Search-Oriented Conversational Query Editing

Kelong Mao[1,2], **Zhicheng Dou**[1,2]*, **Bang Liu**[3], **Hongjin Qian**[1], **Fengran Mo**[3],
Xiangli Wu[4], **Xiaohua Cheng**[4], **Zhao Cao**[4]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]Engineering Research Center of Next-Generation Search and Recommendation, MOE
[3]Université de Montréal, Québec, Canada
[4]Huawei Poisson Lab
{mkl,dou}@ruc.edu.cn

Abstract

Conversational query rewriting (CQR) realizes conversational search by reformulating the search dialogue into a standalone rewrite. However, existing CQR models either are not learned toward improving the downstream search performance or inefficiently generate the rewrite token-by-token from scratch while neglecting the fact that the search dialogue often has a large overlap with the rewrite. In this paper, we propose EdiRCS, a new text editing-based CQR model tailored for conversational search. In EdiRCS, most of the rewrite tokens are selected from the dialogue in a non-autoregressive fashion and only a few new tokens are generated to supplement the final rewrite, which makes EdiRCS highly efficient. In particular, the learning of EdiRCS is augmented with two search-oriented objectives, including contrastive ranking augmentation and contextualization knowledge transfer, which effectively improve it to select and generate more useful tokens from the view of retrieval. We show that EdiRCS outperforms state-of-the-art CQR models on three conversational search benchmarks while having low rewriting latency, and is more robust to out-of-domain search dialogues and long dialogue context.

1 Introduction

With the rise of intelligent assistants (e.g., Siri and Alexa), conversational search is becoming a new search paradigm of the future (Gao et al., 2022). As users can interact with the search engine in the form of natural dialogue, one of the main challenges of conversational search is to accurately understand users' search intents within the dialogue context (Yu et al., 2020; Wu et al., 2022).

Inspired by the success of dense retrieval in ad-hoc search (Karpukhin et al., 2020), recent studies (Lin et al., 2021a; Mao et al., 2022b; Kim and Kim, 2022) show that using a similar contrastive learning approach to train a conversational dense retriever with a dual encoder architecture can effectively resolve such a complex context understanding problem. However, since the ad-hoc search systems have been built, deployed, and optimized for a long time in the industry, replacing the well-established ad-hoc retriever with a totally new-trained conversational dense retriever would be too expensive and even not realistic in the current early days of conversational search.

As such, another type of method, i.e., *Conversational Query Rewriting* (Vakulenko et al., 2021b; Tredici et al., 2021), which explicitly reformulates the whole search dialogue into a context-independent query rewrite and thus can be seamlessly incorporated into any existing ad-hoc search pipelines to realize conversational search, shows greater practical value. Currently, a typical CQR model is built by fine-tuning an autoregressive pre-trained language model (PLM) using search dialogue (input) and manual rewrite (target) text pairs. Despite the promising results obtained, we argue that it has the following two significant limitations, as shown in Figure 1.

First, the single training objective to simply fit the manual rewrite is not aligned with our ultimate goal, i.e., achieving better performance on the downstream conversational search task. On one hand, the quality of manual rewrites is probably not the best from the view of retrieval, since humans are only instructed to rewrite queries to be self-contained outside the dialogue context, but have no knowledge of the downstream retrieval process. On the other hand, no ranking signals from the retrieval side are taken into account when training the model (Wu et al., 2022). Second, for conversational query rewriting, many expected rewrite tokens can be found in the search dialogue in most cases. However, the autoregressive rewriting model still generates the rewrite completely from scratch,

[*] Corresponding author.

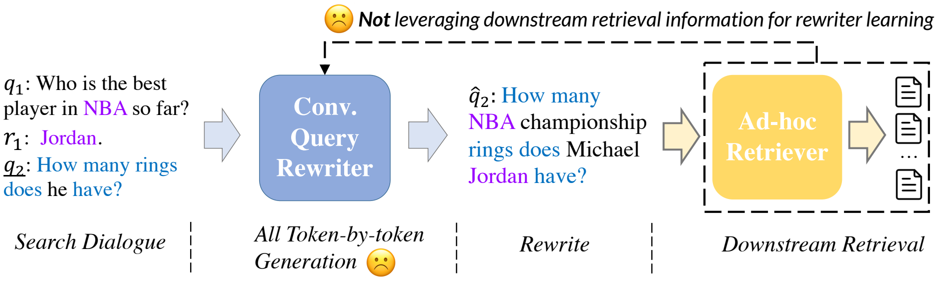
Not leveraging downstream retrieval information for rewriter learning
q_1: Who is the best player in NBA so far?
r_1: Jordan.
q_2: How many rings does he have?
Conv. Query Rewriter
$\hat{q}_2$: How many NBA championship rings does Michael Jordan have?
Ad-hoc Retriever
Search Dialogue
All Token-by-token Generation
Rewrite
Downstream Retrieval

Figure 1: Illustration of the two major limitations of existing autoregressive CQR models. First, the learning of the rewriter does not consider the downstream retrieval process, which would affect the final conversational search performance. Second, most of the rewrite tokens can often be found in the current query (q_2) and dialogue context (q_1 and r_1) while they are still generated from scratch, which is inefficient.

which introduces an over-large search space for token generation and can be unnecessary.

To overcome these limitations, we propose a text Editing-based (Malmi et al., 2022) conversational query Rewriting model tailored for Conversational Search, called EDIRCS. Instead of autoregressively generating the rewrite from scratch, in EDIRCS, most of the rewrite tokens are selected from the search dialogue in a non-autoregressive fashion and only a few new informative tokens are generated to supplement the final rewrite, which makes EDIRCS highly efficient. More importantly, EDIRCS is augmented with **two conversational search-oriented learning objectives.** Specifically, we add a contrastive ranking loss calculated between the dialogue embedding and passage embeddings to improve the model learning toward downstream retrieval performance. Considering that the manual rewrites are not ideal from the view of retrieval, we leverage a specific SPLADE-based (Lassance and Clinchant, 2022) conversational dense retriever, which is fully trained toward conversational search that shows superior context understanding ability, to identify the key tokens that have significant contributions to the retrieval performance, and transfer the knowledge about these key tokens to enhance our rewriting model for both existing token selection and new token generation.

We conduct extensive experiments on three public conversational search datasets and results show that EDIRCS outperforms state-of-the-art conversational query rewriting models when evaluating with both BM25 and a dense retriever ANCE (Xiong et al., 2021) while having low query rewriting latency, and is more robust to out-of-domain search dialogues and long dialogue context.

2 Related Work

Conversational Search. Currently, there are mainly two types of methods to solve the difficult context understanding problem to achieve conversational search, including conversational dense retrieval and conversational query rewriting. Specifically, conversational dense retrieval methods (Yu et al., 2021; Lin et al., 2021a; Mao et al., 2022a,b; Kim and Kim, 2022) encode both the whole search dialogue and passages into embeddings to perform dense retrieval in an end-to-end way, which generally achieve stronger performance but are unfriendly for real deployment. In contrast, conversational query rewriting converts the conversational search problem into an ad-hoc search problem by reformulating the search dialogue into a standalone query rewrite, which is the focus of this work. Existing conversational query rewriting methods include only selecting relevant tokens from the dialogue context (Voskarides et al., 2020; Lin et al., 2021b) and using (dialogue, manual rewrite) text pairs to fine-tune PLMs to be the rewriter (Yu et al., 2020; Lin et al., 2020; Vakulenko et al., 2021a). A significant drawback of these methods is that they focus solely on fitting the manual rewrites but are not trained toward search performance. To tackle it, recent studies (Wu et al., 2022; Chen et al., 2022) have investigated leveraging reinforcement learning for model optimization with retrieval-related rewards. Unlike the existing work, we internalize retrieval information into the learning of our rewriting model through two search-oriented objectives to help it achieve better performance toward conversational search.

Text Editing. Text editing (Malmi et al., 2022) is an effective technique to solve text generation tasks (Reid and Zhong, 2021; Mallinson et al., 2022), where the source and target texts have a large amount of overlap, by predicting efficient edit operations applied to the source sequence, thus often showing lower inference latency and better control over the outputs. In particular, it has been successfully applied in utterance rewriting for dialogue systems (Huang et al., 2021; Hao et al., 2021; Jin et al., 2022), which is very similar to conversational query rewriting. Their major distinction is that the former's utterances usually do not have specific search intents while the latter's user utterances are queries and the latter focuses on improving the downstream search performance. Different from previous work for utterance rewriting, our EdiRCS is particularly improved toward conversational search with simple editing operations and search-oriented learning objectives.

3 Preliminaries

3.1 Task Definition

In this work, we focus on the task of the first-stage passage retrieval of conversational search. Given a search dialogue $s_k = (q_k, r_{k-1}, q_{k-1}, r_{k-2}, ..., q_1)$ (we call it a *session*), our target is to retrieve the relevant passage p for this session, where q_i and r_i denote the query and the system response of the i-th turn, respectively, q_k is the current query, and other turns are the dialogue context. For simplicity, we omit the subscript k in the rest of the paper if not specified.

3.2 Existing Two Types of Methods

Conversational Query Rewriting (CQR) transforms the session s into a de-contextualized query rewrite $\hat{q}$. Then we can feed $\hat{q}$ into any off-the-shelf ad-hoc retriever to realize conversational search.

In contrast, *Conversational Dense Retrieval (CDR)* uses dual encoders f_{S} and f_{P} to map the session and passages to latent vectors, and perform dense retrieval to achieve conversational search. The training usually adopts the ranking loss based on contrastive learning with N negative samples:

$$\mathcal{L}_{\text{rank}} = -\log \frac{e^{f_{\text{S}}(s)\cdot f_{\text{P}}(p^+)}}{e^{f_{\text{S}}(s)\cdot f_{\text{P}}(p^+)} + \sum_{i=1}^{N} e^{f_{\text{S}}(s)\cdot f_{\text{P}}(p_i^-)}}, \quad (1)$$

where p^+ and p^- are the relevant and irrelevant passages for the current turn. It is worth noting that, as the passage information has no change in conversational search compared with that in ad-hoc search, it is common to start with a well-trained ad-hoc dense retriever and only fine-tune the session encoder while freezing the passage encoder (Yu et al., 2021; Lin et al., 2021a).

4 Our Model: EdiRCS

We present EdiRCS, a CQR model that aims to achieve more effective and efficient conversational search based on text editing and augmentation from search-oriented objectives. On the whole, as shown in Figure 2, EdiRCS follows a *selecting-then-generating* text editing architecture, which first selects a few necessary tokens from the session and then generates more useful new tokens to supplement the rewrite. In particular, both the session token selection and new token generation processes benefit from the proposed search-oriented learning to help the generated rewrite achieve better conversational search performance. We finally summarize the whole training and inference processes.

4.1 Conversational Query Editing

In this section, we elaborate on our efficient conversational query editing architecture, including session token selection and new token generation.

Session Token Selection. Since most expected rewrite tokens can be found in the session (Dalton et al., 2020; Voskarides et al., 2020; Anantha et al., 2021), we employ a non-autoregressive selector to directly select those necessary tokens from the session instead of autoregressively generating them from scratch. Specifically, given the input session s, we concatenate all of its tokens and feed them into a 12-layer transformer encoder to obtain the contextualized token embeddings $(\mathbf{h}_1, ..., \mathbf{h}_n)$, where n is the number of tokens of the session. Then, we feed the embeddings into a classification layer to predict the probability $\hat{y}_i^{\text{rt}}$ of retaining each token:

$$r_i = \mathbf{W}\mathbf{h}_i + b, \quad (2)$$

$$\hat{y}_i^{\text{rt}} = \text{sigmoid}(r_i), \quad (3)$$

where $\mathbf{W} \in \mathbb{R}^{1\times d}$ and b are trainable parameters of the classification layer, d is the embedding size, and r_i is the retaining logit. Basically, the selector is trained using the binary cross-entropy loss:

$$\mathcal{L}_{\text{rt}} = -\sum_{i=1}^{n} y_i^{\text{rt}} \log \hat{y}_i^{\text{rt}} + (1 - y_i^{\text{rt}}) \log(1 - \hat{y}_i^{\text{rt}}), \quad (4)$$

where y^{rt} is the binary gold label annotated based on the alignment between the session and the

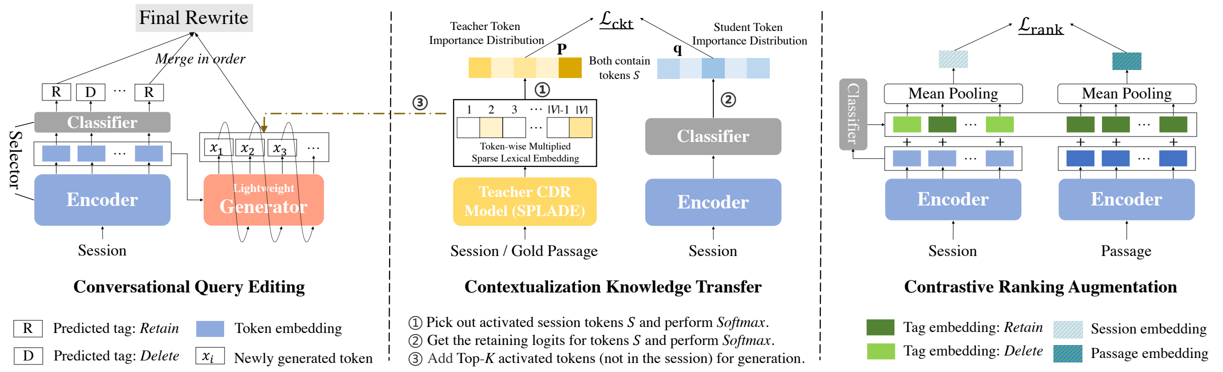

Figure 2: Overview of EDIRCS. It consists of a selector, which selects existing tokens from the input session, and a lightweight generator, which generates new informative tokens. The final rewrite is obtained by merging these two sets of tokens in the original order. Both the training of selector and generator of EDIRCS are enhanced by our proposed search-oriented learning to retain and generate tokens that are important for conversational search.

manual rewrite. For the detailed label annotation process, we refer the readers to Appendix A.

New Token Generation. Although the session often contains most of the rewrite tokens, there are still some important tokens that are not in the session and can only be generated from scratch. Mao et al. (2021) also showed that combining some informative generated text snippets with the original query can improve retrieval performance. Therefore, we incorporate a generator, which will attend to all the token embeddings to generate additional useful tokens for supplementing the rewrite. Considering that the generation difficulty is alleviated in our task since most of the rewrite tokens have been selected before and only a few new tokens are needed to be generated, we employ a lightweight four-layer transformer decoder as the generator and we empirically find that using four layers can already perform well (see § 5.3). The generator is also trained with the standard cross-entropy loss:

$$\mathcal{L}_{\text{gen}} = -\frac{1}{|\mathbf{y}^{\text{gen}}|} \sum_{j}^{|\mathbf{y}^{\text{gen}}|} \log P(y_j^{\text{gen}} | \mathbf{y}_{<j}^{\text{gen}}, \{\mathbf{h}_i\}_{i=1}^{n}), \quad (5)$$

where j is the index of token generation and $\mathbf{y}^{\text{gen}}$ denotes gold labels for all the generated tokens. Label annotation is also introduced in Appendix A.

4.2 Search-Oriented Learning

Considering that simply fitting the manual rewrites is not aligned with our real goal (i.e., passage retrieval), we exploit two search-oriented objectives, including contrastive ranking augmentation and contextualization knowledge transfer, to enhance the training of our text editing-based rewriting toward better retrieval performance.

Contrastive Ranking Augmentation. Borrowing from the conversational dense retrieval, we incorporate a similar contrastive ranking loss (i.e., Eq. 1) upon the selector to reduce the distance between the session and its relevant passage and increase the distances between the session and the irrelevant passages. Specifically, we first update the token embeddings of the session with their predicted tags (*retain* if $y_i^{\text{rt}} > 0.5$, otherwise *delete*):

$$\mathbf{h}'_i = \mathbf{h}_i + TE(\mathbb{I}(y_i^{\text{rt}} > 0.5)), \quad (6)$$

where TE is a trainable tag embedding layer containing two tag embeddings. Then, the session embedding is obtained by performing the mean pooling over all the token embeddings. Similarly, to obtain the passage embedding, we feed the passage into the selector to get its token embeddings, uniformly update them with *retain* tags, and perform the mean pooling. Finally, we use the session and passage embeddings to calculate the ranking loss (Eq. 1) for model optimization. Intuitively, using these diverse ranking signals (i.e., session-passage pairs) to implicitly enhance the tokens embeddings can not only help the selector find tokens in the context that are important for search but also encourage the subsequent generator to generate more useful new tokens.

Contextualization Knowledge Transfer. As existing studies (Lin et al., 2021a; Kim and Kim, 2022) show that CDR models, which are trained directly toward retrieval performance, generally show much better performance, it would be desirable to transfer their strong context understanding abilities into our rewriting model. To this end, in addition to implicitly enhancing the token embeddings with ranking signals, we also help our rewriting model retain and generate tokens that are helpful to retrieval in an explicit knowledge transfer manner.

Specifically, we leverage a high-performing teacher CDR model to explicitly identify those key tokens that have large contributions to the retrieval performance and train our rewriting model to learn this contextualization knowledge. This teacher CDR model is obtained by fine-tuning a lexical ad-hoc retriever SPLADE (Lassance and Clinchant, 2022) using the common ranking loss (Eq. 1). SPLADE can encode a l-length text sequence into a sparse lexical embedding $\mathbf{v} \in \mathbb{R}^{|V|}$, where $|V|$ is the vocabulary size, by predicting token importance in the whole vocabulary space based on the latent token embeddings $(\mathbf{z}_1, ..., \mathbf{z}_l)$ generated by its underlying contextualized encoder:

$$\mathbf{w_i} = \mathbf{EQz}_i + \mathbf{b}, i \in [1, l], \tag{7}$$

$$\mathbf{v}_i = \log(1 + \text{ReLU}(\mathbf{w}_i)), i \in [1, l], \tag{8}$$

$$\mathbf{v} = \text{MaxPool}(\mathbf{v}_1, ..., \mathbf{v}_l), \tag{9}$$

where $\mathbf{Q} \in \mathbb{R}^{d \times d}$ and $\mathbf{b} \in \mathbb{R}^{|V|}$ are trainable parameters, d is the embedding size, and $\mathbf{E} \in \mathbb{R}^{|V| \times d}$ is the input embedding matrix. Note that the output lexical embedding $\mathbf{v}$ is trained to be sparse, i.e., only a few important tokens can be activated with non-zero weights. We feed the session and its gold relevant passage into the SPLADE-based teacher CDR model to get their lexical embeddings $\mathbf{v}^{\text{s}}$ and $\mathbf{v}^{\text{p}}$. The retrieval score is computed as $\sum_{i=1}^{|V|} \mathbf{v}^{\text{s}}[i] \times \mathbf{v}^{\text{p}}[i]$, where $\mathbf{v}[i]$ is the predicted weight of the i-th token. Therefore, the product term $c_i = \mathbf{v}^{\text{s}}[i] \times \mathbf{v}^{\text{p}}[i]$ can represent the contribution of the i-th token to the retrieval. We leverage this knowledge of token retrieval contributions to enhance both our selector and generator toward better search effectiveness.

For the selector, we first pick out a subset S from the vocabulary that contains the session tokens which have contributions to retrieve the gold passage (i.e., $c > 0$). We can obtain the teacher token importance distribution $\mathbf{p} = \text{softmax}([c_1, ..., c_m])$ for these tokens based on their retrieval contributions, where m is the number of tokens in S. Then, we obtain the student token importance distribution $\mathbf{q} = \text{softmax}([r_1, ..., r_m])$ based on the retaining logits (Eq. 2) predicted by the selector[1]. To enhance our selector to be aware that these m important session tokens should be properly selected from the retrieval perspective, we incorporate the following transfer loss:

$$\mathcal{L}_{\text{ckt}} = \frac{1}{m} \sum_{i=1}^{m} \underbrace{\mathbf{p}[i] \log \frac{\mathbf{p}[i]}{\mathbf{q}[i]}}_{\text{term}_1} - \underbrace{\log \sigma(r_i)}_{\text{term}_2} \tag{10}$$

where the first term is to minimize the KL divergence between the teacher and student importance distributions and the second term is to encourage the token i to be selected. σ is *sigmoid* function.

For the generator, we pick out the tokens that have the Top-K largest contributions to the retrieval while not in the session and simply label these tokens as needing to be generated.

4.3 Training and Inference

EDIRCS is trained in a multi-task learning manner:

$$\mathcal{L} = \mathcal{L}_{\text{rt}} + \mathcal{L}_{\text{gen}} + \lambda \mathcal{L}_{\text{rank}} + \beta \mathcal{L}_{\text{ckt}}, \tag{11}$$

where λ and β are hyper-parameters to balance two search-oriented losses. When inference, we only retain the session tokens selected by the selector in their original order and append the new tokens generated by the generator to form the final rewrite.

5 Experiments

5.1 Experimental Settings

Datasets and Evaluation Metrics. We conduct experiments on three widely used conversational search datasets, including QReCC (Anantha et al., 2021) and CAsT-19,20 (Dalton et al., 2020, 2021). QReCC is a large-scale dataset of 14K open-domain search dialogues containing 80K turns with provided training-test split. We randomly select a development set containing 1000 dialogue turns from the training set for parameter tuning. The two CAsT datasets only have no more than 50 search dialogues. Therefore, we use the training set of QReCC to train models and perform in-domain evaluations on the test set of QReCC and perform out-of-domain evaluations on the two CAsT datasets. More dataset details are in Appendix B.

We report the official metrics MRR, Recall@10, and Recall@100 for QReCC, and NDCG@3 and Recall@100 for CAsT datasets.

Compared Systems. We compare our model with three groups of conversational query

[1] Note that a token may appear multiple times in the session and we sum the logits of all positions for the token in this case.

Method	#Params	QR Latency	BM25			Dense Retrieval		
			MRR	R@10	R@100	MRR	R@10	R@100
GPT-2+WS	355M	601ms	0.305	0.489	0.836	0.319	0.510	0.703
T5QR	223M	593ms	0.334	0.538	0.861	0.345	0.531	0.728
QuReTeC	340M	34ms	0.345	0.556	0.863	0.353	0.547	0.733
CQE-sparse	110M	16ms	0.318	0.529	0.834	0.320	0.513	0.709
CONQRR	223M	-	$0.383^{\ddagger}$	$0.601^{\ddagger}$	$0.889^{\ddagger}$	$0.418^{\ddagger}$	$0.651^{\ddagger}$	$0.847^{\ddagger}$
EdiRCS	157M	59ms	**0.412**†	**0.627**†	**0.902**†	**0.421**†	**0.656**†	**0.853**†
For Reference								
Conv-ANCE	125M	N/A	N/A	N/A	N/A	0.471	0.715	0.872
Conv-SPLADE	67M	N/A	N/A	N/A	N/A	0.512	0.709	0.888
Human	N/A	N/A	0.397	0.626	0.985	0.384	0.586	0.781

Table 1: In-domain results on the QReCC test set. The two conversational dense retrieval models are not applicable to use BM25. The QR latency is the average time cost of rewriting per session, which is measured on one RTX 3080 GPU with batch size 1. ‡ denotes the results are replicated from their original paper. † denotes significant improvements of EdiRCS over the other QR baselines expect CONQRR using paired t-test with $p < 0.05$.

rewriting methods. The first group performs text (rewrite) generation from scratch based on fine-tuned PLMs, including **T5QR** (Lin et al., 2020) and **GPT2+WS** (Yu et al., 2020). The second group reformulates the current query by selecting important tokens only from the context, including **QuReTeC** (Voskarides et al., 2020) and **CQE-sparse** (Lin et al., 2021a). The third group is directly trained towards retrieval performance using reinforcement learning, including **CONQRR** (Wu et al., 2022). Besides, we report the performances of **Human** (i.e., using the manual rewrites) and two conversational dense retrievers, **Conv-SPLADE** (i.e., the teacher model) and **Conv-ANCE**, which are fine-tuned from SPLADE (Lassance and Clinchant, 2022) and ANCE (Xiong et al., 2021) with the standard ranking loss (Eq. 1), respectively, for reference. After getting the rewrites, we feed them into an ad-hoc retriever (either BM25 or ANCE) to evaluate the effectiveness of rewriting models for conversational search.

Implementation Details. Experiments are conducted on four NVIDIA GeForce RTX 3080 GPUs. We use the whole encoder and the first four layers of the decoder of *t5-base* to initialize EdiRCS. We train EdiRCS for 5K iterations using the Adam optimizer with a 1e-5 learning rate and 64 batch size. The parameters λ, β, and K are tuned on the development set and finally set to 0.5, 0.2, and 4, respectively. For the ranking loss, we adopt the widely-used in-batch negative sampling plus one hard negative sample randomly selected from Top-50 retrieved passages by BM25. The maximum generated sequence length of EdiRCS is set to 10. For the BM25 retriever, we set its $k_1 = 0.82$ and $b = 0.68$. For the ANCE retriever, we use its checkpoint pre-trained on the MSMARCO dataset at the 600th step. We perform dense retrieval using Faiss (Johnson et al., 2021) with brute force. Code is to be released at `https://github.com/kyriemao/EdiRCS`.

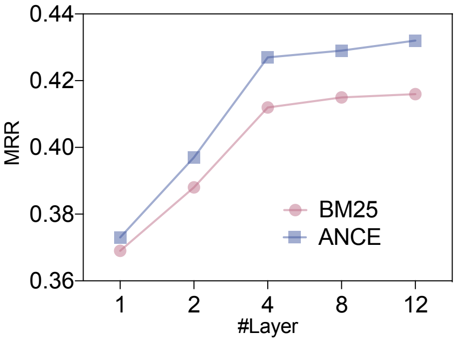

Figure 3: Impact of the number of generator layers evaluated on QReCC.

5.2 Main Results and Analysis

In-Domain Evaluation Results. Evaluation results on the test set of QReCC are shown in Table 1, where we have the following observations:

(1) EdiRCS outperforms all the other QR baselines when evaluated with both BM25 and ANCE. Compared with the second-best model (i.e., CONQRR), the average relative gain of EdiRCS over all the three metrics is +4.5% with BM25 and +1.0% with ANCE, demonstrating the effectiveness of EdiRCS for conversational search. We find that the improvement with ANCE are less than that

with BM25. This may be due to the less coherence of the rewrite generated by EdiRCS than that of CONQRR, since the former is the concatenation of the selected session tokens and new tokens while the latter is autoregressively generated from the T5-based language model. The reduction of coherence may affect the semantic understanding of the ANCE dense retriever to the rewrites, thus affecting its retrieval performance.

Search	Method	CAsT-19		CAsT-20	
		NDCG@3	R@100	NDCG@3	R@100
BM25	T5QR	0.258	0.373	0.141	0.225
	QuReTeC	0.334	0.390	0.173	0.236
	CQE-sparse	0.236	0.358	0.133	0.200
	EdiRCS	**0.345**†	**0.402**†	**0.183**†	**0.244**†
	Human	0.309	0.448	0.240	0.395
ANCE	T5QR	0.417	0.332	0.299	0.353
	QuReTeC	0.430	0.337	0.287	0.346
	CQE-sparse	0.399	0.310	0.271	0.336
	EdiRCS	**0.440**†	**0.353**†	**0.308**†	**0.375**†
	Human	0.461	0.381	0.422	0.465

Table 2: Out-of-domain evaluation results on the two CAsT datasets. † denotes significant improvements of EdiRCS over the other QR baselines using paired t-test with $p < 0.05$.

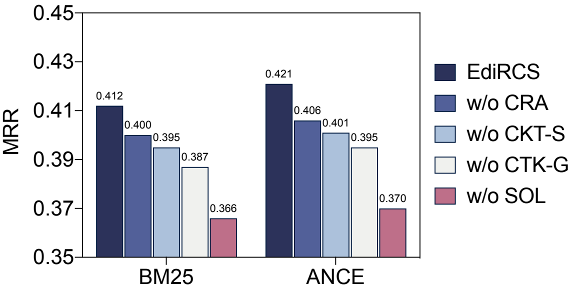

Figure 4: Results of ablation studies on QReCC.

(2) EdiRCS and CONQRR, which learn from the downstream retrieval information, can even significantly outperform using manual rewrites (i.e., Human) when evaluating with ANCE. This demonstrates that manual rewrites are not the best from the view of retrieval and the downstream retrieval information is very valuable for improving query rewriting toward conversational search. Compared with CONQRR which is optimized toward ranking metrics through reinforcement learning, our EdiRCS can not only benefit from ranking signals but also enjoy guidance from a very high-performing CDR teacher (i.e., Conv-SPLADE) to achieve better search effectiveness.

(3) Compared with the other two text editing-based QR models (i.e., QuReTeC and CQE-sparse) which can only select tokens from the dialogue context, EdiRCS supports generating new informative tokens and is augmented with search-oriented learning, thus leading to substantial improvements.

Out-Of-Domain Evaluation Results. We perform out-of-domain evaluations on the two CAsT datasets based on the models trained on QReCC. Results are reported in Table 2. We find that EdiRCS still outperforms the other compared QR models in the zero-shot evaluation with at least +3.5% and +4.4% average relative gains on NDCG@3 and R@100, respectively. This demonstrates the better robustness of EdiRCS to out-of-domain search dialogues. But we also notice that the performance of all QR models still lags behind that of using manual rewrites in most cases, indicating that there still have considerable room for improving the zero-shot capabilities of QR models.

5.3 Efficiency Comparisons

Table 1 also shows the number of parameters and the query rewriting latency. We find that QuReTeC, CQE-sparse, and our EdiRCS have more than 10x speedup than the purely autoregressive QR models (i.e., GPT2+WS, T5QR, and CONQRR[2]). Compared with QuReTeC and CQE-sparse which only have non-autoregressive operations, EdiRCS just has one more lightweight autoregressive generator and is trained to generate only a few tokens ($\leq$ 10), so it is only a little bit slower than these two models and is still quite efficient. Moreover, we find that EdiRCS achieves the best retrieval performance with the second-fewest number of parameters, which verifies the superiority of our proposed learning method.

Besides, we present the impact of the number of generator layers on the retrieval performance in Figure 3. It shows that the marginal benefit of increasing the layer numbers decreases seriously, so we just use four layers to benefit the efficiency.

5.4 Ablation Studies

In this section, we investigate the effects of our proposed search-oriented learning. Specifically, we test four variants of EdiRCS, including: (1) **w/o CRA**: EdiRCS without contrastive ranking augmentation. (2) **w/o CKT-S**: EdiRCS without using contextualization knowledge transfer to enhance the selector (i.e., $\mathcal{L}_{\text{ckt}}$). (3) **w/o CKT-G**: EdiRCS without learning to generate the key tokens that are

[2] The QR latency of CONQRR would be similar to T5QR since they are all based on *t5-base*.

Search Dialogue	Q: What is a **normal blood sugar** level? R: Normal blood sugar levels are less than 100 mg/dL after ... And they are less than 140 mg/dL two hours ... *Q: What does it mean if it's higher than this?*	Q: Where are The Roots from? ... Q: Who started it R: **Tariq Black Thought Trotter and Ahmir Questlove Thompson** started The Roots. *Q: Where did these two meet?*
Gold Passage	**Hyperglycemia** is ... diabetes when the blood glucose level is too high because the body isn't ... hormone **insulin**. Eating too many ... cause your blood sugar to rise.	Tariq Black Thought Trotter and Ahmir ... were both **attending** the **Philadelphia** High **School** for the Creative and Performing Arts...
T5QR	What does it mean if blood sugar higher than this?	Where did Tariq Black and Ahmir Questlove Thompson two meet?
QuReTeC	What does it mean if it's higher than this? normal blood sugar level	Where did these two meet? Tariq Trotter Ahmir Thompson started Roots
EdiRCS	What does mean higher than normal blood sugar level? Hyperglycemia not make insulin.	Where these two meet? Tariq Black Thought and Ahmir Thompson attend Philadelphia School
Human	What does it mean if blood sugar level is higher than normal?	Where did Tariq Black Thought Trotter and Ahmir Questlove Thompson meet?

Table 3: Examples of rewrites generated by Human, T5QR, QuReTeC, and EDIRCS. The current queries are shown in italics. Some important tokens for retrieval in the context and in the gold passages are in bold.

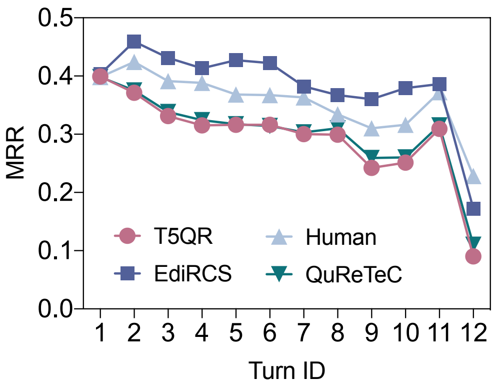

Figure 5: Turn-level performance comparisons using ANCE as the retriever.

not in the original session. (4) **w/o SOL**: EDIRCS without the proposed search-oriented learning. Results are shown in Figure 4.

We find that removing any of the search-oriented objectives results in performance degradation. By contrast, the contextualization knowledge transfer shows to be a little bit more effective than the contrastive ranking augmentation. But the model enhanced with any of the two search-oriented objectives can substantially outperform using nothing (i.e., w/o SOL), suggesting that the proposed search-oriented learning facilitates EDIRCS to achieve better conversational search.

5.5 Multi-turn Analysis

To investigate the long context understanding ability of EDIRCS, we show the fine-grained turn-level model performance in Figure 5. As the dialogue goes on, the context becomes longer and the context understanding problem generally becomes more difficult. We observe that EDIRCS maintains the performance superiority across different turns. Overall, the performance of EDIRCS fluctuates less in deep turns (e.g, from turn No.7 to turn No.11) compared with T5QR and QuReTeC. These observations demonstrate the decent robustness of our EDIRCS to the difficult long context.

5.6 Qualititve Analysis

To further gain qualitative insights, we show and analyze some concrete rewriting examples in Table 3, where EDIRCS achieves better MRR with BM25 than T5QR, QuReTeC, and Human. We find that the rewrite generated by T5QR is coherent while some important information may miss (e.g., "this → normal" for the left example). By contrast, EDIRCS accurately select those important tokens from the session to complement the missing semantics of the current query. Moreover, compared with T5QR, QuReTeC, and Human, a notable advantage of EDIRCS is that some new tokens that are not in the session but are helpful to retrieve the gold passages can be incorporated into the rewrite (e.g., Hyperglycemia and Philadelphia) thanks to the proposed search-oriented learning.

6 Conclusion

In this paper, we present a CQR model EDIRCS based on the text editing paradigm for efficient and effective conversational search. EDIRCS is augmented with two novel search-oriented objectives that can leverage the downstream retrieval information to improve the learning of query rewriting toward conversational search. Experiments on three conversational search datasets demonstrate the superior effectiveness and efficiency of EDIRCS over existing CQR models. We also show that EDIRCS has decent robustness to out-of-domain search dialogues and difficult long context. Future directions include exploring more search-oriented objectives and simultaneously improving the coherence and retrieval performance of the rewrites.

Limitations

This section does not count toward the page limit.

As illustrated in § 5.2 and shown in Table 3, the coherence of the rewrite generated by EDIRCS is not as good as that generated by purely autoregressive rewriters (e.g, T5QR). This may affect the performance of EDIRCS when using dense retrievers. Possible solutions include using an additional token reordering model (Chowdhury et al., 2021) to improve the rewrite coherence or injecting the coherence signals (Hao et al., 2021) or token positions information (Mallinson et al., 2022) into the learning of EDIRCS in an end-to-end way. Another concern is that the effect of our text editing-based model may be limited for a few long-tail cases where many expected rewrite tokens are not in the input session. How to better deal with the search dialogues whose search intents are too implicit or vague to be accurately expressed by inferring from the dialogue context alone is a valuable direction for further improvements of our model.

Acknowledgements

Zhicheng Dou is the corresponding author. This work was supported by National Key R&D Program of China (2022ZD0120103), National Natural Science Foundation of China (62272467), Beijing Outstanding Young Scientist Program (BJJWZYJH012019100020098), the Outstanding Innovative Talents Cultivation Funded Programs 2023 of Renmin University of China, and Public Computing Cloud, Renmin University of China. The work was partially done at Beijing Key Laboratory of Big Data Management and Analysis Methods, and Key Laboratory of Data Engineering and Knowledge Engineering, MOE.

References

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-domain question answering goes conversational via question rewriting. In *NAACL-HLT*, pages 520–534. Association for Computational Linguistics.

Zhiyu Chen, Jie Zhao, Anjie Fang, Besnik Fetahu, Rokhlenko Oleg, and Shervin Malmasi. 2022. Reinforced question rewriting for conversational question answering.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *EMNLP*, pages 2174–2184. Association for Computational Linguistics.

Somnath Basu Roy Chowdhury, Faeze Brahman, and Snigdha Chaturvedi. 2021. Is everything in order? A simple way to order sentences. In *EMNLP (1)*, pages 10769–10779. Association for Computational Linguistics.

Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. Trec cast 2019: The conversational assistance track overview. In *In Proceedings of TREC*.

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

Laura Dietz, Manisha Verma, Filip Radlinski, and Nick Craswell. 2017. Trec complex answer retrieval overview. In *TREC*.

Jianfeng Gao, Chenyan Xiong, Paul Bennett, and Nick Craswell. 2022. Neural approaches to conversational information retrieval. *arXiv preprint arXiv:2201.05176*.

Jie Hao, Linfeng Song, Liwei Wang, Kun Xu, Zhaopeng Tu, and Dong Yu. 2021. RAST: domain-robust dialogue rewriting as sequence tagging. In *EMNLP (1)*, pages 4913–4924. Association for Computational Linguistics.

Mengzuo Huang, Feng Li, Wuhe Zou, and Weidong Zhang. 2021. SARG: A novel semi autoregressive generator for multi-turn incomplete utterance restoration. In *AAAI*, pages 13055–13063. AAAI Press.

Lisa Jin, Linfeng Song, Lifeng Jin, Dong Yu, and Daniel Gildea. 2022. Hierarchical context tagging for utterance rewriting. In *AAAI*, pages 10849–10857. AAAI Press.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Trans. Big Data*, 7(3):535–547.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781. Association for Computational Linguistics.

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

Carlos Lassance and Stéphane Clinchant. 2022. An efficiency study for SPLADE models. In *SIGIR*, pages 2220–2226. ACM.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021a. Contextualized query embeddings for conversational search. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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

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

Jonathan Mallinson, Jakub Adámek, Eric Malmi, and Aliaksei Severyn. 2022. Edit5: Semi-autoregressive text-editing with T5 warm-start. EMNLP 2022.

Eric Malmi, Yue Dong, Jonathan Mallinson, Aleksandr Chuklin, Jakub Adámek, Daniil Mirylenka, Felix Stahlberg, Sebastian Krause, Shankar Kumar, and Aliaksei Severyn. 2022. Text generation with text-editing models. *CoRR*, abs/2206.07043.

Kelong Mao, Zhicheng Dou, and Hongjin Qian. 2022a. 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 (SIGIR)*.

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

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval for open-domain question answering. In *ACL/IJCNLP (1)*, pages 4089–4100. 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. In *CoCo@ NIPS*.

Machel Reid and Victor Zhong. 2021. LEWIS: levenshtein editing for unsupervised text style transfer. In *ACL/IJCNLP (Findings)*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3932–3944. Association for Computational Linguistics.

Marco Del Tredici, Gianni Barlacchi, Xiaoyu Shen, Weiwei Cheng, and Adrià de Gispert. 2021. Question rewriting for open-domain conversational QA: best practices and limitations. In *CIKM*, pages 2974–2978. ACM.

Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2021a. Question rewriting for conversational question answering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 355–363.

Svitlana Vakulenko, Nikos Voskarides, Zhucheng Tu, and Shayne Longpre. 2021b. A comparison of question rewriting methods for conversational passage retrieval. In *ECIR (2)*, volume 12657 of *Lecture Notes in Computer Science*, pages 418–424. Springer.

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

Zeqiu Wu, Yi Luan, Hannah Rashkin, David Reitter, and Gaurav Singh Tomar. 2022. Conqrr: Conversational query rewriting for retrieval with reinforcement learning.

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

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)*, pages 1933–1936.

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)*.

A Gold Label Annotation

In this section, we introduce the gold label annotation for our conversational query editing.

For the session token selection, we wish to annotate the session tokens which also appear in the manual rewrite as *Retain* and annotate the other tokens as *Delete*. Considering that a token may occur multiple times in the session, we iteratively adopt a Greedy Longest Common Substring (GLCS) algorithm for label annotation. Specifically, given the input session s and its manual rewrite t, we first find their longest common substring (*lcs*) $s[a:b] == t[c:d]$, where a, b, c, d are the start and end positions. If there are multiple *lcs*, we greedily choose the one with the smallest a, which is closest to the current query since the current query is at the beginning of s. We annotate all the tokens of $s[a:b]$ as *Retain*. Then, we remove this *lcs* to update both s and t to be $s = s[:a] + s[b:]$ and $t = t[:c] + t[d:]$, and iteratively perform the above annotating process until the length of *lcs* is zero. Finally, we annotate all the remaining tokens in s as *Delete*.

For the new token generation, we sequentially extract the unique tokens which do not in the input session from the manual rewrite to form the gold sequence of generation.

The pseudo-code of the whole annotation process is shown in Algorithm 1.

Algorithm 1 Gold Label Annotation

```
Require: The input session graph $s$ and its manual rewrite $t$.

# Annotation for the session token selection.
while True do
    $a, b, c, d$ = GLCS($s, t$) # get the greedy longest common substring.
    if a == b then
        break
    end if
    Annotating the tokens in $s[a:b]$ as Retain.
    $s = s[:a] + s[b:]$
    $t = t[:c] + t[d:]$
end while
Annotating all the tokens of $s$ as Delete.

# Annotation for the new token generation.
seq = ""
for $i$ in $range(0, \text{len}(t))$ do
    if $t[i]$ not in $s$ and $t[i]$ not in $seq$ then
        $seq = seq + t[i]$
    end if
end for
return seq # the gold sequence of generation.
```

B Dataset Details

In this section, we provide more detailed description about the three used conversational search datasets.

QReCC: It is a large-scale dataset for conversational question answering, which contains 14K information-seeking conversations with 80K query-answer pairs originated from the training set of CAsT-19 (Dalton et al., 2020), QuAC (Choi et al., 2018), and NQ (Choi et al., 2018) with manually generated follow-up queries. Each query has a response answer and a corresponding human rewrite. The entire text corpus for retrieval includes 54M passages and the query-passage relevance is labeled through a heuristic span matching method based on the answer.

CAsT-19 and CAsT-20: They are two widely used conversational search evaluation datasets released by TREC Conversational Assistance Track (CAsT). There are only 50 and 25 human-written information-seeking conversations in CAsT-19 and CAsT-20, respectively, so they are hard to support training and are suitable to be used as the evaluation datasets. The query turns in CAsT-19 can only depend on previous query turns. While in CAsT-20, query turns may also depend on the previous system response. Each query turn in both CAsT-19 and CAsT-20 has a corresponding human rewrite and CAsT-20 additionally provides a canonical response passage for each query turn. The text corpus consists of 38M passages from MS MARCO (Nguyen et al., 2016) and TREC Complex Answer Retrieval (Dietz et al., 2017). More fine-grained query-passage relevance labels is generated by the experts of TREC.