	-	0.397	0.362	62.5	98.5

Table 1: Performance of IterCQR on TopiOCQA and QReCC dataset using dense and sparse retriever. We utilize ANCE for the dense retriever and BM25 for the sparse retriever. **Bold** letters indicate the best performance of reported results; human-rewrite is excluded in this comparison. Note that CONQRR used DualEncoder instead of ANCE.

TopiOCQA Test					
Test	Method	Dense		Sparse	
		MRR	NDCG@3	MRR	NDCG@3
ID	**ConvGQR**	0.256	0.243	0.124	0.107
OOD	**IterCQR**	0.178	0.164	0.137	0.122
QReCC Test					
Test	Method	Dense		Sparse	
		MRR	NDCG@3	MRR	NDCG@3
ID	**CONQRR**	0.418	-	0.383	-
	ConvGQR	0.420	0.391	0.441	0.410
OOD	**IterCQR**	0.401	0.374	0.449	0.424

Table 2: IterCQR performance on unseen datasets. CONQRR and ConvGQR are evaluated in an in-domain (ID) setting, while IterCQR is evaluated in an out-of-domain (OOD) setting.

improvement in the results of the dense retriever on TopiOCQA. Although the reward function of IterCQR is defined in terms of dense passage embeddings, the model exhibits a significant performance in the sparse retriever. In fact, our model surpasses the retrieval performance of directly using human rewrites.

4.2 Generalization on Unseen Datasets

In this section, we show the IterCQR's generalization ability on unseen datasets. We train IterCQR on TopiOCQA and evaluate on QReCC test set and vice versa. The results of these experiments are presented in Table 2.

IterCQR, which was solely trained on QReCC and tested in an out-of-domain(OOD) setting for the TopiOCQA test set, outperforms the in-domain(ID) model in sparse retrieval results, across all evaluation metrics. QReCC-trained IterCQR outperforms TopiOCQA-trained ConvGQR on the TopiOCQA test set in sparse retrieval performance. Evaluating on the QReCC test set, the TopiOCQA-trained IterCQR shows comparable performance to the ID setting models for the dense passage retriever. Notably, in the case of sparse retrieval performance on the QReCC test set, despite being tested in an OOD setting, TopiOCQA-trained IterCQR outperforms QReCC-trained ConvGQR and ConQRR, both of which follow the ID setting. These experiments underscore IterCQR's strong generalization capabilities across diverse datasets.

4.3 Low-resource Setting

To evaluate the performance of IterCQR in a low-resource scenario, we demonstrate the results of models trained on 20% and 50% of the entire TopiOCQA train set in Table 3. Remarkably, even when utilizing only 50% of the full TopiOCQA training data, the models achieve comparable performance to those trained with the entire dataset. Furthermore, compared to ConvGQR, the second best-performing model in the main results in Table 1, both the 20% and 50% models surpass ConvGQR on sparse retrieval results and perform comparably in dense retrieval results. This observation shows

Type	Data	TopiOCQA			
		MRR	NDCG@3	R@10	R@100
Dense	20%	0.204	0.189	35.4	55.6
	50%	0.252	0.242	41.6	58.8
	100%	**0.263**	**0.251**	**42.6**	**62.0**
Sparse	20%	0.144	0.128	25.5	51.7
	50%	0.162	0.145	28.7	52.9
	100%	**0.165**	**0.149**	**29.3**	**54.1**

Table 3: Performance of IterCQR in a low resource scenario. We train IterCQR using 20%, 50%, and 100% of the TopiOCQA train dataset.

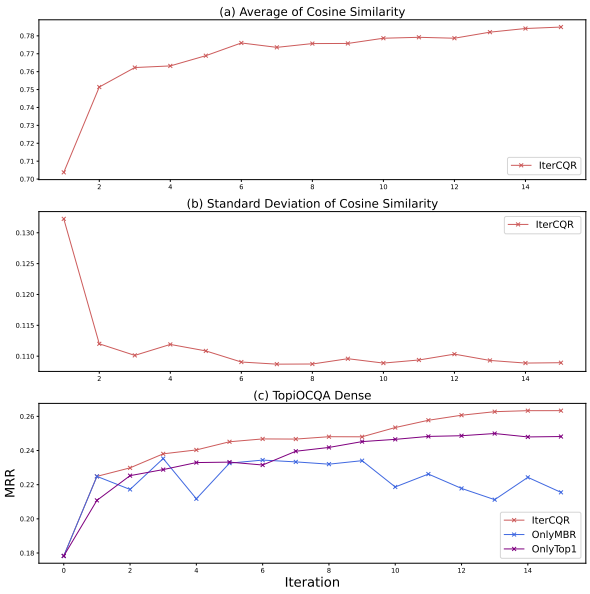

Figure 3: OnlyMBR is the model trained only with the MBR algorithm, and OnlyTop1 is trained only with the Top-1 candidate selection.

that IterCQR is effective in low-resource scenarios, consistently exhibiting state-of-the-art performance even when trained with a limited amount of training data.

4.4 Ablation Study

In this work, we have presented a two-step training approach; MBR algorithm for exploration, and Top-1 candidate selection for exploitation. We conduct an ablation study on the core components of training IterCQR.

In Table 4, it is illustrated that both exploration and exploitation are required; models trained with either one of the components display noticeable drops across metrics tested with both dense and sparse retrievers. This clearly indicates that IterCQR requires two-step training approach, where the model explores-and-exploits the query space.

We argue the superiority of the two-step training approach derives from the fact that the MBR

Type	Method	TopiOCQA			
		MRR	NDCG@3	R@10	R@100
Dense	IterCQR	**0.263**	**0.251**	**42.6**	**62.0**
	OnlyMBR	0.216	0.204	35.6	53.6
	OnlyTop1	0.248	0.234	41.6	61.1
Sparse	IterCQR	**0.165**	**0.149**	**29.3**	**54.1**
	OnlyMBR	0.111	0.099	20.0	52.9
	OnlyTop1	0.150	0.134	26.1	48.1

Table 4: Retrieval performance of IterCQR, OnlyMBR, and OnlyTop1 on TopiOCQA dataset.

training method is particularly effective when a diverse range of candidate qualities exists. After adequate training with MBR, most of the n candidates exhibit high reward values, leading to an unstable training process because of the tendency to penalize candidates with relatively lower reward values, even if the candidates possess good quality. We observe that, in Figure 3 (a), the average cosine similarities of candidates exhibit a noticeable increase after the iteration with MBR training method, whereas the standard deviation decreases significantly as in Figure 3 (b).

Furthermore, in Figure 3 (c) we observe that IterCQR consistently enhances dense retrieval performance, while OnlyMBR, the model trained with only the MBR training algorithm, exhibits unstable learning. Moreover, the OnlyTop1 model, trained solely with Top-1 candidate selection, results in slower learning and a lower performance saturation point than IterCQR. Therefore, it is advisable to initially utilize MBR for exploration, and once the variance in cosine similarity values among candidates has decreased, switch to the exploitation with Top-1 candidate selection to achieve more stable learning and facilitate an efficient query search. See Appendix E for generated queries of IterCQR, OnlyMBR, and OnlyTop1.

5 Analysis

We analyze the effect of iterative rewriting on IterCQR by each iteration. For each iteration, we measure the retrieval performance, token length of the rewritten queries, token overlap with the historical context, and token overlap with the gold passage.

5.1 Effect on Retrieval Performance

We show the retrieval performance of the TopiOCQA-trained model for each iteration in Figure 4. In both the TopiOCQA and QReCC datasets, there is a notable improvement in the MRR and Recall@10 metric as the iterations progress. In particular, applying the MBR training algorithm

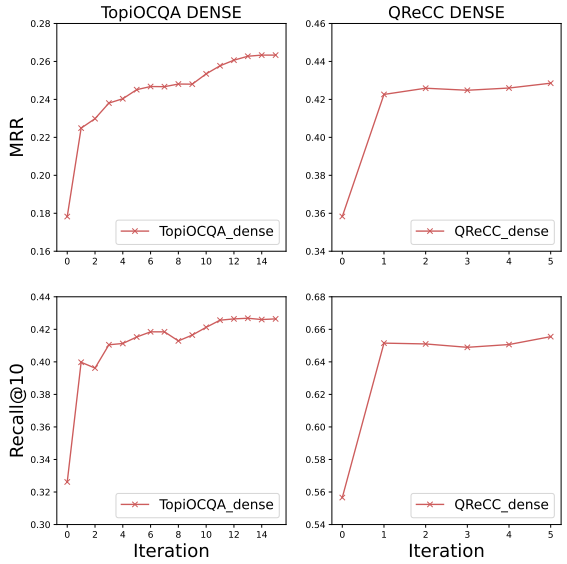

Figure 4: IterCQR dense retrieval performance on TopiOCQA and QReCC datasets for each iteration.

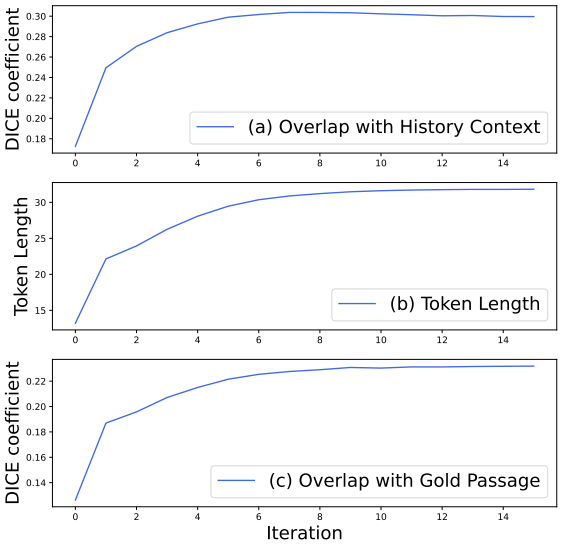

Figure 5: Effect of iterative setting on queries. Overlapping tokens in (a) and (c) is shown by the Sørensen-Dice coefficient, and (b) is reported by the average token length of the reformulated queries in the TopiOCQA test set.

in iteration 1 significantly enhances performance. This outcome suggests that the IterCQR effectively explored the query space, leading to significant improvements in retrieval performance.

5.2 Effect on Query

We analyze the characteristics of the reformulated queries by IterCQR for each step trained with TopiOCQA and present the results in Figure 5. We utilize the Sørensen-Dice coefficient (Sorensen, 1948; Dice, 1945) in Equation 5 to measure the similarity of two strings.

$$D(A, B) = \frac{2 * |A \cap B|}{|A| + |B|} \quad (5)$$

As the iterations progress, we observe a consistent increase in token overlap with historical context as shown in 5 (a). Ultimately, this trend signifies that the IterCQR model progressively learns to extract information from historical context and integrates it into the current conversational query.

Furthermore, when examining the average query token length for each iteration, as shown in Figure 5 (b), evidently, the token length increases with each iteration. These findings from Figure 5 (a) and (b) suggest that the IterCQR model learns to summarize the previous context.

To validate whether this summarization is helpful for retrieval, we measure the overlapping tokens with the gold passage using the Dice coefficient. As depicted in Figure 5 (c), we consistently observe an increase in the token overlap with the gold passage. This implies that IterCQR's queries ultimately provides a stronger retrieval signal towards the gold passage, thereby contributing to better retrieval performance, as shown in Figure 4. We provide generated queries by each iteration in Appendix F.

Additionally, in Figure 5, we consistently observe a sharp increase in iteration that employ MBR training method, specifically iteration 1. This sharp increase aligns with the pattern of sharp increases in the retrieval performance in Figure 4. This pattern shows the effectiveness of the MBR training algorithm for exploration.

6 Conclusion

In this work, we propose IterCQR, a methodology that iteratively improves CQR model without relying on human rewrites. IterCQR leverages retrieval signals when training CQR model, which provides retriever-friendly guidance for CQR. We demonstrate the effectiveness of IterCQR through state-of-the-art performance on the QReCC and TopiOCQA datasets. In addition, the experimental results indicate that IterCQR learns to summarize previous contextual history, which leads to improved retrieval performance as the iteration progresses. Furthermore, IterCQR exhibits superior performance in challenging settings such as generalization on unseen datasets and low-resource setting.

Limitations

IterCQR employs the LLM, specifically `gpt-3.5-turbo`, to create the rewritten queries in initial dataset D_0 for training CQR model. However, utilizing LLM to generate a rewrite requires inference costs. Furthermore, the IterCQR initial performance relies on the LLM's performance. Still, we show that IterCQR can maintain its retrieval performance using only 50% of the entire dataset, which could improve data efficiency and save LLM inference costs.

Since IterCQR generates n candidates for each training instance, it necessitates larger storage capacity. Additionally, the iterative framework can lead to relatively longer training times, though it does not require additional cost in the inference time.

IterCQR leverages dense embedding information as a reward term. Consequently, as the iterative learning process continues, it becomes increasingly optimized for dense retriever performance. However, the dense reward signal may not consistently enhance retrieval performance for sparse retrievers. Nonetheless, it is worth highlighting that IterCQR outperforms powerful baseline methods in terms of sparse retrieval performance, despite using dense retrieval reward.

Ethical Statement

We conducted experiments utilizing publicly available datasets, all of which are in English. Our CQR model tends to generate summary expansions of the previous dialogue history. These expansion terms are dependent on the contextual history of the dialogue. Therefore, if there is bias or inappropriate statements in the previous history context, the generated queries may also potentially contain such information.

References

Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2021. Topiocqa: Open-domain conversational question answeringwith topic switching. *CoRR*, abs/2110.00768.

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.

Zhuyun Dai, Vincent Y. Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B. Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples.

Pradeep Dasigi, Matt Gardner, Shikhar Murty, Luke Zettlemoyer, and Eduard Hovy. 2019. Iterative search for weakly supervised semantic parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2669–2680, Minneapolis, Minnesota. Association for Computational Linguistics.

Marco Del Tredici, Xiaoyu Shen, Gianni Barlacchi, Bill Byrne, and Adrià de Gispert. 2022. From rewriting to remembering: Common ground for conversational QA models. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 70–76, Dublin, Ireland. Association for Computational Linguistics.

Lee R Dice. 1945. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.

Christophe Van Gysel and Maarten de Rijke. 2018. Pytrec_eval: An extremely fast python interface to trec_eval. *CoRR*, abs/1805.01597.

Youngsoo Jang, Jongmin Lee, and Kee-Eung Kim. 2022. GPT-critic: Offline reinforcement learning for end-to-end task-oriented dialogue systems. In *International Conference on Learning Representations*.

Sungdong Kim and Gangwoo Kim. 2022. Saving dense retriever from shortcut dependency in conversational search.

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, Online. Association for Computational Linguistics.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021a. Contextualized query embeddings for conversational search. *CoRR*, abs/2104.08707.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

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

Kelong Mao, Zhicheng Dou, Fengran Mo, Jiewen Hou, Haonan Chen, and Hongjin Qian. 2023. Large language models know your contextual search intent: A prompting framework for conversational search.

Kelong Mao, Zhicheng Dou, and Hongjin Qian. 2022. Curriculum contrastive context denoising for few-shot conversational dense retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 176–186.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. A discrete hard EM approach for weakly supervised question answering. 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 2851–2864, Hong Kong, China. 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.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

David A. Smith and Jason Eisner. 2006. Minimum risk annealing for training log-linear models. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 787–794, Sydney, Australia. Association for Computational Linguistics.

Thorvald Sorensen. 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Biologiske skrifter*, 5:1–34.

Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020a. Query resolution for conversational search with limited supervision. *CoRR*, abs/2005.11723.

Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020b. 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 '20, page 921–930, New York, NY, USA. Association for Computing Machinery.

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

Shan Wu, Chunlei Xin, Hongyu Lin, Xianpei Han, Cao Liu, Jiansong Chen, Fan Yang, Guanglu Wan, and Le Sun. 2023. Ambiguous learning from retrieval: Towards zero-shot semantic parsing. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14081–14094, Toronto, Canada. Association for Computational Linguistics.

Zeqiu Wu, Yi Luan, Hannah Rashkin, David Reitter, Hannaneh Hajishirzi, Mari Ostendorf, and Gaurav Singh Tomar. 2022. CONQRR: Conversational query rewriting for retrieval with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 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. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *CoRR*, abs/2007.00808.

Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. Few-shot generative conversational query rewriting. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1933–1936, New York, NY, USA. Association for Computing Machinery.

Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-shot conversational dense retrieval. *CoRR*, abs/2105.04166.

A Implementation Details

In this work, we evaluate IterCQR on widely-used conversational search datasets. TopiOCQA dataset consists of 3,920 conversations with average of 13 question-answer turns for each conversation. For the TopiOCQA dataset, IterCQR is trained for 5 epochs for the initial M_0 model, 2 epochs for MBR, and 5 epochs for the rest of the Top-1 candidate selection. The TopiOCQA-trained model is trained over 15 iterations.

QReCC dataset includes 13.6k conversations and each conversation consists of 6 turns in average. For the QReCC dataset, IterCQR is trained for 10 epochs for the M_0 model, MBR training for 2 epochs, and 5 epochs for Top-1 candidate selection. The final results of the QReCC-trained model are obtained after 5 iterations. For the experiments, we report the result with a single run of IterCQR, because it is costly to generate multiple initial dataset with LLM. All experiments are conducted using a single Nvidia A6000 GPU. Training time differs by the training method: MBR training requires about 5 hours for one epoch, and Top-1 candidate selection requires about 40 minutes. We utilize T5-base as a backbone of the CQR model, which consists of 220M trainable parameters.

B Comparison between MBR and SCST

Type	Method	TopiOCQA			
		MRR	NDCG@3	R@10	R@100
Dense	SCST-20	20.6	19.2	35.6	55.4
	SCST-50	19.9	19.0	34.6	52.6
	MBR	**22.5**	**21.1**	**40.0**	**58.9**
Sparse	SCST-20	13.9	11.2	27.2	51.5
	SCST-50	14.4	12.6	26.9	50.5
	MBR	**16.4**	**14.8**	**30.2**	**54.8**

Table 5: Comparison between SCST and MBR training algorithm. In the case of SCST, sampling was employed during candidate generation while MBR utilized beam search. We report the results considering two variations in top-k sampling: top-k=20 and 50.

In this section, we compare the Self-Critical Sequence Training (SCST) utilized in CONQRR with our MBR training in IterCQR. In the case of the SCST, sampling was employed during candidate generation, and we report two variations in top-k sampling: top-k=20 and top-k=50. The outcomes presented in Table 5 indicate the performance of M_1, trained using the MBR training algorithm and the SCST algorithm, starting from the same initial model M_0. Evidently, MBR is a more effective algorithm for CQR, outperforming both dense and sparse retrievers for all evaluated metrics.

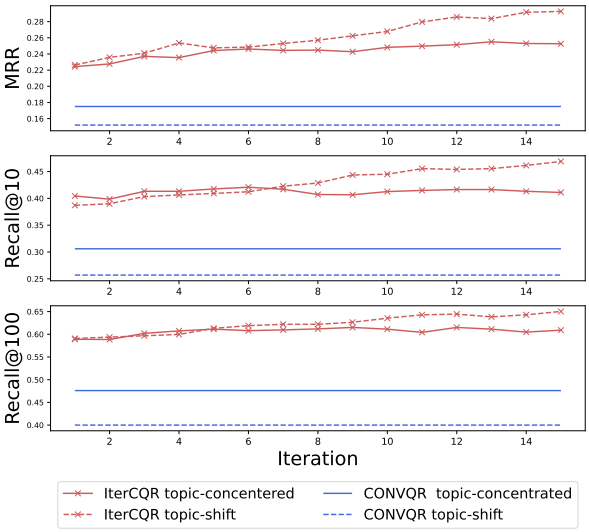

Figure 6: Performance of TopiOCQA-trained IterCQR on both topic-shifted and topic-concentrated examples. The TopiOCQA test set was divided by topic-shifted and topic-concentrated samples based on the topic label from the dataset. We report the retrieval performance of each iteration with metrics MRR, Recall@10, and Recall@100, respectively.

C Topic-shift Analysis

CQA differs from standard Question Answering (QA) because of its interactive and conversational nature, introducing the concept of topic-shift during the conversation. This topic-shift phenomenon has been recognized as a challenging aspect for models in various studies (Adlakha et al., 2021; Wu et al., 2022). To assess the capability of IterCQR to handle turns with topic shifts, we conducted an analysis of two distinct criteria associated with topic shifts.

C.1 Topic-shift by Topic Label

We divide the TopiOCQA dataset into two categories: topic-shift and topic-concentrated, based on the assigned topic labels from the dataset. In this categorization, we determine a topic-shifted instance if the topic label in the current turn differs from the topic label from the immediately preceding turn. According to this criterion, the dataset is divided as topic-concentrated for 73% , and topic-shift for 27%.

We evaluate the performance of the TopiOCQA-trained IterCQR with a dense retriever in address-

Model	IR	Topic-Concentrated			Topic-Shifted		
		MRR	Recall@10	Recall@100	MRR	Recall@10	Recall@100
T5QR	BM25	0.352	54.4	84.0	**0.252**	45.1	79.1
CONQRR(mix)	BM25	0.419	63.1	**91.2**	**0.252**	45.9	**82.1**
CONQRR(RL)	BM25	0.444	66.2	90.3	0.233	44.5	78.4
IterCQR	BM25	**0.544**	**72.4**	89.7	0.249	**49.7**	77.7
Human Rewrite	BM25	0.440	66.7	98.8	0.318	56.7	98.4

Table 6: Performance of QReCC-trained IterCQR on topic-concentrated and topic-shifted samples. In this experiment, topic shift was determined by gold passage ID in the QReCC dataset. For a fair comparison with CONQRR, we only report the performances on sparse retrieval.

ing topic-shift scenarios as shown in Figure 6. We observe a consistent improvement in performance across iterations for both topic-shifted and topic-concentrated scenarios. Comparing our results with the performance of ConvGQR that we have reproduced, clearly, IterCQR outperforms ConvGQR in both topic-concentrated and topic-shifted scenarios, presenting our model's superior performance. ConvGQR performs better with topic-concentrated samples than with topic-shift instances, proving that topic-shifted turns are more difficult for the retriever. However, with the MRR metric, IterCQR performs better on topic-shifted instances than on topic-concentrated cases. In addition, in terms of Recall@10 and Recall@100, IterCQR initially shows a better performance on topic-concentrated cases; however, as the iterations progress, topic-shifted cases surpass that in the topic-concentrated cases. This observation highlights the significant influence of IterCQR's cosine similarity reward based on dense representation, emphasizing performance improvement through iterative reformulation.

C.2 Topic-shift by Gold Passage ID

For the second criterion of topic-shift, we divide the QReCC test set based on the gold passage IDs within the dataset. In this setting, if the gold passage ID of the current turn does not appear in any preceding turn within the same conversational session, a topic-shift is considered to have occurred. According to this criterion, topic-concentrated instances account for 30% of the dataset, whereas topic-shifted samples constitute 70%.

Table 6 provides a comparison of the results of topic-shifted cases determined by gold passage IDs. The scores reported in the CONQRR paper are presented in Table 6. Note that it is hard to fairly compare IterCQR and CONQRR on dense retriever performance because CONQRR used DualEncoder instead of ANCE. Hence, in this experiment, we compare the results on the sparse retriever.

The results indicate that IterCQR outperforms CONQRR across all instances, in both the topic-concentrated and topic-shifted scenarios, particularly in terms of the MRR metric. Especially, MRR scores of IterCQR are notably superior, even surpassing the human rewrite performance. IterCQR demonstrates comparable performance to CONQRR in addressing topic-shifted instances. However, overall, human rewrites outperform all the models, demonstrating exceptional robustness in topic-shift scenarios.

Dataset	Method	MRR	NDCG@3	R@10
TopiOCQA	IterCQR	0.263	0.251	42.6
	IterCQR+ expansion	0.277	0.264	44.6
QReCC	IterCQR	0.429	0.402	65.5
	IterCQR+ expansion	0.444	0.417	67.3

Table 7: Performance with potential answer expansion.

D IterCQR with Other Expansions

In this experiment, we show that the factor of performance improvement in IterCQR is orthogonal to that of potential answer expansion introduced in ConvGQR(Mo et al., 2023). For TopiOCQA and QReCC, we concat the reformulated query generated by the IterCQR with an expansion term trained on gold answers of the dataset. As shown in Table 7, with the potential answer expansion, there is an additional improvement in the retrieval performance, even though IterCQR has already achieved state-of-the-art performance on both datasets.

E Qualitative Analysis on Ablation Study

In section 4.4, we demonstrate that IterCQR exhibits superior performance when compared to solely using MBR training method and Top-1 candidate selection. We provide comparative anlaysis

on the queries generated by OnlyMBR and OnlyTop1 model with those generated by IterCQR in Table 8. Reformulated queries produced by IterCQR contain essential information that effectively directs to the gold passage, consequently yielding significant improvements in retrieval performance.

F Effect of Iterative Setting on Queries

In section 5.2, we empirically show that IterCQR learns to generate summary expansion of the preceding history context as the iteration progresses. In Table 9, we provide examples of generated queries at different intermediate iterations of IterCQR, specifically at iterations 0, 1, 5, and 15. Evidently, IterCQR progressively acquires the capacity to distill essential information from the previous dialogue context, resulting in a higher token overlap with the gold passage and retrieval performance.

Furthermore, IterCQR is capable of fixing the reformulation errors that exist in the earlier iteration of the IterCQR model. In Table 1, the retrieval performance of M_0 trained on LLM-generated initial dataset D_0 is far inferior compared to our final IterCQR model. We illustrate instances in Table 10 which IterCQR mitigates the errors in M_0 as iteration progresses.

G Queries Generated by IterCQR

In this section, we present the queries generated by IterCQR on both the TopiOCQA and QReCC dataset in Table 11 and Table 12, respectively. Note that the TopiOCQA dataset does not have a human rewrite; therefore, we demonstrate a reformulated query generated by a model trained on the QReCC dataset's human rewrite.

H Prompt Used for Dataset initilaization

We report the prompt used for generating D_0 using `gpt-3.5-turbo`. We use the instruction: *"This is a part of conversational question answering. Rewrite the current query as a stand-alone question based on the previous conversation so that it could be context-independent."*. We concat three preceding original queries and answers for the dialogue history context and the current query as the input for the LLM.

I Failure Case

We demonstrate the failure case of IterCQR. Training the IterCQR with cosine similarity with the gold passage leads the model to increase the keyword overlap with the gold passage. However, this could also result in the repetition of the keyword in the reformulated query, as shown in Table 13.

Type	Content
Original Query	what was the series about?
OnlyTop1 Query	What is Grey's Anatomy about in the series "Grey's Anatomy" broadcasted by the American Broadcasting
OnlyMBR Query	What is "Grey's Anatomy", an American television series produced in 2005 and featuring several films including Melissa and George and Alex Proy
IterCQR Query	What is Grey's Anatomy, a television series premiered on **March 27, 2005** by **the American Broadcasting Company (ABC)**.
Gold Passage	Grey's Anatomy Introduction Grey's Anatomy is an American medical drama television series that premiered on **March 27, 2005**, on **the American Broadcasting Company (ABC)** as a mid-season replacement. The fictional series focuses on the lives of surgical interns, residents, and attendings as they develop into seasoned doctors ...
Original Query	who directed it?
OnlyTop1 Query	Who directed Ride Me to Hell, the episode "Ride Me to Hell," the third episode of the American animated television series "Ugly Americans,"
OnlyMBR Query	Who directed "Ride Me to Hell", an episode of American animated television series featuring teenage lead characters including John and Julie, and Ryan in 1992 and ending
IterCQR Query	Who directed Ride Me to Hell, the episode of the American animated television series "Ugly Americans", which aired on **July 14, 2011** titled
Gold Passage	Ride Me to Hell Introduction "Ride Me to Hell" is the third episode of the of the American animated television series "Ugly Americans", and the seventeenth overall episode of the series. It originally aired on Comedy Central in the United States on **July 14, 2011**. In the episode, ...

Table 8: Generated queries by TopiOCQA-trained onlyTop1, onlyMBR, and IterCQR. Red words represnt rewritten entity and blue words show the summary expansion included only in IterCQR.

Type	Content
Original Query	name the structural parts of this.
IterCQR (0)	What are the structural parts of the Milky Way galaxy?
IterCQR (1)	What are the structural components of the Milky Way galaxy, which contains the Solar System?
IterCQR (5)	What are the structural components of the Milky Way galaxy, which contains ten billion white dwarfs, a billion neutron stars, and a
IterCQR (15)	What are the structural components of the Milky Way, which is the **second-largest galaxy in the Local Group with its stellar disk approximately in diameter**
Gold Passage	'Milky Way Size and mass The Milky Way is the **second-largest galaxy in the Local Group** (after the Andromeda Galaxy), **with its stellar disk approximately in diameter** and, on average, approximately thick. The Milky Way is approximately 890 billion times the mass of the Sun. To compare the relative physical scale of the Milky Way, if the Solar System out to Neptune were the size of a US quarter (), the Milky Way would be approximately the size of the contiguous United States. There is a ring-like filament of stars rippling above and below the relatively flat galactic plane, wrapping around the Milky Way at a diameter of , which may be part of the Milky Way itself.
Original Query	where is it located?
IterCQR (0)	Where is the aforesaid administration located?
IterCQR (1)	Where was the National Oceanic and Atmospheric Administration (NOAA) located in 1970?
IterCQR (5)	Where was the U.S. National Oceanic and Atmospheric Administration (NOA) located during its formation on October 3, 1970?
IterCQR (15)	Where is the U.S. National Oceanic and Atmospheric Administration (NOAA), **an American scientific agency**, formed on October 3, 1970
Gold Passage	National Oceanic and Atmospheric Administration History NOAA traces its history back to multiple **agencies**, some of which were among the oldest in the federal government: The most direct predecessor of NOAA was the Environmental Science Services Administration (ESSA), into which several existing **scientific agencies** such as the United States Coast and Geodetic Survey, the Weather Bureau and the uniformed Corps were absorbed in 1965. NOAA was established within the Department of Commerce via the Reorganization Plan No. 4 and formed on October 3, 1970, after U.S. President Richard Nixon proposed creating a new agency to serve a national need for "better protection of life and property from natural hazards... for a better understanding of the total environment... [and] for exploration and development leading to the intelligent use of our marine resources

Table 9: Queries generated by TopiOCQA-trained IterCQR on intermediate iterations. IterCQR(t) represents the model on iteration t. Red words note rewritten entity and blue words show the summary expansion included only in IterCQR.

Type	Content
Original Query	what was the series about?
IterCQR (0)	What is the plot of **the series "Dark City"**?
IterCQR (1)	What was **the series "Grey's Anatomy"** about primarily about sports programming primarily on weekend afternoons
IterCQR (15)	What is **Grey's Anatomy**, a television series premiered on March 27, 2005 by the American Broadcasting Company (ABC).
Gold Passage	Grey's Anatomy Introduction Grey's Anatomy is an American medical drama television series that premiered on March 27, 2005, on the American Broadcasting Company (ABC) as a mid-season replacement. The fictional series focuses on the lives of surgical interns, residents, and attendings as they develop into seasoned doctors while balancing personal and professional relationships. The title is an allusion to "Gray's Anatomy", a classic human anatomy textbook first published in 1858 in London and written by Henry Gray. Shonda Rhimes developed the pilot and continues to write for the series. She is also one of the executive producers alongside Betsy Beers, Mark Gordon, Krista Vernoff, Rob Corn, Mark Wilding, and Allan Heinberg and recently Ellen Pompeo.

Table 10: IterCQR mitigates initial model's error. IterCQR(t) represents the model on iteration t. Red words show the coreference from the original query, orange words show the initial model M_0 error and blue words show the correct rewritten entity by IterCQR.

TopiOCQA Dataset	
Previous Turns	Query: What is the symbol of flag of ecuador? Answer: It **consists of horizontal bands of yellow (double width), blue and red.** Query: Who designed it? Answer: UNANSWERABLE
Original Query	Does it resemble any other flag?
Human Rewrite	Does flag of ecuador resemble any other flag
LLM Rewrite	Does the flag of Ecuador resemble any other flag?
IterCQR Query	What are some other flags that resemble the flag of Ecuador, which is **consists of horizontal bands of yellow (double width), blue, and red**
Gold Answer	Yes, Colombia and Venezuela
Gold Passage	Flag of Ecuador Introduction The national flag of Ecuador, which **consists of horizontal bands of yellow (double width), blue and red**, was first adopted by law in 1835 and later on 26 September 1860. The design of the current flag was finalized in 1900 with the addition of the coat of arms in the center of the flag. Before using the **yellow, blue and red** tricolor, Ecuador used white and blue flags that contained stars for each province of the country. The design of the flag is very similar to those of **Colombia and Venezuela**, which are also former constituent territories of Gran Colombia.

Table 11: Reformulated Queries by IterCQR on TopiOCQA dataset.The green and red words stand for overlap with previous context and rewritten entity. Underlined words notate the content that contains the gold answer for the given query. Note that human rewrite in the TopiOCQA dataset refers to the output of the model trained on the QReCC dataset's human oracle.

QReCC Dataset	
Previous Turns	Query: What is the role of work cover nsw Answer: The agency **WorkCover** NSW creates **regulations to promote productive, healthy and safe workplaces for workers in New South Wales**. Query: What else does the agency do Answer: The agency WorkCover NSW created **regulations** for employers too.
Original Query	Was there any controversy with the agency
Human Rewrite	Was there any controversy with nsw
LLM Rewrite	Has the agency WorkCover NSW faced any controversy?
IterCQR Query	Was there any controversy surrounding the agency **WorkCover** NSW's **regulations** to promote productive, **healthy and safe workplaces** for workers in **New South Wales.**
Gold Answer	In December 2005, the Independent Commission Against Corruption found that 23 NSW WorkCover employees had issued false certificates of competency.
Gold Passage	Dangerous Goods (Gas Installations) **Regulation** 1998 (NSW) Dangerous Goods (Road and Rail Transport) **Regulation** 2009 (NSW) Explosives **Regulation** 2005 (NSW) Occupational **Health** and **Safety Regulation** 2001 (NSW) Sporting Injuries Insurance **Regulation** 2009 (NSW) Workers Compensation (Bush Fire, Emergency and Rescue Services) **Regulation** 2007 (NSW) Workers' Compensation (Dust Diseases) **Regulation** 2008 (NSW) Workers Compensation **Regulation** 2010 (NSW) ... **In December 2005, the Independent Commission Against Corruption found that 23 WorkCover employees had issued false certificates of competency**, which ICAC states significantly undermined **workplace** safety on building sites. ... [18] In 2002, a **New South Wales** parliamentary committee criticized the **WorkCover** Authority. [19]
Original Query	When was the agency formed
Human Rewrite	when was nsw formed
LLM Rewrite	When was WorkCover NSW formed?
IterCQR Query	When was the NSW WorkCover agency formed to **promote productive, healthy and safe workplaces for workers in New South Wales.**
Gold Answer	The WorkCover Authority of New South Wales was a New South Wales Government agency established in 1989.
Gold Passage	**WorkCover** Authority of **New South Wales** - Wikipedia CentralNotice **WorkCover** Authority of **New South Wales** From Wikipedia, the free encyclopedia Jump to navigation Jump to search Authority of **New South Wales** Statutory authority overview **Formed 1989** Dissolved 2015 Jurisdiction **New South Wales** Parent Statutory authority Department of Finance and Services Key documents Safety, Return to Work and Support Board Act, 2012 (NSW) Work Health and Safety Act, 2011 (NSW) Website **workcover** .nsw .gov .au The **WorkCover** Authority of **New South Wales** or **WorkCover** NSW is a **New South Wales** Government agency established in 1989. The agency creates regulations to **promote productive, healthy and safe workplaces for workers** and employers in **New South Wales**. [1] The agency formed part of the Safety, Return to Work and Support Division established pursuant to the Safety, Return to Work and Support Board Act, 2012 (NSW). On 1 September 2015, **WorkCover** NSW was replaced by three new entities – Insurance and Care NSW (icare),...The information below pertains to the former **WorkCover** NSW. **WorkCover** NSW no longer exists, however its functions have been split between the aforementioned newly created agencies.

Table 12: Reformulated Queries by IterCQR on QReCC dataset. The green and red words stand for overlap with previous context and rewritten entity. Underlined words notate the content that contains the gold answer for the given query.

Type	Content
Previous Turns	Query: What is the meaning of the song alejandro Answer: The song bids farewell to her lovers.
Original Query	whose song is it?
Gold Answer	Lady Gaga
IterCQR Query	Who released the song "Alejandro" by Alejandro, a song that bids farewell to her lovers and bids far
Gold Passage	Alejandro (song) Introduction "Alejandro" is a song by American singer Lady Gaga. It was released as the third single from her third EP, "The Fame Monster" (2009). Co-written and produced by Gaga and Nadir "RedOne" Khayat, it was inspired by her "Fear of Men Monster". The singer bids farewell to her lovers over mid-tempo synth-pop music with a Europop beat. Contemporary critics predominantly gave "Alejandro" positive reviews and noted that it takes influence from the pop acts ABBA and Ace of Base. The song charted in the United Kingdom and Hungary due to digital sales following the album\'s release. Upon release, "Alejandro" charted again in the United Kingdom as well as in Australia, Canada, New Zealand, Sweden, and the United States while topping the Czech, Finnish, Mexican, Venezuelan, Polish, Russian, Bulgarian, and Romanian charts.

Table 13: Failure case of TopiOCQA-trained IterCQR. In this case, the IterCQR query includes the keyword "Alejandro" repeatedly, deviating from the previous dialogue context.