Adaptive Query Rewriting: Aligning Rewriters through Marginal Probability of Conversational Answers

Tianhua Zhang$^{\heartsuit *}$ **, Kun Li**$^{\heartsuit *}$**, Hongyin Luo**$^{\diamondsuit}$**,**
Xixin Wu$^{\heartsuit}$**, James Glass**$^{\diamondsuit}$**, Helen Meng**$^{\heartsuit}$
$^{\heartsuit}$The Chinese University of Hong Kong, Hong Kong SAR, China
$^{\diamondsuit}$Massachusetts Institute of Technology, Cambridge MA, USA
`thzhang@link.cuhk.edu.hk, kunli@se.cuhk.edu.hk`

Abstract

Query rewriting is a crucial technique for passage retrieval in open-domain conversational question answering (CQA). It decontexualizes conversational queries into self-contained questions suitable for *off-the-shelf* retrievers. Existing methods attempt to incorporate retriever's preference during the training of rewriting models. However, these approaches typically rely on extensive annotations such as in-domain rewrites and/or relevant passage labels, limiting the models' generalization and adaptation capabilities. In this paper, we introduce AdaQR (**Ada**ptive **Q**uery **R**ewriting), a framework for training query rewriting models with limited rewrite annotations from seed datasets and completely no passage label. Our approach begins by fine-tuning compact large language models using only ~10% of rewrite annotations from the seed dataset training split. The models are then utilized to self-sample rewrite candidates for each query instance, further eliminating the expense for human labeling or larger language model prompting often adopted in curating preference data. A novel approach is then proposed to assess retriever's preference for these candidates with the probability of answers conditioned on the conversational query by marginalizing the Top-K passages. This serves as the reward for optimizing the rewriter further using Direct Preference Optimization (DPO), a process free of rewrite and retrieval annotations. Experimental results on four open-domain CQA datasets demonstrate that AdaQR not only enhances the in-domain capabilities of the rewriter with limited annotation requirement, but also adapts effectively to out-of-domain datasets.

1 Introduction

Passage retrieval in open-domain conversational question answering (CQA) have gained significant prominence in recent years (Anantha et al., 2021). Unlike standard retrieval with single-turn queries (Kwiatkowski et al., 2019), it poses unique challenges in resolving conversational dependencies like omission, ambiguity, and coreference resolution (Qu et al., 2020; Adlakha et al., 2022). Many existing methods (Yu et al., 2021; Lin et al., 2021b; Li et al., 2022) address these challenges by training specialized retrievers. However, re-training retrievers for conversational search can be costly and may not fully leverage the benefits of *off-the-shelf* single-turn retrievers (Wu et al., 2022).

A prevalent approach for overcoming this challenge involves *query rewriting* (QR) (Elgohary et al., 2019; Vakulenko et al., 2020; Yu et al., 2020). In this method, conversational queries are decontextualized into self-contained, standalone queries, which are then processed by *off-the-shelf* retrievers to find relevant information. Earlier studies (Elgohary et al., 2019; Anantha et al., 2021) focused on fine-tuning language models to reformulate human rewrites. However, Ye et al. (2023) noted that human annotations may only resolve ambiguity while overlooking informative context within conversations. They suggested using language language models (LLMs) for rewrite generation. Recent research (Wu et al., 2022; Mo et al., 2024; Yoon et al., 2024) underscores the significance of incorporating retrieval signals during rewriter training to enhance downstream retrieval performance. Yoon et al. (2024) aligned fine-tuned models with retriever's feedback on the ranking of gold passages using Direct Preference Optimization (DPO).

Nevertheless, these approaches often necessitate substantial amounts of rewrite and/or passage labels for supervision, yet resources are scarce and expensive for collection (Yu et al., 2021). Moreover, they mainly optimize QR systems for in-domain performance, i.e., training labels are from the validated datasets, while the adaptation ability and out-of-domain performance are under-explored. Therefore, this paper centers on: (1) *effectively and*

---

[*] Equal contribution.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 13444–13461
November 12-16, 2024 ©2024 Association for Computational Linguistics

efficiently training QR models with limited annotation requirements and, (2) *examining their capacity for adaptation with preference alignment under weak supervision.*

Following the paradigm for aligning LLMs which takes supervised fine-tuning (SFT) and preference optimization sequentially (Ouyang et al., 2022), optimization towards retrievers' preferences can be applied to the models that have undergone supervised fine-tuning for query rewriting. Aligning with retrievers' preferences can further tune the rewriter to reformulate queries with better recall of relevant passages (Yoon et al., 2024). Importantly, we also find it to be capable of adapting rewriters to out-of-domain CQA scenarios (§4). However, a core issue is how the retrievers' preferences should be modeled. While Yoon et al. (2024) uses the ranking of gold passage as the retrievers' preferences, we aim to explore the extent to which the use of labeled data can be reduced. We argue that the corresponding answers within the conversation can be employed to formulate the retriever's preferences. Moreover, conversation answers are more readily accessible than the gold passages, as they naturally happen when the CQA data is constructed.

We propose a novel preference optimization approach, AdaQR, for query rewriters, aiming to optimize rewriters to cater for retrievers, by utilizing conversation answers to model retrievers' preferences. Specifically, we first let the SFT rewriter to self-sample several rewrites for each query instance, bypassing the need for costly human labeling or large language models prompting. These rewrites are then used as the queries to retrieve a set of passages by a target retriever. Subsequently, we calculate the conditional probability of the answer for each retrieved passage and the conversation, and obtain the marginal probability of the answer by marginalizing the passages set. The marginal probability of the answer serves as the reward quantifying the retrievers' preferences over rewrites. Finally, we pair these rewrites based on their reward for optimizing the SFT rewriter with DPO.

We examine the in-domain performance of AdaQR where the training data for SFT and preference optimization all comes from the validated datasets. Empirical results show that AdaQR greatly improves the quality of rewrites generated by the rewriter, compared with the SFT-only counterpart, leading to comparable or even better performance over existing SOTA QR methods. More importantly, the out-of-domain evaluation, where the preference optimization is applied to a out-of-domain SFT rewriter, also observes the same performance gain, justifying the ability of AdaQR to adapt the rewriter to the target domains.

The key contributions of this work are: (1) We propose AdaQR, a preference optimization approach for enhancing query rewriter. (2) AdaQR models retrievers' preferences by leveraging the answer within conversations, allowing training query rewriters for various conversational question answering tasks even without passage labels. (3) Experiments show AdaQR can not only amplify rewriters' in-domain capability, but also adapt them to out-of-domain conversational question answering tasks.

2 Methodology

2.1 Task Formulation

We focus on the *query rewriting* task for conversational passage retrieval using *off-the-shelf* retrievers. Given the conversation history $H_{<t} = \{q_i, a_i\}_{i=1}^{t-1}$, where t denotes the current turn number, a query rewriting model $\mathcal{M}_\theta$ is trained to transform the the current question q_t into a standalone, self-contained query $\hat{r}_t$. We omit the subscript t in subsequent description for simplicity. The retriever $\mathcal{R}$, which remains unchanged, takes $\hat{r}$ as input to search for relevant passages from the corpus P. In the complete training process of QR, we only need limited rewrite labels $\{r\}$ from two seed datasets, with no passage annotations required.

2.2 Overview

We propose AdaQR to build query rewriters applicable to various conversational question answering scenarios, through preference optimization involving no in-domain rewrite or passage labels. AdaQR uses the probability of answers as the reward quantifying the retriever's preferences over the rewrites, to further optimize an already tuned rewriter on out-of-domain data and adapt it to a target dataset. As shown in Fig. 1, AdaQR, warm-started with a SFT query rewriter fine-tuned with a limited number of (in- or out-of-domain) labeled data beforehand (§2.3), operates in the pipeline: (1) Sample rewrites from the SFT rewriter, which are then used as search queries for retrieving passages; (2) Derive the marginal probability of the answer as reward based on the conversation and retrieved passages (§2.4); (3) Construct preference pairs using the reward and tune the rewriter with Direct Preference

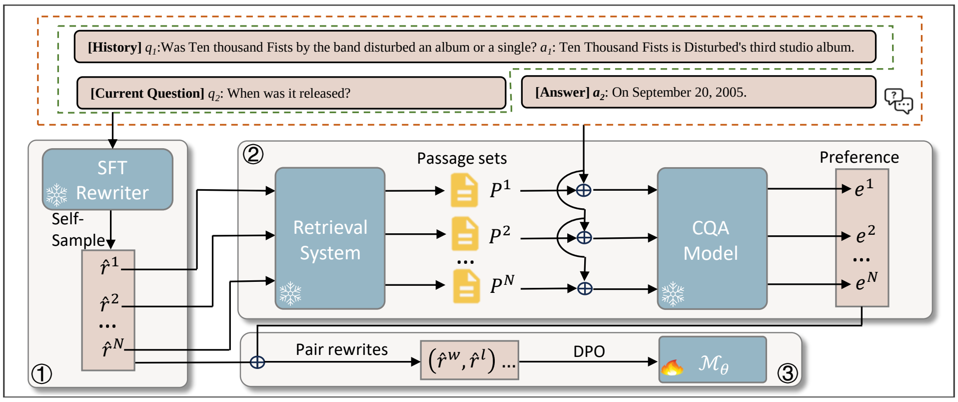

Figure 1: Illustration of AdaQR which applies preference optimization to the rewriter $\mathcal{M}_\theta$.

Optimization (Rafailov et al., 2024) (§2.5).

2.3 Supervised Fine-Tuning

Limited Rewrite Labels To empower models with basic query rewriting capability, we need a limited number of rewrite labels for supervised fine-tuning. We separately curate labels (~10% of the training set) from two seed datasets, QReCC (Anantha et al., 2021) and TopiOCQA (Adlakha et al., 2022) for comprehensive adaptation performance analysis of AdaQR. We derive `QReCC-SFT` with $3,850$ rewrite labels generated by ChatGPT (`gpt-3.5-turbo`) under few-shot learning setting[1] from previous work (Ye et al., 2023). We derive `TopiOCQA-SFT` with $4,278$ training instances, with rewrite labels generated by `gpt-4`. The same instruction and in-context learning examples used in QReCC from Ye et al. (2023) are followed. The complete prompt is detailed in Appendix Table 6. Note that no passage label is required in this stage. The resulting two fine-tuning models $\mathcal{M}_{SFT}$, are subsequently adapted and tested on additional datasets.

Training Objective We use the curated labeled data to fine-tune a language model, equipping it with basic query rewriting ability. Given the conversation history H and the current query q, the LM $\mathcal{M}_\theta$ is trained to predict the rewrite r by minimizing the negative log-likelihood as

$$\mathcal{L}_{SFT} = -\log p_{\mathcal{M}_\theta}(r|H, q) \qquad (1)$$

It should be noted that the models trained in the above way might have insufficient rewriting abilities, especially for out-of-domain conversational query (will be shown in §4). Next, we apply AdaQR to these SFT rewriters to enhance their capabilities for both in- and out-of-domain scenarios.

2.4 Reward Collection

Rewrite Self-sampling To obtain training data for preference optimization, we use the fine-tuned models, $\mathcal{M}_{SFT}$, to sample three rewrite candidates $\{\hat{r}^i\}_{i=1}^3$ for each conversational query q with the temperature $T = 1$. This strategy of self-sampling by the model produces reasonable rewrite candidates for preference optimization, bypassing the need of expensive human labeling or large language model prompting often adopted in collecting preference data (Ouyang et al., 2022; Bai et al., 2022; Yoon et al., 2024).

Reward Calculation We propose a novel reward calculation method that relies solely on conversation turns, eliminating the need for passage labels. Motivated by Lewis et al. (2021), we use weak supervision with the **marginal probability of *answers*** as the preference feedback, treating the retrieved passages as a latent variable. Concretely, for each rewrite candidate $\hat{r}^i$, we retrieve top-K[2] passages $P^i = \{p_k^i\}_{k=1}^K$ with the retriever $\mathcal{R}$. A pre-trained large language model $\mathcal{A}$ then calculates the log probability $\mathcal{S}_k$ of generating the target answer a conditioned on each retrieved passage p_k^i and the original question q concatenated after the conversation history H:

$$\mathcal{S}_k = \log p_{\mathcal{A}}(a|H, q, p_k^i) \qquad (2)$$

We select a pre-trained model due to the conjecture that LLMs have inherent capabilities established during pretraining, while later fine-tuning or alignment may affect the distribution of logits, resulting in alignment tax or capability misalignment (Huang et al., 2023; Lin et al., 2024; Gekhman et al., 2024). To control for the influence of the rewrite in the probability calculation, we use the original conversation as input question rather than the rewritten

---

[1] Ye et al. (2023) provides rewrite labels in both few-shot learning (FSL) setting and advanced editor setting. We use the labels generated in the initial FSL setting.

[2] $K = 5$ except in §5.2 where we evaluate the effect of K.

queries. This ensures the grounding passage p_k^i is solely responsible for the contribution to the score $\mathcal{S}_k$. By marginalizing the top K passages, we calculate the marginal probability of answer e^i as the retrievers' preference for rewrite candidate $\hat{r}^i$:

$$e^i = \Sigma_{k=1}^{K} \mathcal{P}_{\mathcal{R}}(p_k^i | \hat{r}^i) \mathcal{S}_k \tag{3}$$

where $\mathcal{P}_{\mathcal{R}}(p_k^i|\hat{r}^i)$ denotes the distribution over passages obtained by applying a softmax function to their retrieval scores. Intuitively, a more effective rewrite used as a search query improves the chances of recalling potentially relevant passages. A passage with heightened relevance further enhances the likelihood of generating the answer. Consequently, this rewrite leads to a better reward with a higher marginal probability.

2.5 Preference Optimization

Our goal is to align the SFT rewrite model $\mathcal{M}_\theta$ to generate rewrites preferred by the target retriever, quantified by the marginal answer probabilities e as in Eq. 3. To this end, we apply Direct Preference Optimization (Rafailov et al., 2024) with e as reward to tune $\mathcal{M}_\theta$.

Preference Pairs Construction For each conversation example, we construct pairwise preference data $\{(H, q, \hat{r}^w, \hat{r}^l)\}$ by selecting pairs of rewrites $(\hat{r}^w, \hat{r}^l)$ from $\{\hat{r}^i\}_{i=1}^3$ such that $e^w - e^l > \delta$, where $\delta > 0$ is a hyperparameter. Due to the characteristics of e^i, this constraint ensures that the preferred rewrite $\hat{r}^w$ will lead to useful passages more likely than the dispreferred one $\hat{r}^l$. Unlike conventional ones, our preference data is developed without any human annotations, by using the automatic measurement of retriever preferences in §2.4.

Training Objective Using the pairwise preference data, we tune the model $\mathcal{M}_\theta$ with DPO. The training objective is to minimize

$$\begin{aligned} L_{DPO} = -\log \sigma(\beta \log \frac{\mathcal{M}_\theta\,(\hat{r}^w \mid q, H)}{\mathcal{M}_{SFT}\,(\hat{r}^w \mid q, H)} \\ -\beta \log \frac{\mathcal{M}_\theta\,(\hat{r}^l \mid q, H)}{\mathcal{M}_{SFT}\,(\hat{r}^l \mid q, H)}) \end{aligned} \tag{4}$$

where $\mathcal{M}_{SFT}$ is the reference model from which $\mathcal{M}_\theta$ is initialized, σ is the sigmoid function, and β is a hyperparameter. With this objective, the model is optimized to maximize the contrast between preferred and dispreferred rewrites. It thus is encouraged to generate rewrites with higher marginal probabilities of the answers, which are more likely to lead to the useful passages. See the complete algorithm of AdaQR in Appendix Algorithm 1.

3 Experiments

Datasets We evaluate AdaQR for conversational retrieval task on four conversational question answering (CQA) benchmarks: QReCC (Anantha et al., 2021), TopiOCQA (Adlakha et al., 2022), Doc2Dial (Feng et al., 2020) and MultiDoc2Dial (Feng et al., 2021). The answers in QReCC exhibit relatively large word-level overlap to supporting passages while TopiOCQA uses free-form responses as answers (See Analysis 5.1). TopiOCQA and MultiDoc2Dial involve *topic-shift* with turns in a conversation grounded on multiple documents.

Retrieval Systems We investigate the performance of AdaQR using both sparse and dense retrievers. BM25 serves as the sparse retriever for all datasets. For dense retriever, we employ ANCE (Xiong et al., 2020) trained on MS-MARCO (Bajaj et al., 2018) passage retrieval tasks for all datasets except Doc2Dial, to align with previous studies (Jang et al., 2024; Yoon et al., 2024). We use E5-unsupervised (Wang et al., 2024) for Doc2Dial following Liu et al. (2024). Notably, we refrain from additional training of the retrievers for our specific task. More details are listed in Appendix B.1.

Evaluation Metrics We assess retrieval performance using several metrics: Mean Reciprocal Rank (**MRR**) calculates the average rank of gold passages. Normalized Discounted Cumulative@3 (**NDCG**) evaluates the top-3 retrieval results by considering both relevance and rank. **Recall@**k reflects whether the retriever successfully identifies the gold passages within top-k results.

Implementation We fine-tuned two SFT models with two seed datasets: `QReCC-SFT` and `TopiOCQA-SFT`. Each SFT model underwent further training with DPO, using retriever feedback collected in §2.4 across all four datasets respectively. Our backbone model for all configurations is Mistral-7B (Jiang et al., 2023). To ascertain the generalization ability across different LMs, we further assessed the performance of Gemma-7B (Team et al., 2024) and Llama2-7B (Touvron et al., 2023) for all benchmarks with sparse retrieval in Appendix Table 12. We used the pretrained `Mistral-7B-v0.1` as the CQA model for reward calculation. See training details in Appendix B.2.

Baselines (1) **T5QR** (Lin et al., 2020) fine-tunes T5-base (Raffel et al., 2020) to mimic human rewrites. (2) **CONQRR** (Wu et al., 2022) optimizes query rewriters using reinforcement learning, with the ranking of passages having maximum

		QReCC (8209)							TopiOCQA (2514)					
Type	Method	MRR	MAP	NDCG	R@1	R@5	R@10	R@50	MRR	NDCG	R@1	R@5	R@10	R@100
Sparse Retrieval	Original	6.7	6.4	5.7	3.8	8.6	11.3	18.3	2.1	1.8	1.2	2.9	4.0	9.1
	Human Rewrite	40.6	39.1	37.1	25.5	51.5	63.8	89.4	-	-	-	-	-	-
	T5QR*	33.4	-	30.2	-	-	53.8	-	11.3	9.8	-	-	22.1	44.7
	CONQRR*	38.3	-	-	-	-	60.1	-	-	-	-	-	-	-
	ConvGQR*	44.1	-	41.0	-	-	64.4	-	12.4	10.7	-	-	23.8	45.6
	IterCQR*	46.7	-	44.1	-	-	64.4	-	16.5	14.9	-	-	29.3	54.1
	LLM IQR*	49.4	47.9	46.8	36.4	58.9	67.0	83.1	-	-	-	-	-	-
	GPT4 Prompting*	-	-	-	-	-	-	-	18.5	-	-	-	35.1	62.9
	Llama2 Distill*	-	-	-	-	-	-	-	19.0	-	-	-	35.5	64.6
	RETPO*	50.0	-	47.3	-	-	69.5	-	28.3	26.5	-	-	48.3	73.1
	QReCC-SFT	45.9	44.4	43.2	32.4	56.0	64.9	83.7	16.3	14.6	10.2	22.4	29.3	52.1
	+ Gold-Label	51.9	50.3	49.4	38.8	**61.6**	**69.8**	**86.8**	**20.6**	18.5	12.8	**28.7**	**37.2**	**65.1**
	+ Ours	**52.3**	**50.8**	**49.9**	**39.8**	61.3	69.1	85.0	20.5	**18.9**	**13.4**	27.8	34.8	61.3
	TopiOCQA-SFT	40.8	39.3	37.6	26.6	51.6	62.4	83.4	17.7	15.5	9.9	25.7	34.4	62.0
	+ Gold-Label	48.5	47.0	45.9	34.5	59.0	68.6	**87.2**	**20.5**	**18.1**	**12.3**	**29.0**	**38.2**	**68.0**
	+ Ours	**50.6**	**49.0**	**48.0**	**37.0**	**60.7**	**69.6**	86.7	20.3	18.0	**12.3**	28.2	37.1	66.2
Dense Retrieval	Original	7.5	7.2	6.9	4.9	9.6	11.5	15.5	4.1	3.8	2.3	6.1	7.8	13.4
	Human Rewrite	39.5	37.6	36.7	25.3	50.7	60.3	75.1	-	-	-	-	-	-
	T5QR*	34.5	-	31.8	-	-	53.1	-	23.0	22.2	-	-	37.6	54.4
	CONQRR*	41.8	-	-	-	-	65.1	-	-	-	-	-	-	-
	ConvGQR*	42.0	-	39.1	-	-	63.5	-	25.6	24.3	-	-	41.8	58.8
	IterCQR*	42.9	-	40.2	-	-	65.5	-	26.3	25.1	-	-	42.6	62.0
	InstructLLM*	43.5	-	40.5	-	-	66.7	-	25.3	23.7	-	-	45.1	69.0
	RETPO*	44.0	-	41.1	-	-	66.7	-	30.0	28.9	-	-	49.6	68.7
	QReCC-SFT	41.2	39.4	38.5	27.1	52.7	61.9	76.4	25.6	24.2	16.4	37.0	43.8	63.5
	+ Gold-Label	**45.5**	**43.5**	**42.8**	**30.4**	**58.3**	**67.7**	**81.5**	**36.4**	**35.2**	24.3	**51.2**	**59.8**	**79.6**
	+ Ours	45.3	**43.5**	42.7	**30.4**	58.1	67.2	81.4	36.0	34.6	**24.5**	50.5	58.2	78.7
	TopiOCQA-SFT	39.8	37.8	36.9	25.8	50.9	60.4	75.7	33.4	31.9	22.8	46.7	54.7	73.8
	+ Gold-Label	43.2	41.2	40.4	28.0	**55.8**	**65.8**	**80.9**	37.5	36.1	25.8	51.8	60.7	79.8
	+ Ours	**43.4**	**41.5**	**40.8**	**28.3**	**55.8**	65.6	80.4	**38.1**	**36.6**	**26.3**	**53.0**	**61.3**	**79.9**

Table 1: Evaluation results of sparse and dense retrieval on QReCC and TopiOCQA. Two SFT models (`QReCC-SFT` and `TopiOCQA-SFT`) are evaluated to demonstrate the in-domain and out-of-domain performance. We include baselines following Yoon et al. (2024) and Ye et al. (2023), denoted with *. Methods requiring in-domain passage labels are marked with background color in this and subsequent tables. The best scores among our implementation (`SFT`, `Gold-Label` and `Ours`) under each setting are in **bold**. See experimental details in Appendix B.3.

token overlap to answers as weak supervision. (3) **ConvGQR** (Mo et al., 2024) trains query rewriting and expansion models with Mean Squared Error between embeddings of query and relevant passage as an auxiliary loss. (4) **IterCQR** (Jang et al., 2024) iteratively trains the query rewriter with cosine similarity between gold passages and reformulated queries by ChatGPT as IR signal. (5) **LLM IQR** (Ye et al., 2023) introduces "rewrite-then-edit" to prompt ChatGPT first generates rewrites and then edits them according to pre-defined criteria. (6) **RETPO** (Yoon et al., 2024) prompts `gpt-4` to generate multiple rewrites and collect gold passage ranking as retrieval feedback upon all training data of QReCC and TopioCQA. **RETPO** fine-tunes Llama2-7b (Touvron et al., 2023) to replicate the rewrite with the best retrieval preference, termed as (7) **Llama2 Distill**, and then aligned it with retrieval preference using DPO. (8) **GPT4 Prompting** (Yoon et al., 2024) generates rewrites for test questions with `gpt-4`. (9) We implement **Gold-Label** under our setting, with the ranking of the gold passages as the retrieval preference. This serves as an upper-bound for the performance of our weakly supervised AdaQR. (10) We implement two vanilla baselines under our setting, i.e., **Original** and **Human Rewrite**, where the original questions and dataset provided human rewrites[3] are used as retrieval queries.

4 Main Results

Tables 1 and 2 show the main result of AdaQR across 4 benchmarks and 2 retrievers, alongside the comparisons with baseline approaches.

Preference optimization brings improvement over SFT. AdaQR (`Ours`) shows consistent and evident improvement over its `SFT`-only counterpart across all the combinations of datasets and retrievers, indicating the effectiveness of preference optimization in further enhancing rewriters' abilities.

AdaQR improves in- and out-of-domain performance. For in-domain scenarios, on QReCC, our

---

[3] Among the four datasets we evaluated, only QReCC provides human-labeled rewrites.

		Doc2dial (640)					
Type	Method	MRR	NDCG	R@1	R@5	R@10	R@20
Sparse	Original	32.7	31.5	22.7	44.5	52.3	61.3
	GPT4o 0-shot	51.8	51.1	37.8	69.8	77.0	83.6
	GPT4o 1-shot	53.8	53.2	40.2	70.0	77.8	86.7
	QReCC-SFT	56.0	55.7	42.3	72.0	81.7	89.5
	+ Gold-Label	**60.6**	**60.6**	**46.1**	**78.0**	**85.2**	**91.4**
	+ Ours	59.9	59.7	**46.1**	77.3	84.5	90.0
	TopiOCQA-SFT	56.4	55.9	41.9	74.5	82.3	88.8
	+ Gold-Label	**62.1**	**62.1**	47.5	**81.1**	**87.2**	**92.8**
	+ Ours	61.8	61.8	**47.7**	80.5	86.3	91.4
Dense	Original	23.7	21.8	16.1	30.9	40.0	46.1
	GPT4o 0-shot	44.9	43.4	32.5	59.1	68.3	77.2
	GPT4o 1-shot	45.6	43.9	33.0	60.3	68.9	78.9
	QReCC-SFT	47.3	46.8	32.0	64.2	75.8	84.4
	+ Gold-Label	**54.4**	**53.8**	**38.6**	**73.0**	**84.8**	**91.9**
	+ Ours	53.6	52.9	38.4	71.3	81.9	91.1
	TopiOCQA-SFT	46.5	45.1	32.3	62.7	73.8	81.4
	+ Gold-Label	**53.9**	**53.6**	**38.4**	**71.6**	**82.0**	**88.9**
	+ Ours	51.3	50.4	35.6	69.2	79.7	87.8
		MultiDoc2dial (648)					
Sparse	Original	34.6	38.4	25.0	44.6	54.2	63.9
	GPT4o 0-shot	47.8	46.7	34.7	63.0	72.5	80.6
	GPT4o 1-shot	48.7	47.9	35.0	65.6	74.2	83.3
	QReCC-SFT	51.4	50.9	38.6	65.7	75.3	82.7
	+ Gold-Label	**55.7**	55.4	42.0	**71.9**	**81.3**	**88.0**
	+ Ours	55.6	**55.6**	**42.1**	71.5	80.3	87.0
	TopiOCQA-SFT	52.1	50.9	38.6	69.0	77.0	84.7
	+ Gold-Label	55.3	54.1	42.4	70.5	80.3	**87.7**
	+ Ours	**56.6**	**55.9**	**43.5**	**73.0**	**81.3**	87.2
Dense	Original	23.8	21.8	15.4	33.5	39.5	45.2
	GPT4o 0-shot	39.3	37.1	27.6	54.5	64.0	71.9
	GPT4o 1-shot	39.8	37.8	27.5	54.0	65.9	73.6
	QReCC-SFT	41.6	40.7	27.6	57.3	67.9	75.9
	+ Gold-Label	**45.7**	**44.8**	**31.5**	**62.5**	**72.1**	**83.5**
	+ Ours	44.3	42.9	30.3	60.5	71.3	81.3
	TopiOCQA-SFT	39.9	37.8	27.8	53.1	64.0	74.4
	+ Gold-Label	**45.1**	**43.3**	**31.9**	**58.6**	**72.8**	**82.4**
	+ Ours	43.8	42.2	30.4	57.6	70.2	80.9

Table 2: Evaluation results of sparse and dense retrieval on Doc2Dial and MultiDoc2Dial. Both two SFT models (`QReCC-SFT` and `TopiOCQA-SFT`) are out-of-domain evaluation. We include zero-shot and one-shot (an example from QReCC) learning performance of GPT4o as comparison baselines. See prompt in Appendix A and more analysis in Appendix D.

approach, `QReCC-SFT+Ours`, outperforms all baselines; while on TopiOCQA, `TopiOCQA-SFT+Ours` surpasses baselines other than `RETPO` and its ablated variant `Llama2 Distill` under BM25 retriever, in terms of average performance. These baselines all necessitate passage labels, especially `RETPO` and `Llama2 Distill` which involve extensive use of both passage labels and rewrite labels from the combination of above two datasets. For out-of-domain scenarios across four datasets, `Ours` that began with a heterogeneous-seed SFT and then underwent preference optimization on target datasets (e.g., `QReCC SFT+Ours` on TopiOCQA, `TopiOCQA SFT+Ours` on QReCC), still exceeds most of baselines. In most cases, the heterogeneous-seed `SFT` lags behinds the baselines, but gets close to or surpasses them after preference optimization (`+Ours`). Together, our approach can not only amplify rewriters' in-domain capabilities but also successfully adapt them to out-of-domain tasks, even in the absence of passage labels.

Conversation answer is as effective as passage labels for preference optimization. Both our approach and its variant with `Gold-Label`, are implemented in the same paradigm but with different types of reward (marginal probability of answer vs. ranking of gold passages). The two approaches have comparable performances, and even sometimes `Ours` outperforms `Gold-Label`, demonstrating the effective role of the marginal probability-based reward for modeling the retrievers' preference, while such reward is more accessible. These makes AdaQR a quite cost-effective method for adapting rewriters to various CQA tasks.

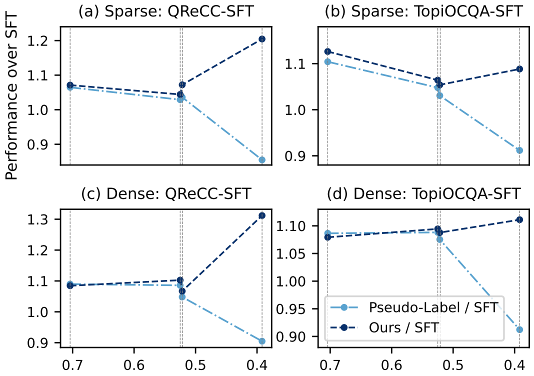

Figure 2: Average performance of `Pseudo-Label` and `Ours` over `SFT` as the F1-score (x-axis) between the answers and gold passages declines. Scores > 1 denote improvement over `SFT`. The four vertical lines correspond to the F1-scores of QReCC (0.704), Doc2Dial (0.525), MultiDoc2Dial (0.522) and TopiOCQA (0.392).

AdaQR has applicability to both sparse and dense retrievers. Positive effect of our approach can be seen for both sparse (BM25) and dense (E5, ANCE) retrievers, which verifies the general applicability of our approach to various retrievers. On the other hand, note that the ANCE retriever has better performance than the BM25 for TopiOCQA, as exemplified by `SFT`'s higher retrieval metrics under ANCE over under BM25. We also observe that for TopiOCQA, the improvement of `Ours` over `SFT` under ANCE is greater than under BM25. Therefore, it is reasonable to speculate that the benefit brought by our approach would get more pronounced with better retrievers.

Similar patterns also manifest in the cases of using Gemma and Llama2 as the base models (see

		QReCC (8209)				TopiOCQA (2514)				Doc2Dial (640)				MultiDoc2dial (648)			
Type	Method	MRR	R@5	R@50	AVG	MRR	R@5	R@100	AVG	MRR	R@5	R@20	AVG	MRR	R@5	R@20	AVG
Sparse	QReCC-SFT	45.9	56.0	83.7	61.8	16.3	22.4	52.1	30.3	56.0	72.0	89.5	72.5	51.4	65.7	82.7	66.6
	+ Pseudo-Label	50.5	60.8	**86.1**	65.8	13.1	17.4	47.2	25.9	58.0	75.8	**90.0**	74.6	52.3	69.6	85.0	69.0
	+ **Ours**	**52.3**	**61.3**	85.0	**66.2**	**20.5**	**27.8**	**61.3**	**36.5**	**59.9**	**77.3**	**90.0**	**75.7**	**55.6**	**71.5**	**87.0**	**71.4**
	TopiOCQA-SFT	40.8	51.6	83.4	58.6	17.7	25.7	62.0	35.1	56.4	74.5	88.8	73.2	52.1	69.0	84.7	68.6
	+ Pseudo-Label	47.8	59.1	**87.2**	64.7	16.8	23.0	56.3	32.0	61.1	78.3	90.6	76.7	54.6	71.6	86.0	70.7
	+ **Ours**	**50.6**	**60.7**	86.7	**66.0**	**20.3**	**28.2**	**66.2**	**38.2**	**61.8**	**80.5**	**91.4**	**77.9**	**56.6**	**73.0**	**87.2**	**72.3**
Dense	QReCC-SFT	41.2	52.7	76.4	56.8	25.6	37.0	63.5	42.0	47.3	64.2	84.4	65.3	41.6	57.3	75.9	58.2
	+ Pseudo-Label	**45.5**	**58.1**	**82.1**	**61.9**	21.9	31.3	61.0	38.0	52.3	70.8	89.5	70.9	43.1	59.1	80.7	61.0
	+ **Ours**	45.3	**58.1**	81.4	61.6	**36.0**	**50.5**	**78.7**	**55.1**	**53.6**	**71.3**	**91.1**	**72.0**	**44.3**	**60.5**	**81.3**	**62.1**
	TopiOCQA-SFT	39.8	50.9	75.7	55.5	33.4	46.7	73.8	51.3	46.5	62.7	81.4	63.5	39.9	53.1	74.4	55.8
	+ Pseudo-Label	43.3	**56.0**	**81.6**	**60.3**	29.0	40.2	71.3	46.8	**51.3**	68.0	**88.0**	69.1	43.6	**57.6**	78.9	60.0
	+ **Ours**	**43.4**	55.8	80.4	59.9	**38.1**	**53.0**	**79.9**	**57.0**	**51.3**	**69.2**	87.8	**69.5**	**43.8**	**57.6**	**80.9**	**60.7**

Table 3: Comparison between two weakly supervised approaches: `Pseudo-Label` and `Ours`. Evaluation results of sparse and dense retrieval with two SFT versions, i.e., `QReCC-SFT` and `TopiOCQA-SFT`, are listed.

Table 12). This points to the general effectiveness of AdaQR with different open-source LMs.

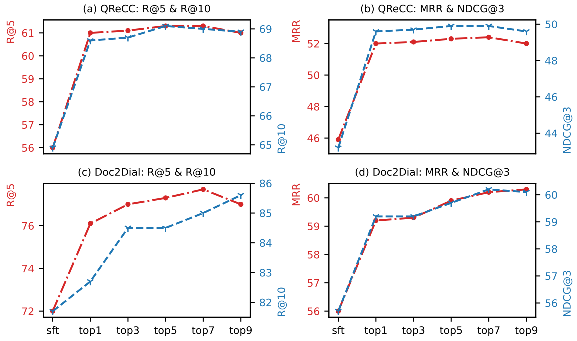

Figure 3: Retrieval performance with varying top-K values ($k = 1, 3, 5, 7, 9$) in reward calculation using `QReCC-SFT`. See detailed results in Appendix Table 13. We further analyze two passage organization types (concatenation and marginalization) in Appendix F.

5 Analysis

5.1 Comparison of Weakly Supervised Approaches

We compare our reward calculation approach (`Ours`) to a word-level based weak supervision method (`Pseudo-Label`) under the same setting, as shown in Table 3. Both approaches eliminate the need for in-domain labels. Our method derives retrieval feedback through assessing the probability of the target answer by marginalizing the top-K passages. On the contrary, `Pseudo-Label` uses the ranking of pseudo-relevant passages that have the maximum F1-score to the answer as the retriever's preferences, following Wu et al. (2022).

AdaQR surpasses `Psuedo-Label` on TopiOCQA, Doc2Dial and MultiDoc2Dial across all settings in terms of average performance. The relatively good performance of `Pseudo-Label` on QReCC is attributed to the dataset's characteristics, where the answers exhibit a high level of overlap with sentences in the supporting passages. Consequently, this straightforward word-level based weak supervision can readily identify relevant passage, with 82% of gold passages detected. However, **`Pseudo-Label` is notably susceptible to the influence of the word-level overlap requirement.** We visualize the performance over `SFT` across four datasets with decreasing F1-scores in Figure 2. The retrieval performance of `Pseudo-Label` drops significantly as the F1-score decreases. Notably, when assessed on TopiOCQA, which features free-form responses as answers, this approach even negatively impacts the results, resulting in performance inferior to the SFT-only version, i.e., `Pseudo-Label`/`SFT` < 1. In contrast, AdaQR measures the retrievers' preferences from semantic-level, **demonstrating greater robustness and stability, providing consistent performance improvement across all settings.**

Moreover, as analyzed in Appendix E and Figure 5, **AdaQR shows notable enhancement in addressing challenging queries with *topic-shift*** over `SFT`, achieving performance ratio comparable to the `Gold-Label` counterpart. However, `Pseudo-Label`'s heavy reliance on word-level overlap hampers its effectiveness.

5.2 Effect of K Values in Reward Calculation

Figure 3 depicts the retrieval performance across varying numbers of top retrieved passages ($K = 1, 3, 5, 7, 9$) for reward calculation in §2.4. Detailed performance metrics are presented in Table 13 of Appendix. Direct preference optimization across all candidate values of K significantly improves the retrieval performance over the `SFT` verison, reflecting the efficacy of our proposed reward calcu-

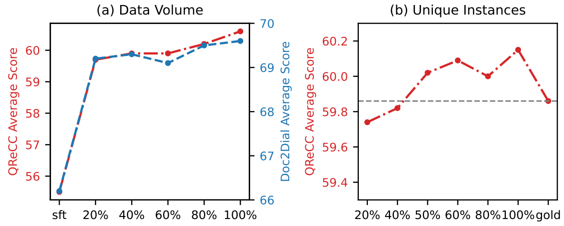

Figure 4: Average retrieval performance with varying number of training data during preference optimization under `QReCC-SFT` setting.

lation methodology. Relying solely on the top-1 retrieved passage may yield sub-optimal results. For instance, only 36.4% of training instances rank the gold passage first with BM25 on QReCC. For Doc2Dial, increasing the value of K tends to enhance the overall performance. **AdaQR demonstrates robustness against potential irrelevant information when more passages are involved by enlarging** K, and effectively reflects retrievers' preferences without requiring in-domain passages or rewrite labels. We opt for $K = 5$ across all settings to strike a balance between effectiveness and efficiency, avoiding manual bias towards the best configuration.

5.3 Effect of Data Volume in Preference Optimization

We randomly sample rewrite pairs from those used in the main experiment at various proportions, $20\%-80\%$ with an interval of 20%, and then use DPO to tune `QReCC-SFT` on these sets of preference pairs individually. The average performance on QReCC (in-domain) and Doc2dial (out-of-domain) is plotted in Fig. 4(a). The performance generally improves with larger data volume. Notably, even with only 20% of pairs, our approach still achieves satisfactory improvement over its SFT version.

AdaQR allows us to incorporate extra instances in QReCC training set that lack gold passage labels for preference optimization, while these examples cannot be used for training of `Gold-Label` approach. To verify the benefit of these unlabeled data enabled by our weak supervision, we collect the pair from each example with the largest reward gap for both `Gold-Label` and `Ours`[4], and gradually reduce the size of data used for `Ours`. In Fig. 4(b), `Ours` with 40% of pairs reaches the similar level of performance as `Gold-Label`, at which point `Ours` uses 22% less number of training pairs than `Gold-Label`. Crucially, when trained on $\geq 50\%$ of pairs, `Ours` consistently exceeds `Gold-Label` and peaks with full data, **highlighting AdaQR's advantage of exploiting unlabeled data**.

Method	MRR	MAP	NDCG	R@1	R@5	R@10	R@50
GPT SFT	45.9	44.4	43.2	32.4	56.0	64.9	83.7
+ Ours	**52.3**	**50.8**	**49.9**	**39.8**	61.3	69.1	85.0
Human SFT	35.5	34.2	32.2	21.8	45.6	56.5	81.0
+ Ours	51.9	50.4	49.4	38.7	**61.5**	**69.8**	**86.5**

Table 4: Comparison of sparse retrieval results on QReCC dataset between AdaQR trained with GPT and human rewrite annotations during SFT stage.

5.4 Effect of AdaQR with Different SFT Annotation

AdaQR only requires a limited number of labeled rewrite annotations from seed datasets for the supervised fine-tuning (SFT) stage. During preference optimization, the marginal probability of answers is proposed as the weak supervision for rewrite candidates derived by self-sampling, a process free of rewrite and retrieval annotations. To analyze AdaQR's effectiveness and robustness under different SFT annotations, we also adopt human rewrites provided by QReCC dataset as SFT supervision and report the results in Table 4. The same $3,850$ training instances are used, differing only in target rewrite source, i.e., human versus GPT. There is a notable performance gap between human rewrites and GPT rewrites after the SFT stage (e.g., 21.8 vs. 32.4 on R@1). As revealed in Ye et al. (2023), this may because human rewrites tend to just resolve ambiguity, whereas LLM-generated rewrites often include more informative context. Nevertheless, following the preference optimization in our approach, the two settings yield comparable retrieval performance. **AdaQR significantly enhances the performance and reduces the gap by incorporating retrieval feedback into the training phase of the rewriter**. This demonstrates the robustness of AdaQR, as it can be effectively applied to diverse annotations in seed training datasets.

5.5 Case Study

For an intuitive understanding of AdaQR's effect on the rewriters, we provide two cases including the conversations and the corresponding rewrites

---

[4] Even with the same δ value, `Gold-Label` and `Ours` would obtain different numbers of preference pairs for a given instance, due to the difference in their reward calculation. Here we only collect the pair with the largest reward gap for each instance, instead of all pairs with reward gap greater than δ in §2.5, avoiding the variation in the number of preference pairs.

by our models both before and after preference alignment. The rewrites generated after AdaQR alignment generally contain more informative context. More importantly, the rewriters after alignment have a better skill of understanding conversation history, which is reflected in their more accurate clarification of the original questions.

[Conversation history]
Q: Who was lise meitner? A: She was an austrian-swedish physicist.
...
Q: Why did she have to flee from germany? A: Because she lost her austrian citizenship.
[Current query]
Why?
[Topiocqa-SFT rewrite]
Why did Lise Meitner have to flee from Germany?
[AdaQR (Topiocqa-SFT+Sparse DPO) rewrite]
Why did Lise Meitner lose her Austrian citizenship and have to flee from Germany?

[Conversation history]
Q: What can you tell me about Retirement Benefits. A: We want you to know what Social Security can mean for you and your family s financial future.
...
Q: What Else Affects I Retirement Benefits? A: You can choose to keep working beyond your full retirement age. If you do, you can increase your future Social Security benefits . Each extra year you work adds another year of earnings to your Social Security record. Do you work beyond your full retirement age?
[Current query]
Yes?
[Qrecc-SFT rewrite]
Do you work beyond your full retirement age?
[AdaQR (Qrecc-SFT+Sparse DPO) rewrite]
If I choose to keep working beyond my full retirement age, can I increase my future Social Security benefits by adding another year of earnings to my Social Security record for each extra year I work?

Table 5: Cases of rewrites by the models before and after preference alignments. The conversations are from TopiOCQA and Doc2dial respectively.

6 Related Works

Conversational Retrieval is a precursor task to open-domain conversational question answering. Many existing approaches (Yu et al., 2021; Li et al., 2022; Lin et al., 2021b) fine-tune specialized dense retrievers. However, to leverage the benefits of *off-the-shelf* single-turn retrievers, conversational query rewriting has been applied to transform each conversational question into a standalone query (Elgohary et al., 2019; Vakulenko et al., 2020; Yu et al., 2020). Previous approaches typically train query rewriting models with human (Elgohary et al., 2019; Anantha et al., 2021) or LLM-based (Ye et al., 2023; Jang et al., 2024) rewrite labels. However, acquiring in-domain labels for training on each specific dataset proves costly and the standard fine-tuning alone fails to incorporate retrieval feedback, potentially resulting in sub-optimal performance (Wu et al., 2022; Yoon et al., 2024). Although recent studies (Wu et al., 2022; Mo et al., 2024; Yoon et al., 2024) suggest integrating signals from retrievers with preference alignment, they require large amounts of in-domain labels. To alleviate this data bottleneck, we propose to train effective query rewriting models and assess the adaptation performance with only limited rewrite labels and completely no passage annotation.

Preference Optimization is a critical research area focused on ensuring that large language models adhere to human values and intents (Bai et al., 2022; Ouyang et al., 2022; Rafailov et al., 2024). Ouyang et al. (2022) fine-tunes LLMs with human feedback to align them with user intent, involving collecting human-annotated demonstrations and rankings of model outputs, followed by supervised learning and reinforcement learning. Kim et al. (2023) uses synthetic feedback instead of human annotations, by reward modeling with synthetic feedback to simulate high-quality demonstrations. Besides improving general abilities of LLMs, some work also focuses on specific aspects. For example, Tian et al. (2023) uses truthfulness measurements as a proxy preference signal to encourage factuality in the model. Yoon et al. (2024) utilizes the gold passage's ranking as the retrievers' preferences to optimize the model for rewriting search queries. Our approach is similar to Yoon et al. (2024) in the target task, query rewriting. However, the preference signals used for aligning our model are synthesized without any retrieval-related labeled data.

7 Conclusion

We introduce AdaQR for enhancing query rewriters with minimal to zero in-domain labels, through the preference optimization towards retrievers' preferences. A novel feature of AdaQR is in measuring retrievers' preferences with marginal probabilities of answers based on conversations and retrieved passages, enabling improvement or adaptation of query rewriters without labeled data. Experiments show that AdaQR brings significant improvement to query rewriters' in-domain performance and adapts them well to out-of-domain conversational question answering tasks. AdaQR shows promise in establishing effective query rewriters for arbitrary conversational question answering tasks with minimal effort.

Acknowledgement

This research was supported by the Center for Perceptual and Interactive Intelligence (CPII) Ltd under the Innovation and Technology Commission's

InnoHK Scheme.

Limitations

Although AdaQR demonstrates notable generalization and adaptation capabilities with limited rewrite label requirements, there are still some limitations. First, our evaluation in this study was conducted on four datasets. We used QReCC and TopiOCQA as two separate seed datasets, and adapted SFT rewriters to the other three datasets as out-of-domain evaluations. This may not encompass all possible scenarios. Secondly, although we explore the effect of AdaQR with different SFT annotations (`Human` and `GPT`) in § 5.4, we did not conduct an in-depth analysis of how the quantity of annotations in the supervised fine-tuning stage might affect the overall performance, due to computational constraints. Our objective is to use a small number of labels to achieve good retrieval performance. Further investigation into the impact of label quality and quantity remains an avenue for potential enhancement of rewriting performance and to reduce demands on annotation. Lastly, we propose AdaQR to derive retrievers' preference using the conditional probability of answers. While we briefly analyzed the impact of different types of passage organization (concatenation and marginalization) in Appendix F, our primary focus in this paper lies in the thorough analysis of a marginalization approach due to its robustness and effectiveness. Nevertheless, we acknowledge the potential benefit of delving deeper into concatenation-based methods, which might offer valuable insights for the research community in tackling the query rewriting task.

Ethics Statement

Query rewriting is instrumental in clarifying users' search intents during information-seeking conversations, improving the retrieval of relevant passages. Our work can greatly enhance the performance of query rewriters. Nevertheless, it is important to recognize that our approach cannot always guarantee perfect rewrites and may retrieve irrelevant or even nonfactual information. This is partly due to the inherent shortcomings of large language models, which serve as the foundation models in our approach. These models have a propensity to generate hallucinations. Other potential reasons include imperfect retrievers and limited search scopes. Low-quality retrieval result may confuse or even mislead users. Therefore, to ensure the reliability of retrieval results in practical scenarios, it is crucial to implement effective filtering mechanisms, such as rerankers, to identify and exclude passages containing nonfactual information.

References

Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2022. Topiocqa: Open-domain conversational question answering with topic switching. *Preprint*, arXiv: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.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova Dassarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, John Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Christopher Olah, Benjamin Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *ArXiv*, abs/2204.05862.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset. *Preprint*, arXiv:1611.09268.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *ArXiv*, abs/2305.14314.

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

Song Feng, Siva Sankalp Patel, Hui Wan, and Sachindra Joshi. 2021. MultiDoc2Dial: Modeling dialogues grounded in multiple documents. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6162–6176, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Song Feng, Hui Wan, Chulaka Gunasekara, Siva Patel, Sachindra Joshi, and Luis Lastras. 2020. doc2dial: A goal-oriented document-grounded dialogue dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8118–8128, Online. Association for Computational Linguistics.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. Does fine-tuning llms on new knowledge encourage hallucinations? *Preprint*, arXiv:2405.05904.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685.

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

Yunah Jang, Kang il Lee, Hyunkyung Bae, Hwanhee Lee, and Kyomin Jung. 2024. Itercqr: Iterative conversational query reformulation with retrieval guidance. *Preprint*, arXiv:2311.09820.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *Preprint*, arXiv:1702.08734.

Sungdong Kim, Sanghwan Bae, Jamin Shin, Soyoung Kang, Donghyun Kwak, Kang Yoo, and Minjoon Seo. 2023. Aligning large language models through synthetic feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13677–13700, Singapore. Association for Computational Linguistics.

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

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Kun Li, Tianhua Zhang, Liping Tang, Junan Li, Hongyuan Lu, Xixin Wu, and Helen Meng. 2022. Grounded dialogue generation with cross-encoding re-ranker, grounding span prediction, and passage dropout. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 123–129, Dublin, Ireland. Association for Computational Linguistics.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021a. Pyserini: An easy-to-use python toolkit to support replicable ir research with sparse and dense representations. *Preprint*, arXiv:2102.10073.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021b. Contextualized query embeddings for conversational search. *Preprint*, arXiv:2104.08707.

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.

Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, Hanze Dong, Renjie Pi, Han Zhao, Nan Jiang, Heng Ji, Yuan Yao, and Tong Zhang. 2024. Mitigating the alignment tax of rlhf. *Preprint*, arXiv:2309.06256.

Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Chatqa: Surpassing gpt-4 on conversational qa and rag. *Preprint*, arXiv:2401.10225.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Fengran Mo, Kelong Mao, Yutao Zhu, Yihong Wu, Kaiyu Huang, and Jian-Yun Nie. 2024. Convgqr: Generative query reformulation for conversational search. *Preprint*, arXiv:2305.15645.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. 2022. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155.

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. 2020. Open-retrieval conversational question answering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20. ACM.

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

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. *Journal of Machine Learning Research*, 21(140):1–67.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2020. Question rewriting for conversational question answering. *Preprint*, arXiv:2004.14652.

Christophe Van Gysel and Maarten de Rijke. 2018. Pytrec_eval: An extremely fast python interface to trec_eval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 873–876, New York, NY, USA. Association for Computing Machinery.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

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. *Preprint*, arXiv:2112.08558.

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

Fanghua Ye, Meng Fang, Shenghui Li, and Emine Yilmaz. 2023. Enhancing conversational search: Large language model-aided informative query rewriting. In *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. 2024. Ask optimal questions: Aligning large language models with retriever's preference in conversational search. *Preprint*, arXiv:2402.11827.

Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. Few-shot generative conversational query rewriting. *Preprint*, arXiv:2006.05009.

Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-shot conversational dense retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21. ACM.

A Prompts

Supervised Fine-Tuning Label Collection Table 6 presents the prompt used for rewrite generation on TopiOCQA dataset (§2.3). 4278 training instances are derived with Azure OpenAI `gpt-4 (0314)`. The in-context learning examples are from the QReCC dataset. We do not use in-domain demonstrations for TopiOCQA rewrite label generation since our approach can train effective rewrite models with a limited number of rewrite instances without requiring optimal labels. For QReCC, we use 3850 rewrite labels provided by Ye et al. (2023), who offering annotations in both few-shot learning (FSL) setting and advanced editor setting. In the editor setting, ChatGPT refines the rewrites from FSL, functioning as a rewrite editor to provide more competitive results. We use the labels generated in the initial FSL setting.

GPT4o Prompting Baselines Tables 7 and 8 list the prompt for GPT4o prompting used as Doc2Dial and MultiDoc2Dial baselines in Table 2.

B Experimental Details

B.1 Retrieval

For sparse retrieval BM25, we use Pyserini (Lin et al., 2021a) for efficient search and set $k_1 = 0.82$ and $b = 0.68$ in QReCC, and $k_1 = 0.9$ and $b = 0.4$ in TopiOCQA, Doc2Dial and MultiDoc2Dial respectively. k_1 controls the non-linear term frequency normalization and b is the scale of the inverse document frequency. For dense retrieval, we use Faiss (Johnson et al., 2017) with Exact Search for Inner Product (IndexFlatIP). We employ ANCE (Xiong et al., 2020) across all dataset except Doc2Dial, with checkpoint trained on MS-MARCO (Bajaj et al., 2018) passage retrieval tasks[5], aligning with previous studies (Jang et al., 2024; Yoon et al., 2024). Our evaluation on dense retrieval of Doc2Dial employs E5-unsupervised

Given a question and its context, decontextualize the question by addressing coreference and omission issues. The resulting question should retain its original meaning and be as informative as possible, and should not duplicate any previously asked questions in the context.

Context: [Q: When was Born to Fly released? A: Sara Evans's third studio album, Born to Fly, was released on October 10, 2000.]
Question: Was Born to Fly well received by critics?
Rewrite: Was Born to Fly well received by critics?

Context: [Q: When was Keith Carradine born? A: Keith Ian Carradine was born August 8, 1949. Q: Is he married? A: Keith Carradine married Sandra Will on February 6, 1982.]
Question: Do they have any children?
Rewrite: Do Keith Carradine and Sandra Will have any children?

Context: [Q: Who proposed that atoms are the basic units of matter? A: John Dalton proposed that each chemical element is composed of atoms of a single, unique type, and they can combine to form more complex structures called chemical compounds.]
Question: How did the proposal come about?
Rewrite: How did John Dalton's proposal that each chemical element is composed of atoms of a single unique type, and they can combine to form more complex structures called chemical compounds come about?

Context: [Q: What is it called when two liquids separate? A: Decantation is a process for the separation of mixtures of immiscible liquids or of a liquid and a solid mixture such as a suspension. Q: How does the separation occur? A: The layer closer to the top of the container-the less dense of the two liquids, or the liquid from which the precipitate or sediment has settled out-is poured off.]
Question: Then what happens?
Rewrite: Then what happens after the layer closer to the top of the container is poured off with decantation?

Context: {`current_context`}
Question: {`current_question`}
Rewrite:

Table 6: Prompt for rewrite generation. Ye et al. (2023) used this prompt to generate rewrite labels for QReCC under few-shot learning setting with ChatGPT (`gpt-3.5-turbo`). We follow the same prompt to derive rewrite labels for TopiOCQA.

Given a question and its context, decontextualize the question by addressing coreference and omission issues. The resulting question should retain its original meaning and be as informative as possible, and should not duplicate any previously asked questions in the context.

Context: {`current_context`}
Question: {`current_question`}
Rewrite:

Table 7: Prompt for GPT4o-0shot prompting on Doc2Dial and MultiDoc2Dial benchmarks. We use the same instruction as the rewrite generation in Table 6.

---

[5] `https://huggingface.co/sentence-transformers/msmarco-roberta-base-ance-firstp`

Given a question and its context, decontextualize the question by addressing coreference and omission issues. The resulting question should retain its original meaning and be as informative as possible, and should not duplicate any previously asked questions in the context.

Context: Q: When was Keith Carradine born? A: Keith Ian Carradine was born August 8, 1949.
Q: Is he married? A: Keith Carradine married Sandra Will on February 6, 1982.
Question: Do they have any children?
Rewrite: Do Keith Carradine and Sandra Will have any children?

Context: `{current_context}`
Question: `{current_question}`
Rewrite:

Table 8: Prompt for GPT4o-1shot prompting on Doc2Dial and MultiDoc2Dial benchmarks. We use the same instruction as the rewrite generation in Table 6. The out-of-domain demonstration example is also used in Table 6 by Ye et al. (2023).

(Wang et al., 2024) following Liu et al. (2024)[6]. We set the maximum sequence length to 512 for both ANCE and E5-unsupervised We employ the `pytrec_eval` toolkit (Van Gysel and de Rijke, 2018) for retrieval metric values computation.

B.2 Training

We use AdamW (Loshchilov and Hutter, 2017) optimizer with learning rates of 1e-4 and 1e-5 for SFT, DPO stages respectively. The learning rates of both stages undergo a warmup of 10% of overall training steps, followed by a linear decrease until 0. We set the hyperparameter $\delta = 0.1$ for organizing preference pairs and β=0.1 during DPO stage.

We resort to quantized LoRA (QLoRA) (Hu et al., 2021; Dettmers et al., 2023) as the parameter-efficient fine-tuning technique for training our models with an NVIDIA A6000 GPU. Specifically, QLoRA is applied to query and value attention matrices inside each decoder block with a fixed rank of 8, a scaling factor of 16, and a dropout probability of 0.05. The model weights are loaded in 4-bit NormalFloat Quantization.

B.3 Evaluation

We train query rewriting models with instances that are not first-turn queries, as these are typically self-contained in application. To ensure fair comparisons with previous baselines during evaluation, we incorporate all first-turn query test instances for both QReCC and TopiOCQA benchmarks. For TopiOCQA, we use the original questions as search inputs. Following previous works (Wu et al., 2022; Ye et al., 2023; Anantha et al., 2021), we replace all first user queries in QReCC conversations with their corresponding human rewrites as retrieval queries. This step is necessary due to the ambiguity of some questions in this dataset, necessitating additional topical information. Consequently, the performance of first-turn instances remains consistent across experiments within our setup, i.e., `SFT`, `Gold-Label`, `Pseudo-Label` and `Ours`. We present results for benchmarks Doc2Dial and MultiDoc2Dial on non-first-turn test instances.

C Data Statistics

We list the data statistics of four benchmarks in Table 11. As described in Appendix B.3, we train query rewriting models with instances that are not first-turn queries. During the preference optimization, conversational answers are needed to calculate the marginal probability as reward. The number of evaluation instances for QReCC, TopiOCQA, Doc2Dial and MultiDoc2Dial are 8209, 2514, 640, and 648 respectively.

D More Results on Doc2Dial and MultiDoc2Dial

We use Doc2Dial and MultiDoc2Dial primarily for a comprehensive assessment on the domain adaptation capabilities of AdaQR, a task that previous QR works have not explored. To demonstrate AdaQR's out-of-domain performance, we implemented zero-shot and one-shot GPT-4o as two baselines, considering the prominent performance of LLMs. In Table 9, we provide additional results on Doc2Dial and MultiDoc2Dial, contrasting AdaQR with ChatQA (Liu et al., 2024). Since ChatQA fine-tunes the Dragon retriever instead of employing query rewriting, we compare ChatQA's performance with AdaQR + Dragon retriever **without** any additional training. Notably, our approach uses rewrites generated by AdaQR aligned with other dense retrievers as reported in Table 2, rather than with Dragon. Aligning AdaQR with Dragon as the target retriever could potentially yield further performance improvements.

---

[6] `https://huggingface.co/intfloat/e5-base-unsupervised`

Method	**R@1**	**R@5**	**R@10**	**R@50**	**MRR**	**NDCG@3**
			Doc2Dial			
Dragon + Fine-tuned (ChatQA)	37.2	73.4	83.9	97.2	53.0	51.9
Dragon + AdaQA rewrite (Ours)	47.2	82.8	91.6	99.1	63.1	63.3
			MultiDoc2Dial			
Dragon + Fine-tuned (ChatQA)	25.5	54.3	65.7	88.3	39.3	37.9
Dragon + AdaQA rewrite (Ours)	37.7	72.2	84.3	96.3	53.6	53.1

Table 9: Comparison between ChatQA and AdaQR on Doc2Dial and MultiDoc2Dial datasets.

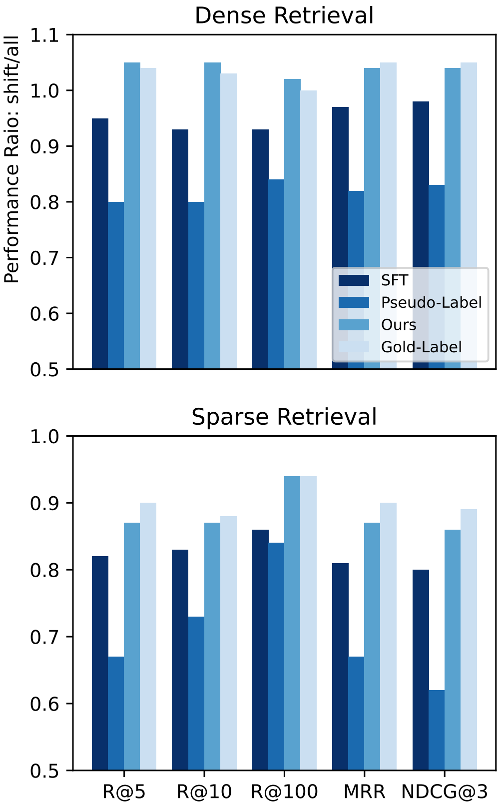

Figure 5: Performance ratio of turns with topic-shift and all instances on the test set of TopiOCQA with dense and sparse retrievers.

E Analysis of Topic Shift

Figure 5 visualizes the results of topic-shift instances in TopiOCQA, measured by the performance ratio of topic-shift turns over the overall performance. A test example is considered *topic-shift* if the gold passage of the current question differs from the latest history turn. The statistics for initial turns, topic-shift turns and topic-concentrated turns are 205, 672 and 1637 respectively. **AdaQR demonstrates significant improvement in handling challenging topic-shift instances compared to `SFT` and `Pseudo-Label`.** `Pseudo-Label`'s heavy reliance on word-level overlap between answers and passages not only decreases the overall performance, but also impairs its capability to handle examples involving topic changes. We achieve a performance ratio comparable to the `Gold-Label` counterpart and even surpasses it on certain metrics with a dense retriever.

F Analysis of Passage Concatenation and Marginalization

			QReCC (8209)				
	MRR	MAP	NDCG	R@1	R@5	R@50	avg
C5	51.9	50.3	49.5	38.9	**61.4**	**85.7**	56.3
M5	**52.3**	**50.8**	**49.9**	**39.8**	61.3	85.0	**56.5**
↑ (%)	0.8	0.9	0.9	2.1	-0.1	-0.8	0.4
C9	51.0	49.4	48.5	37.8	60.8	**86.1**	55.6
M9	**52.0**	**50.5**	**49.6**	**39.5**	**61.0**	84.7	**56.2**
↑ (%)	2.1	2.2	2.3	4.5	0.3	-1.6	1.1
			Doc2Dial (640)				
	MRR	NDCG	R@1	R@5	R@10	R@20	avg
C5	59.1	58.6	45.5	75.9	83.1	**91.4**	68.9
M5	**59.9**	**59.7**	**46.1**	**77.3**	**84.5**	90.0	**69.6**
↑ (%)	1.3	1.8	1.4	1.8	1.7	-1.5	0.9
C9	59.7	59.2	46.1	76.1	82.8	**91.7**	69.3
M9	**60.3**	**60.1**	**46.7**	**77.0**	**85.6**	90.9	**70.1**
↑ (%)	1.1	1.6	1.4	1.2	3.4	-0.9	1.2

Table 10: Comparison between two passage organization types in reward calculation with `QReCC-SFT` setting: `Concatenated (C)` vs. `Marginalized (M)`. We consider two K values, top-5 passages (`C5` and `M5`) and top-9 passages (`C9` and `M9`) since `C1` and `M1` are identical. ↑ (%) denotes the percentage improvement of `M` over `C`, relative to `C`.

To fully assess the effect of using the conditional probability of answers as retrieval preference, we introduce and compare two types of passage organization designs: (1) `M5` follows the description in §2.4 that computes the probability by marginalizing top-K passages. (2) `C5` is proposed as a variation to `M5`, which computes the probability of answers conditioned on concatenated top-K passages as a single input. The retrieval performance with top-5 and top-9 passages are reported in Table 10. In general, `M5` surpasses `C5` on the average performance and metrics with relatively small $@k$ values. On QReCC, we observed that the improvement percentage increases when the inclusion of more passages during the reward calculation, i.e., 9 vs. 5. This suggest that `M5` may exhibit greater resilience to noise in information. Nevertheless, `C5` demonstrates its own advantage by achieving relatively strong performance with fewer input tokens during reward calculation.

Algorithm 1 AdaQR Algorithm

```
Input: LLM $\mathcal{M}_\theta$, Pre-trained LLM $\mathcal{A}$, retriever $\mathcal{R}$, seed dataset $D_S = \{H, q, r\}$, target dataset $D_T = \{H, q, a\}$ with corpus $P$, threshold $\delta$
Output: Aligned query rewriting LLM $\mathcal{M}_\theta$
// Supervised Fine-Tuning
initialize $\mathcal{M}_{SFT} = \mathcal{M}_\theta$ and fine-tune $\mathcal{M}_{SFT}$ on $D_S$ using $\mathcal{L}_{SFT} = -\log p_{\mathcal{M}_\theta}(r|H, q)$
for $(H_{<t}, q_t, a_t) \in D_T$ do
    $\hat{r}^1, \hat{r}^2, \hat{r}^3 \sim \mathcal{M}_{\mathcal{SFT}}(\cdot|H, q)$ ▷ sample 3 rewrite candidates from $\mathcal{M}_{\mathcal{SFT}}$
    // Reward Collection
    for $i \in \{1, 2, 3\}$ do
        $e^i, \mathcal{X} \leftarrow 0, \emptyset$
        $P^i = \{p_1^i, ..., p_k^i\}_{k=1}^K \sim \mathcal{R}(\hat{r}^i, P)$ ▷ retrieve top-$K$ passages for each rewrite with retriever $\mathcal{R}$
        for $k \in \{1, ..., K\}$ do
            $\mathcal{S}_k = \log p_{\mathcal{A}}(a|H, q, p_k^i)$ ▷ compute the log probability of answer conditioned on the retrieved passage and conversational query
            $e^i \mathrel{+}= \mathcal{P}_{\mathcal{R}}(p_k^i|\hat{r}^i)\mathcal{S}_k$
        end for
    end for
    // Preference Pairs Construction
    for $w \in \{1, 2, 3\}$ do
        for $l \in \{1, 2, 3\}$ and $l > w$ do
            if $e^w - e^l > \delta$ then
                $\mathcal{X} \leftarrow \mathcal{X} \cup (H, q, r^w, r^l)$
            end if
        end for
    end for
end for
// Preference Optimization
initialize $\mathcal{M}_\theta = \mathcal{M}_{SFT}$ and fine-tune $\mathcal{M}_\theta$ on $\mathcal{X}$ using
$\mathcal{L}_{DPO} = -\log \sigma(\beta \log \frac{\mathcal{M}_\theta(r_w|q_t, H_{<t})}{\mathcal{M}_{SFT}(r_w|q_t, H_{<t})} - \beta \log \frac{\mathcal{M}_\theta(r_l|q_t, H_{<t})}{\mathcal{M}_{SFT}(r_l|q_t, H_{<t})})$
return $\mathcal{M}_\theta$
```

Dataset	Train (all/w-ans/w-ans+non-turn1)	Evaluation (all/w-psg/w-psg+non-turn1)	Corpus Size
QReCC	51928/47463/39677	16451/8209/6852	~50M
TopiOCQA	45450/41798/38862	2514/2514/2309	~50M
Doc2Dial	21998/21998/10011	640/640/640	1557
MultiDoc2Dial	24603/24603/11101	648/648/648	1559

Table 11: Data statistics of QReCC, TopiOCQA, Doc2Dial and MultiDoc2Dial. For training split, we report the total number of instances (`all`), the number of instances with answers (`w-ans`), and the number of instances with answers that are not first turns (`w-ans+non-turn1`). For evaluation, we report the total number of instances (`all`), the number of instances with passage labels (`w-psg`), and the number of instances with passage labels that are not first turns (`w-psg+non-turn1`) since reference labels are needed for metric scores calculation.

		Gemma-7B						Llama-7B					
	Method	**MRR**	**MAP**	**NDCG**	**R@1**	**R@5**	**R@50**	**MRR**	**MAP**	**NDCG**	**R@1**	**R@5**	**R@50**
QReCC	QReCC-SFT	45.7	44.2	42.9	32.3	55.5	83.6	45.4	43.9	42.4	32.1	55.3	83.4
	+ Pseudo-Label	50.7	49.1	48.0	37.5	60.3	86.3	50.9	49.3	48.3	37.8	60.7	85.2
	+ Gold-Label	51.2	49.7	48.7	37.8	61.3	**86.7**	51.3	49.7	48.7	38.2	**60.9**	**85.7**
	+ Ours	**52.5**	**50.8**	**50.1**	**39.4**	**61.7**	84.8	**52.0**	**50.4**	**49.6**	**39.5**	**60.9**	83.8
	Method	**MRR**	**NDCG**	**R@1**	**R@5**	**R@50**	**R@100**	**MRR**	**NDCG**	**R@1**	**R@5**	**R@50**	**R@100**
TopiOCQA	QReCC-SFT	16.4	14.7	10.5	22.6	45.4	52.7	15.3	13.6	9.2	21.3	45.0	51.4
	+ Pseudo-Label	13.4	11.8	8.6	18.0	38.4	45.7	12.87	11.4	8.4	16.9	37.4	45.1
	+ Gold-Label	**20.5**	18.3	12.5	**28.6**	**57.8**	**65.8**	**19.4**	**17.2**	11.9	**27.0**	**56.2**	**64.5**
	+ Ours	20.4	**18.5**	**13.1**	28.2	54.8	62.7	18.86	**17.2**	**12.3**	25.5	50.8	58.8
	Method	**MRR**	**NDCG**	**R@1**	**R@5**	**R@10**	**R@20**	**MRR**	**NDCG**	**R@1**	**R@5**	**R@10**	**R@20**
Doc2Dial	QReCC-SFT	57.9	57.2	44.7	75.0	82.5	90.8	56.2	55.8	42.3	74.1	80.9	88.3
	+ Pseudo-Label	58.8	58.5	44.7	76.3	84.2	91.6	58.9	59.1	44.5	75.8	84.1	90.5
	+ Gold-Label	60.3	60.2	46.1	78.3	85.6	93.0	**60.3**	**60.2**	**46.6**	**75.9**	**85.3**	91.6
	+ Ours	**60.9**	**60.8**	**46.9**	**78.8**	**86.1**	**93.1**	59.3	59.5	45.6	74.7	84.5	**91.7**
	Method	**MRR**	**NDCG**	**R@1**	**R@5**	**R@10**	**R@20**	**MRR**	**NDCG**	**R@1**	**R@5**	**R@10**	**R@20**
MultiDoc2Dial	QReCC-SFT	52.0	50.4	40.1	65.4	74.7	84.3	51.2	50.8	37.8	67.0	75.5	83.2
	+ Pseudo-Label	53.3	52.0	41.1	67.4	76.1	86.1	54.7	54.3	41.7	69.9	78.1	86.0
	+ Gold-Label	54.7	53.7	42.0	68.4	79.0	**88.1**	55.2	55.0	41.5	71.0	78.4	87.2
	+ Ours	**55.1**	**53.9**	**42.1**	**70.2**	**80.1**	**88.1**	**56.5**	**55.8**	**43.4**	**73.3**	**80.4**	**87.8**

Table 12: Evaluation results of sparse retrieval (BM25) on QReCC, TopiOCQA, Doc2Dial and MultiDoc2Dial with Gemma-7b and Llama-7b. Due to computational constraints, we experiment AdaQR with QReCC as seed dataset only.

	QReCC (8209)							**Doc2Dial (640)**					
	R@1	**R@5**	**R@10**	**R@50**	**MRR**	**MAP**	**NDCG**	**R@1**	**R@5**	**R@10**	**R@20**	**MRR**	**NDCG**
QReCC SFT	32.4	56.0	64.9	83.7	45.9	44.4	43.2	42.3	72.0	81.7	89.5	56.0	55.7
+ Top-1	39.4	61.0	68.6	84.4	52.0	50.4	49.6	45.3	76.1	82.7	90.0	59.2	59.2
+ Top-3	39.6	61.1	68.7	84.7	52.1	50.6	49.7	45.0	77.0	84.5	**91.1**	59.3	59.2
+ Top-5	**39.8**	**61.3**	**69.1**	**85.0**	52.3	**50.8**	**49.9**	46.1	77.3	84.5	90.0	59.9	59.7
+ Top-7	39.7	**61.3**	69.0	84.9	**52.4**	50.7	**49.9**	46.4	**77.7**	85.0	90.2	60.2	**60.2**
+ Top-9	39.5	61.0	68.9	84.7	52.0	50.5	49.6	**46.7**	77.0	**85.6**	90.9	**60.3**	60.1

Table 13: Evaluation results of sparse retrieval (BM25) on QReCC and Doc2Dial with varying top-K values ($K = 1, 3, 5, 7, 9$) during reward calculation with `QReCC-SFT` setting.

	QReCC (8209)									**Doc2Dial (640)**						
	MRR	MAP	NDCG	R@1	R@5	R@10	R20	R@50	AVG	MRR	NDCG	R@1	R@5	R@10	R@20	AVG
QReCC SFT	45.9	44.4	43.2	32.4	56.0	64.9	73.6	83.7	55.5	56.0	55.7	42.3	72.0	81.7	89.5	66.2
+ 20% Data	51.2	49.6	48.7	38.5	60.5	68.8	76.2	84.0	59.7	59.4	59.2	45.6	75.9	83.3	**91.6**	69.2
+ 40% Data	51.7	50.1	49.2	39.1	60.8	68.4	76.0	84.1	59.9	59.6	59.4	45.9	76.7	83.4	90.9	69.3
+ 60% Data	51.7	50.1	49.2	39.1	60.6	68.4	76.0	84.2	59.9	59.4	59.4	45.6	76.3	83.6	90.6	69.1
+ 80% Data	52.0	50.4	49.5	39.6	60.8	68.5	76.2	84.5	60.2	59.5	**59.7**	45.3	**77.3**	**84.5**	90.6	69.5
+ 100% Data	**52.3**	**50.8**	**49.9**	**39.8**	**61.3**	**69.1**	**76.7**	**85.0**	**60.6**	**59.9**	**59.7**	**46.1**	**77.3**	**84.5**	90.0	**69.6**

Table 14: Evaluation results of sparse retrieval (BM25) on QReCC and Doc2Dial with varying number of training data (20%, 40%, 60%, 80% and 100%) during preference alignment with `QReCC-SFT` setting.