ConvGQR: Generative Query Reformulation for Conversational Search

Fengran Mo[1], **Kelong Mao**[2], **Yutao Zhu**[1], **Yihong Wu**[1], **Kaiyu Huang**[3*], **Jian-Yun Nie**[1*]
[1]University of Montreal, Quebec, Canada
[2]Gaoling School of Artificial Intelligence, Renmin University of China
[3]Institute for AI Industry Research, Tsinghua University, Beijing, China
{fengran.mo,yihong.wu}@umontreal.ca, yutaozhu94@gmail.com
nie@iro.umontreal.ca, mkl@ruc.edu.cn, huangkaiyu@air.tsinghua.edu.cn

Abstract

In conversational search, the user's real search intent for the current conversation turn is dependent on the previous conversation history. It is challenging to determine a good search query from the whole conversation context. To avoid the expensive re-training of the query encoder, most existing methods try to learn a rewriting model to de-contextualize the current query by mimicking the manual query rewriting. However, manually rewritten queries are not always the best search queries. Thus, training a rewriting model on them would lead to sub-optimal queries. Another useful information to enhance the search query is the potential answer to the question. In this paper, we propose **ConvGQR**, a new framework to reformulate conversational queries based on generative pre-trained language models (PLMs), one for query rewriting and another for generating potential answers. By combining both, ConvGQR can produce better search queries. In addition, to relate query reformulation to the retrieval task, we propose a knowledge infusion mechanism to optimize both query reformulation and retrieval. Extensive experiments on four conversational search datasets demonstrate the effectiveness of ConvGQR.

1 Introduction

Conversational search (Gao et al., 2022) is a rapidly developing branch of information retrieval, which aims to satisfy complex information needs through multi-turn conversations. The main challenge is to determine users' real search intents based on the interaction context and formulate good search queries accordingly. Existing methods can be roughly categorized into two groups. The first group directly uses the whole context as a query and trains a model to determine the relevance between the long context and passages (Qu et al., 2020; Hashemi et al., 2020; Yu et al., 2021; Lin et al., 2021b; Mao et al., 2022a,b; Kim and Kim, 2022; Mo et al., 2023). This approach requires additional training of retriever to take the long context as input, which is not always feasible (Wu et al., 2021). What is available in practice is a general retriever (*e.g.*, ad-hoc search retriever) that uses a stand-alone query. The second group of approaches aims at producing a de-contextualized query using query reformulation techniques (Elgohary et al., 2019). Such a query can be submitted to any *off-the-shelf* retrievers. We focus on this second approach.

Two types of query reformulation techniques have been widely studied in the literature, *i.e.*, *query rewriting* and *query expansion*. The former trains a generative model to rewrite the current query to mimic the human-rewritten one (Yu et al., 2020; Vakulenko et al., 2021a), while the latter focuses on expanding the current query by relevant terms selected from the context (Kumar and Callan, 2020; Voskarides et al., 2020). Although both approaches achieve promising results, they are all studied separately. Two important limitations are observed: (1) Query rewriting and query expansion can produce different effects. Query rewriting tends to deal with ambiguous queries and add missing tokens, while query expansion aims to add supplementary information to the query. Both effects are important for query reformulation. It is thus beneficial to use both of them. (2) Previous query rewriting models have been optimized to produce human-rewritten queries, independently from the passage ranking task. Even though human-rewritten queries usually perform better than the original queries, existing studies have shown that they may not be the best search queries alone (Lin et al., 2021b; Wu et al., 2021). Therefore, it is useful to incorporate additional criteria directly related to ranking performance when reformulating a query. As shown in Fig. 1 (left), although the human-rewritten query recovers the crucial missing information (i.e. "goat") from the context, it is

[*]Corresponding authors.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 4998–5012
July 9-14, 2023 ©2023 Association for Computational Linguistics

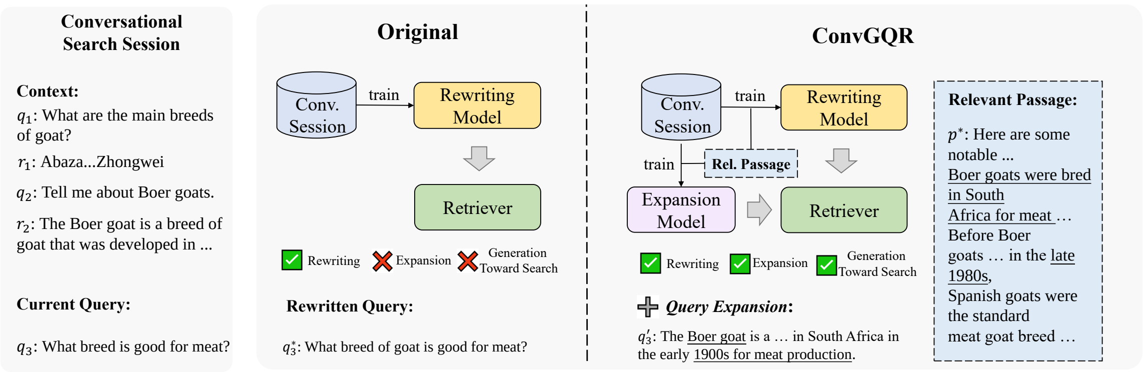
Conversational Search Session
Context:
q_1: What are the main breeds of goat?
r_1: Abaza...Zhongwei
q_2: Tell me about Boer goats.
r_2: The Boer goat is a breed of goat that was developed in ...
Current Query:
q_3: What breed is good for meat?
Original
Conv. Session
train
Rewriting Model
Retriever
Rewriting
Expansion
Generation Toward Search
Rewritten Query:
q_3^*: What breed of goat is good for meat?
ConvGQR
Conv. Session
train
Rewriting Model
train
Rel. Passage
Expansion Model
Retriever
Rewriting
Expansion
Generation Toward Search
Query Expansion:
q_3': The Boer goat is a … in South Africa in the early 1900s for meat production.
Relevant Passage:
p^*: Here are some notable ... Boer goats were bred in South Africa for meat … Before Boer goats … in the late 1980s, Spanish goats were the standard meat goat breed …

Figure 1: An example of conversational search session and the high-level comparison between the original method and our ConvGQR. The dashed box illustrates the potential connection (underline) between the relevant passage and expansion terms.

still possible to further improve the search query.

To tackle these problems, we propose ConvGQR, a new **G**enerative **Q**uery **R**eformulation framework for **Conv**ersational search. It combines query rewriting with query expansion. The right side of Fig. 1 illustrates the differences between ConvGQR and the existing query rewriting method. In addition to query rewriting based on human-rewritten queries, ConvGQR also learns to generate the potential answer of the query (*e.g.*, the answer in the downstream question-answering task) and uses it for query expansion. This strategy is motivated by the fact that a passage containing the generated potential answer is more likely a relevant passage, because either the generated answer is the right answer, or it may co-occur with the right answer in the same passage. The final query reformulation model is trained by combining both query rewriting and query expansion criteria in the loss function. Moreover, the learning of both query rewriting and expansion are guided by the relevant passage information through our knowledge infusion mechanism to encourage query generation toward better search performance. We carry out extensive experiments on four conversational search datasets using both dense and sparse retrievers, and the results show that our method outperforms most existing query reformulation methods. Our further analysis confirms the complementary contributions of query rewriting and query expansion.

Our contributions are summarized as follows: (1) We propose ConvGQR to integrate query rewriting and query expansion for conversational search. In particular, query expansion is performed by adding the generated potential answer by a generative PLM. This is a way to exploit PLM's capability of capturing rich world knowledge. (2) We further design a knowledge infusion mechanism to optimize query reformulation with the guidance of passage retrieval. (3) We demonstrate the effectiveness of ConvGQR with two off-the-shelf retrievers (sparse and dense) on four datasets. Our analysis confirms the complementary effects of both components in conversational search.

2 Related Work

Conversational Query Reformulation The intuitive idea is that a well-formulated search query from the conversation context can be submitted to an *off-the-shelf* retriever for search without modifying it. Query rewriting and query expansion are two typical query reformulation methods. Query rewriting aims to train a rewriting model to mimic human-rewritten queries. This approach is shown to be able to solve the ambiguous problem and recover some missing elements (e.g. anaphora) from the context (Yu et al., 2020; Lin et al., 2020; Vakulenko et al., 2021a; Mao et al., 2023a). However, Wu et al. (2021) and Lin et al. (2021b) argue that the human-rewritten queries might not necessarily be the optimal queries. Wu et al. (2021) enhances the rewriting model by leveraging reinforcement learning. However, it turns out that reinforcement learning requires a long time for training. To be more efficient, Lin et al. (2021b) proposes a query expansion method by selecting the terms via the normalization score of their embeddings but still needs to re-train a retriever. Some earlier query expansion methods (Kumar and Callan, 2020; Voskarides et al., 2020) also focus on selecting useful terms from conversational context. The previous studies show that query rewriting and

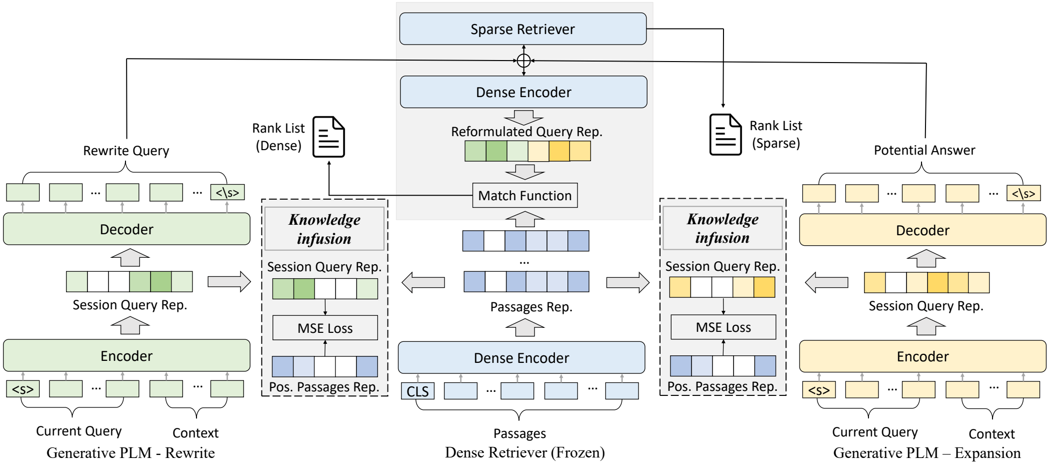

Figure 2: Overview of ConvGQR. Two generative PLMs are used to generate a rewritten query and expansion terms for both training and inference. The knowledge infusion mechanism (dashed boxes) is only applied during training.

query expansion can both enhance the search query and produce better retrieval results. However, these approaches have been used separately. Our ConvGQR model thus integrates both query rewriting and query expansion to reformulate a better conversational query. Moreover, a new knowledge infusion mechanism is used to connect query reformulation with retrieval.

Query Expansion via Potential Answers Earlier studies on question answering (Ravichandran and Hovy, 2002; Derczynski et al., 2008) demonstrate that an effective way to expand a query is to extract answer patterns or select terms that could be possible answers as expansion terms. Recently, some generation-augmented retrieval methods (Mao et al., 2021; Chen et al., 2022) focus on exploiting the knowledge captured in PLMs (Roberts et al., 2020; Brown et al., 2020) to generate the potential answer as expansion terms. We draw inspiration from these studies and apply the idea to conversational search.

3 Methodology

3.1 Task Formulation

We formulate the conversational search task in this paper as retrieving the relevant passage p from a large passage collection C for the current user query q_i given the conversational historical context $H^k = \{q_k, r_k\}_{k=1}^{i-1}$, where the q_k and r_k denote the query and the system answer of the k^{th} previous turn, respectively. In this paper, we aim to design a query reformulation model to transform the current query q_i together with the conversational historical context H^k into a de-contextualized rewritten query for conversational search.

3.2 Our Approach: ConvGQR

A first desired behavior of query reformulation is to produce a similar rewritten query as a human expert. This will solve some ambiguities arisen in the current query (*e.g.*, omission and coreference). So, query rewriting will be an integral part of our approach. Query rewriting can be cast as a text generation problem: given the query in the current turn and its historical context, we aim to generate a rewritten query. Inspired by the large capability of PLM, we rely on a PLM for query rewriting to mimic the human query rewriting process.

However, as the human-rewritten query might not be optimal (Yu et al., 2020; Anantha et al., 2021) and the standard query rewriting models are agnostic to the retriever (Lin et al., 2021b; Wu et al., 2021), a query rewriting model alone cannot produce the best search query. Therefore, we also incorporate a component to expand the query by adding additional terms that are likely involved in relevant passages. Several query expansion methods can be used. In this paper, we choose to use the following one which has proven effective in question answering (Mao et al., 2021; Chen et al., 2022): we use the current query and its context to generate a potential answer to the question (query). The generated answer is used as expansion terms. This approach leverages the large amount of world knowledge implicitly captured in a large PLM[1]. The generated potential answer can be useful for

[1] As shown by the recent success of ChatGPT, PLMs can generate correct answers to a large variety of questions.

passage retrieval in two situations: (1) the generated answer is correct, so a passage containing the same answer could be favored; (2) the generated answer is not a correct answer, but it co-occurs with a correct answer in a passage. This can also help determine the correct passage, and this is indeed the very assumption behind many query expansion approaches used in IR. Motivated by this, we use another PLM to generate the potential answer to expand the current query.

The overview of our proposed ConvGQR is depicted in Fig. 2. It contains three main components: query rewriting, query expansion, and knowledge infusion mechanism. The last component connects query reformulation and retrieval.

3.2.1 Query Reformulation by Combining Rewriting and Expansion

Both query rewriting and expansion use the historical context $H^k = \{q_k, r_k\}_{k=1}^{i-1}$ concatenated with the current query q_i as input. Similar to the input used in Wu et al. (2021), a separation token "[SEP]" is added between each turn and the turns are concatenated in reversed order, as in Eq. 1.

$$\mathrm{S} = \texttt{[CLS]}\ q_i\ \texttt{[SEP]}\ q_{i-1} \cdots q_1\ \texttt{[SEP]}. \quad (1)$$

Query Rewriting The objective of the query rewriting model is to induce a function $\mathcal{M}(H^k, q_i) = q^*$ based on a generative PLM, where q^* is a sequence used as the supervision signal (which is a human-rewritten query in the training data). Then, the information contained in H^k but missing in q_i can be added to approach q^*. Finally, the overall objective can be viewed as optimizing the parameter $\theta_{\mathcal{M}}$ of the function $\mathcal{M}$ by maximum likelihood estimation:

$$\theta_{\mathcal{M}} = \arg\max_{\theta_{\mathcal{M}}} \prod_{k=1}^{i-1} \Pr\left(q^* | \mathcal{M}\{H^k, q_i\}, \theta_{\mathcal{M}}\right). \quad (2)$$

Query Expansion Recent research demonstrates that the current PLMs have the ability to directly respond to a question as a close-book question answering system (Adlakha et al., 2022) through its captured knowledge. Although the correctness of the generated answer is not guaranteed, the potential answer can still act as useful expansion terms (Mao et al., 2021), which can guide the search toward a passage with the potential answer or a similar answer. To train the generation process, we leverage the gold answer r^* for each query turn as the training objection. r^* could be a short entity, a consecutive segment of text, or even non-consecutive text segments, depending on the dataset. In inference for a new query, the potential answers are generated by the query expansion model and used to expand the previously rewritten query.

The final form of the reformulated query is the concatenation of the rewritten query and the generated potential answer. The two generative PLMs for rewriting and expansion are fine-tuned with the negative log-likelihood loss to predict the corresponding target with an input sequence $\{w_t\}_{t=1}^{T}$ as Eq. 3, however, with different training data.

$$\mathcal{L}_{\text{gen}} = -\sum_{t=1}^{T} \log \left(\Pr(w_t | w_{1:t-1}, H^k, q_i)\right). \quad (3)$$

3.2.2 Knowledge Infusion Mechanism

An important limitation of the existing generative conversational query reformulation methods is that they ignore the dependency between generation and retrieval. They are trained independently. To address this issue, we propose a knowledge infusion mechanism to optimize both query reformulation and search tasks during model training. The intuition is to require the generative model to generate a query representation that is similar to that of a relevant passage. If the hidden states of the generative model contain the information of the relevant passage, the queries generated by these representations would be able to improve the search results because of the increased semantic similarity.

To achieve this goal, an effective way is to inject the knowledge included in the relevant passage representation into the query representation when fine-tuning the generative PLMs. Concretely, we first deploy an *off-the-shelf* retriever acting as an encoder to produce a representation $\mathbf{h}_{p_+}$ for the relevant passage. To maintain consistency, the retriever is the same as the one we use for search. Thus, the representation space for passages is kept the same for both query reformulation and retrieval stages. Once the session query representation $\mathbf{h}_S$ is encoded by the generative model, we distill the knowledge of $\mathbf{h}_{p_+}$ and infuse it into the $\mathbf{h}_S$ by minimizing the Mean Squared Error (MSE) as Eq. 4. Both $\mathbf{h}_S$ and $\mathbf{h}_{p_+}$ are sequence-level representations based on the first special token "[CLS]". Finally, the overall training objective $\mathcal{L}_{\text{ConvGQR}}$ consists of query generation loss $\mathcal{L}_{\text{gen}}$ and retrieval loss $\mathcal{L}_{\text{ret}}$. A weight factor α is used to balance the influence of query generation and retrieval.

$$\mathcal{L}_{\text{ret}} = \text{MSE}(\mathbf{h}_S, \mathbf{h}_{p_+}), \tag{4}$$

$$\mathcal{L}_{\text{ConvGQR}} = \mathcal{L}_{\text{gen}} + \alpha \cdot \mathcal{L}_{\text{ret}}. \tag{5}$$

3.3 Training and Inference

Two generative models with different targets for query rewriting and expansion are trained separately. The final output of the ConvGQR is the concatenation of the rewritten query and the generated potential answer. The knowledge infusion mechanism is applied only for the training stage, which guides optimization toward both generation and retrieval. The dense retriever is frozen to encode passages for generative PLMs training.

3.4 Retrieval Models

We apply ConvGQR to both dense and sparse retrieval models. We use ANCE (Xiong et al., 2020) fine-tuned on the MS MARCO (Bajaj et al., 2016), which achieves state-of-the-art performance on several retrieval benchmarks, as the dense retriever. The sparse retrieval is the traditional BM25.

4 Experiments

Datasets Following previous studies (Wu et al., 2021; Kim and Kim, 2022), four conversational search datasets are used for our experiments. The TopiOCQA (Adlakha et al., 2022) and QReCC (Anantha et al., 2021) datasets are used for normal query reformulation training. Two other widely used TREC CAsT datasets (Dalton et al., 2020, 2021) are only used for zero-shot evaluation as no training data is provided. The statistics and more details are provided in Appendix A.

Evaluation Metrics To evaluate the retrieval results, we use four standard evaluation metrics: MRR, NDCG@3, Recall@10 and Recall@100, as previous studies (Anantha et al., 2021; Adlakha et al., 2022; Mao et al., 2022a). We adopt the `pytrec_eval` tool (Van Gysel and de Rijke, 2018) for metric computation.

Baselines We mainly compare `ConvGQR` with the following query reformulation (QR) baselines for both dense and sparse retrieval: (1) `Raw`: The query of current turn without reformulation. (2) `GPT2QR` (Anantha et al., 2021): A strong GPT-2 (Radford et al., 2019) based QR model. (3) `CQE-sparse` (Lin et al., 2021b): A weakly-supervised method to select important tokens only from the context via contextualized query embeddings. (4) `QuReTeC` (Voskarides et al., 2020): A weakly-supervised method to train a sequence tagger to decide whether each term contained in historical context should be added to the current query. (5) `T5QR` (Lin et al., 2020): A strong T5-based (Raffel et al., 2020) QR model. (6) `ConvDR` (Yu et al., 2021): A strong ad-hoc search retriever fine-tuned on conversational search data using knowledge distillation between the rewritten query representation and the historical context representation. (7) `CONQRR` (Wu et al., 2021): A reinforcement-learning and T5-based QR model which adopts both BM25 and conversational fine-tuned T5-encoder as retrievers. Note that `CQE-sparse` and `ConvDR` need to train a new conversational query encoder to determine the relevance between the long context and passages, while the other baseline methods and our `ConvGQR` are based on the off-the-shelf retriever only.

For *zero-shot* scenario, in addition to the `QuReTeC` method originally fine-tuned on QuAC datasets (Choi et al., 2018), we also perform comparisons with (8) `Transformer++` (Vakulenko et al., 2021a): A GPT-2 based QR model fine-tuned on CANARD dataset (Elgohary et al., 2019). (9) `Query Rewriter` (Yu et al., 2020): A GPT-2 based QR model fine-tuned on large-scale search session data. Besides, the results of `Human-Rewritten` queries in the original datasets are also provided.

Implementation Details We implement the generative PLMs for `ConvGQR` based on T5-base (Raffel et al., 2020) models. When fine-tuning the generative PLMs, the dense retriever is frozen and acts as a passage encoder. For the zero-shot scenario, we use the generative models trained on QReCC to produce the reformulated queries and retrieve relevant passages. The dense retrieval and sparse retrieval (BM25) are performed using Faiss (Johnson et al., 2019) and Pyserini (Lin et al., 2021a), respectively. More details are provided in Appendix A and our released code[2].

4.1 Main Results

Main evaluation results on QReCC and TopiOCQA are reported in Table 1.

[2] `https://github.com/fengranMark/ConvGQR`

Type	Method	QReCC				TopiOCQA			
		MRR	NDCG@3	R@10	R@100	MRR	NDCG@3	R@10	R@100
Dense	Raw	10.2	9.3	15.7	22.7	4.1	3.8	7.5	13.8
	GPT2QR	33.9	30.9	53.1	72.9	12.6	12.0	22.0	33.1
	CQE-sparse	32.0	30.1	51.3	70.9	-	-	-	-
	QuReTeC	35.0	32.6	55.0	72.9	11.2	10.5	20.2	34.4
	T5QR	34.5	31.8	53.1	72.8	23.0	22.2	37.6	54.4
	ConvDR	38.5	35.7	58.2	77.8	-	-	-	-
	CONQRR	41.8	-	**65.1**	**84.7**	-	-	-	-
	ConvGQR (Ours)	**42.0**‡	**39.1**‡	63.5	81.8	**25.6**‡	**24.3**‡	**41.8**‡	**58.8**‡
	Human-Rewritten	38.4	35.6	58.6	78.1	-	-	-	-
Sparse	Raw	6.5	5.5	11.1	21.5	2.1	1.8	4.0	9.2
	GPT2QR	30.4	27.9	50.5	82.3	6.2	5.3	12.4	26.4
	CQE-sparse	31.8	29.2	52.9	83.4	-	-	-	-
	QuReTeC	34.0	30.5	55.5	86.0	8.5	7.3	16.0	31.3
	T5QR	33.4	30.2	53.8	86.1	11.3	9.8	22.1	44.7
	CONQRR	38.3	-	60.1	**88.9**	-	-	-	-
	ConvGQR (Ours)	**44.1**‡	**41.0**‡	**64.4**‡	88.0	**12.4**‡	**10.7**‡	**23.8**‡	**45.6**‡
	Human-Rewritten	39.7	36.2	62.5	98.5	-	-	-	-

Table 1: Performance of dense and sparse retrieval with query reformulation methods on two datasets. ‡ denotes significant improvements with t-test at $p < 0.05$ over all compared methods (except CONQRR). **Bold** and underline indicate the best and the second best result (except Human-Rewritten).

	QReCC		TopiOCQA	
	MRR	NDCG@3	MRR	NDCG@3
ConvGQR	**42.0**	**39.1**	**25.6**	**24.3**
- infusion	41.5	38.7	25.0	23.7
- expansion	36.9	33.9	24.6	23.3
- both	36.4	33.5	23.4	22.5

Table 2: Ablation study of different components.

We find that ConvGQR achieves significantly better performance on both datasets in terms of MRR and NDCG@3 and outperforms other methods on most metrics, either with dense retrieval or sparse retrieval. For example, on QReCC with sparse retrieval, it improves 15.1% MRR and 33.9% NDCG@3 over the second best results. This indicates the strong capability of ConvGQR on retrieving relevant passages at top positions. These results demonstrate the strong effectiveness of our method. Besides, we notice that CONQRR, which also leverages the downstream retrieval information but with reinforcement learning, may achieve better performance on some recall metrics, indicating that the downstream retrieval information is helpful to conversational search and should be carefully exploited.

Moreover, we find ConvGQR can even perform better than human-rewritten queries on QReCC. It confirms our earlier assumption that the human-rewritten query (oracle query) is not the silver bullet for conversational search. This finding is consistent with some recent studies (Lin et al., 2021b; Wu et al., 2021; Mao et al., 2023b). The improvements of ConvGQR over human-rewritten queries are mainly attributed to our query expansion and knowledge infusion, which introduce retrieval signals to the learning of query reformulation.

4.2 Ablation Study

Compared to a standard query rewriting method, our proposed ConvGQR has two additional components, *i.e.*, a query expansion component based on generated potential answers and a knowledge infusion mechanism. We investigate the impact of different components by conducting an ablation study on both QReCC and TopiOCQA. The results are shown in Table 2. We observe that removing any component leads to performance degradation and removing all of them drops the most. In fact, when both components are removed, ConvGQR degenerates to the T5QR model. The improvement of ConvGQR over T5QR directly reflects the gains brought by query expansion and knowledge infusion. The above analysis confirms the effectiveness of the added components.

	CAsT-19		CAsT-20	
	MRR	NDCG@3	MRR	NDCG@3
`Transformer++`	69.6	**44.1**	29.6	18.5
`Query Rewriter`	66.5	40.9	37.5	25.5
`CQE-Sparse`	67.1	39.9	42.3	27.1
`QuReTeC`	68.9	43.0	43.0	28.7
`T5QR`	70.1	41.7	42.3	29.9
`ConvGQR`	**70.8**‡	43.4	**46.5**‡	**33.1**‡
`Human-Rewritten`	74.0	46.1	59.1	42.2

Table 3: Zero-shot dense retrieval performance of different query reformulation methods. ‡ denotes significant improvements with t-test at $p < 0.05$ over all compared methods. **Bold** indicates the best result (except `Human-Rewritten`).

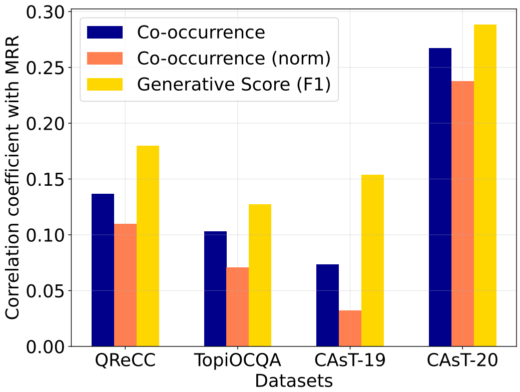

Figure 3: Pearson correlation coefficient (PCC) between three generative evaluation metrics with MRR scores.

4.3 Zero-Shot Analysis

The zero-shot evaluation is conducted on CAsT datasets to test the transferability of `ConvGQR`. By comparing with the other strongest QR methods in Table 3, we have the following main findings.

The `ConvGQR` outperforms all the other methods on the more difficult dataset CAsT-20 and matches the best results on CAsT-19, which demonstrates its strong transferability to new datasets. The human-rewritten queries in CAsT datasets achieve the highest retrieval scores, because they have been formulated carefully by experts for search. This observation is different from the results of QReCC in Table 1, for which query rewriting has been done by crowd-sourcing. However, this observation should not lead to the conclusion that human-rewritten queries should be used as the gold standard for the training of query rewriting, because it is difficult to obtain a large number of high-quality human-rewritten queries as in the CAsT datasets. As one can see in Table 7, these datasets only contain a very limited number of queries. Therefore, the generated expansion terms based on the knowledge captured in PLM is still a valuable means to obtain superior performance for new queries.

In addition, combining Table 1 and Table 3, we notice that the effectiveness of `ConvGQR` for dense retrieval varies with datasets. A potential reason is the different degrees of co-occurrence of generated expansion terms within their relevant passages. This will be further analyzed in Section 4.4.

4.4 Impact of Generated Answer for Retrieval

The aforementioned hypothesis of the `ConvGQR` for query expansion is that the PLM-generated potential answers might contain useful expansion terms that co-occur with the right answer in the relevant passages. To see how expansion terms are related to retrieval performance, we use three metrics to analyze their correlation with the retrieval score.

Correlation Analysis Specifically, for each rewritten query with expansion terms, we first calculate the token overlaps between the generated answers and the relevant passages, which can measure their co-occurrence. However, the potential problem is that the generated answers or relevant passages are of variable lengths. Therefore, we further normalize it by the length of its corresponding relevant passage. Besides, we compute the F1 scores between the generated answers and the gold answers to explore if the generation quality has an impact on retrieval effectiveness. Finally, we calculate the Pearson Correlation Coefficient (PCC) for all these three generative evaluation metrics with the respective MRR scores of every reformulated query.

The results are shown in Fig. 3. The relative PCC value can reflect the helpfulness of generated answers for different datasets to some extent. For example, the PCC of QReCC and CAsT-20 are higher than TopiOCQA and CAsT-19, suggesting that the potential answers are more useful in the first datasets. This is consistent with our previous experimental observations that QReCC and CAsT-20 have larger improvements by `ConvGQR` compared to TopiOCQA and CAsT-19. Thus, the co-occurrence between generated answer and the relevant passage is crucial for the retrieval effectiveness for `ConvGQR`.

The PCC of generative score F1 is the highest

	Query Form	QReCC				TopiOCQA			
		MRR	NDCG@3	R@10	R@100	MRR	NDCG@3	R@10	R@100
Dense	Rewritten Query	36.4	33.5	56.6	76.0	23.4	22.5	39.8	56.2
	Generated Answer	33.4	30.6	51.9	70.4	3.7	3.2	6.9	14.4
	Concatenation	**41.5**	**38.7**	**63.7**	**81.4**	**25.0**	**23.7**	**42.3**	**57.9**
Sparse	Rewritten Query	33.8	30.6	54.3	86.7	11.3	9.8	22.1	**44.7**
	Generated Answer	33.7	31.3	49.2	69.6	2.0	1.7	3.9	9.6
	Concatenation	**43.4**	**40.6**	**63.8**	**88.1**	**11.6**	**10.2**	**22.5**	42.8

Table 4: Performance of both dense and sparse retrieval on different reformulated query forms.

among the three metrics, which indicates its strong correlation with retrieval effectiveness. However, utilizing generated answers alone as search queries could produce false positive results as we will demonstrate in the subsequent analysis. As a result, it may not reflect the genuine correlation strength in comparison to the co-occurrence metric.

Effects of Different Generated Forms We show the performance of using three different forms of generated queries, i.e. the rewritten query, the generated answer, and the concatenation of them, as the reformulated query for retrieval in Table 4. We find that using the concatenation of both significantly outperforms the two other forms alone, indicating that these two forms can complement each other to achieve better retrieval performance, which confirms again our initial hypothesis. Besides, we find that using the rewritten query alone performs better than using the generated answer, especially on TopiOCQA. The potential reason is the different forms of answers in the datasets: QReCC is more related to factoid questions than TopiOCQA. The correct answer with non-factoid question type is more difficult for a PLM to directly generate. So, the generated answers may be of less utility.

4.5 Impact of Knowledge Infusion Loss

We conduct an analysis of the impact of two knowledge infusion loss functions trying to approach the query representation to that of the relevant passage: contrastive learning (CL) loss and mean square error (MSE) loss. They correspond to Eq. 6 and Eq. 4. The difference between them is that the MSE loss only considers positive passages $\mathbf{h}_{p_+}$ while the CL loss also considers negative passages $\mathbf{h}_{p_-}$ for model training as follows:

$$\mathcal{L}_{\text{CL}} = -\log \frac{e^{(\mathbf{h}_S \cdot \mathbf{h}_{p_+})}}{e^{(\mathbf{h}_S \cdot \mathbf{h}_{p_+})} + \sum_{\mathbf{P}^-} e^{(\mathbf{h}_S \cdot \mathbf{h}_{p_-})}}. \quad (6)$$

	Type	MRR	NDCG@3	R@10	R@100
CL	Dense	41.7	38.9	62.8	80.9
MSE	Dense	42.0	39.1	63.5	81.8
CL	Sparse	43.9	40.9	64.0	87.5
MSE	Sparse	44.1	41.0	64.4	88.0

Table 5: Retrieval performance of two knowledge infusion loss functions on QReCC.

We compare the conversational search results of the reformulated queries training by these two loss functions on QReCC and report the results in Table 5. We can find that the reformulated queries trained by CL loss are slightly worse than those with MSE loss. In most previous literature (Xiong et al., 2020; Karpukhin et al., 2020), the CL loss usually performs better for dense retrieval training, thus we expected similar results. The reason for the opposite result might be as follows: since ConvGQR is mainly a generation task rather than a retrieval task, a positive passage can provide a clear signal to instruct the right direction for the target generation, while the additional negative passages used in CL loss only suggest the wrong directions to avoid. Intuitively, the generation objective has only one correct optimization direction but many wrong directions in the high dimensional latent space. This may make it difficult for the knowledge infusion mechanism to determine the correct direction to follow, resulting in sub-optimal queries. Note that despite the above observation, our method ConvGQR trained with CL loss still outperforms most of the existing baselines.

4.6 Case Study

We finally show a case in Table 6 to help understand more intuitively the impact of expansion terms on ConvGQR. The model is expected to rewrite the query and generate the potential answer toward the human-rewritten query and the gold answer. Although the model produces the same rewritten query as the human, which solves the anaphora

Context: (QReCC Session 2)
q_1: What are the main breeds of goat?
r_1: Abaza...Zhongwei
q_2: Tell me about boer goats.
r_2: The Boer goat is a breed of goat that was developed ... Their name is derived from the Afrikaans (Dutch) ...
Current Query: q_3: What breed is good for meat?
Human-Rewritten: q_3^*: What breed of goat is good for meat?
ConvGQR Reformulated Query:
$\hat{q_3}$: What breed of goat is good for meat? The Boer goat is a breed of goat that was developed in South Africa in the early 1900s for meat production.
Relevant Passage:
p^*: Here are some notable breeds ... Boer goats were bred in South Africa for meat ... *Before Boer goats became available in the United States in the late 1980s, Spanish goats were the standard meat goat breed ...*
Dense Score: 0.06 (Human-Rewritten) **1.00 (Ours)**
Sparse Score: 0.03 (Human-Rewritten) **0.13 (Ours)**

Table 6: A successful example illustrating the reformulated query by ConvGQR. Rewritten and expanded query are in blue and orange, respectively. The expansion terms and gold answer are underlined and *italicized* in the relevant passage.

problem of "goat" with the context, the query expansion generated by ConvGQR with the knowledge of "Boer goat" can still improve the performance for both dense and sparse retrieval. In this case, even though the generated answer is not a correct answer to the question, there is a strongly similar description (underlined) that co-occurs with the right answer in the relevant passage. This example shows a typical case where the generated answer can be highly useful expansion terms. More cases are provided in Appendix B.

5 Conclusion

In this paper, we present a new conversational query reformulation framework, ConvGQR, which integrates query rewriting and query expansion toward generating more effective search queries through a new knowledge infusion mechanism. Extensive experimental results on four public datasets demonstrate the superior effectiveness of our model for conversational search. We also carried out detailed analyses to understand the effects of each component of ConvGQR on the performance improvements.

Limitations

Our work demonstrates the feasibility of combining query rewriting and query expansion to reformulate a conversational query for passage retrieval. Within our proposed ConvGQR, the rewriting and expansion are based on two PLMs trained with different data, which introduce additional training load and model parameters for storage. Thus, designing an integrated model that can simultaneously generate the query rewrite and the expanded terms would be a promising improvement to our method. Another limitation is that the potential answer acting as expansion terms could be generated from more resources (*e.g.*, pseudo-relevant feedback and knowledge graph) rather than only relying on the generative PLMs. Besides, more alternative methods for knowledge infusion can be tested to connect query reformulation with the search task.

Acknowledgements

This work has been partly supported by the China Scholarship Council, the Institute for AI Industry Research, Tsinghua University and a discovery grant from the Natural Science and Engineering Research Council of Canada.

References

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

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

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Haonan Chen, Zhicheng Dou, Yutao Zhu, Zhao Cao, Xiaohua Cheng, and Ji-Rong Wen. 2022. Enhancing user behavior sequence modeling by generative tasks for session search. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 180–190.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184.

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

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

Leon Derczynski, Jun Wang, Robert Gaizauskas, and Mark A Greenwood. 2008. A data driven approach to query expansion in question answering. In *Coling 2008: Proceedings of the 2nd workshop on Information Retrieval for Question Answering*, pages 34–41.

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.

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

Helia Hashemi, Hamed Zamani, and W Bruce Croft. 2020. Guided transformer: Leveraging multiple external sources for representation learning in conversational search. In *Proceedings of the 43rd international acm sigir conference on research and development in information retrieval*, pages 1131–1140.

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

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

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

Vaibhav Kumar and Jamie Callan. 2020. Making information seeking easier: An improved pipeline for conversational search. In *Empirical Methods in Natural Language Processing*.

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

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

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

Kelong Mao, Zhicheng Dou, Haonan Chen, Fengran Mo, and Hongjin Qian. 2023a. Large language models know your contextual search intent: A prompting framework for conversational search. *arXiv preprint arXiv:2303.06573*.

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*, pages 176–186.

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*, pages 2935–2946.

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

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

Fengran Mo, Jian-Yun Nie, Kaiyu Huang, Kelong Mao, Yutao Zhu, Peng Li, and Yang Liu. 2023. Learning to relate to previous turns in conversational search. *arXiv preprint arXiv:2306.02553*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca

Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

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

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual meeting of the association for Computational Linguistics*, pages 41–47.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*.

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*, pages 355–363.

Svitlana Vakulenko, Nikos Voskarides, Zhucheng Tu, and Shayne Longpre. 2021b. A comparison of question rewriting methods for conversational passage retrieval. In *European Conference on Information Retrieval*, pages 418–424. Springer.

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

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Zeqiu Wu, Yi Luan, Hannah Rashkin, David Reitter, and Gaurav Singh Tomar. 2021. Conqrr: Conversational query rewriting for retrieval with reinforcement learning. *arXiv preprint arXiv:2112.08558*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

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*, 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*, pages 829–838.

A More Detailed Experimental Setup

A.1 Datasets

The statistics of each dataset are presented in Table 7 and the details are in the following:

QReCC focuses on the query rewriting problem within conversational scenarios by approaching the human-rewritten query. Thus, it provides an oracle query for each conversation turn. We argue that it might not be the optimal one.

TopiOCQA focuses on the challenge of the topic switch under conversational settings, whose sessions are longer than QReCC and thus present more difficulties for query reformulation. Different from QReCC, it does not provide human-rewritten queries.

CAsT-19 and **CAsT-20** are two standard conversational search benchmarks provided in the TREC Conversational Assistance Track (CAsT). The gold answers to each query are the same as their relevant passages. The newer one (CAsT-20) is known to be more challenging.

A.2 Implementation

We implement all models by PyTorch (Paszke et al., 2019) and Huggingface's Transformers (Wolf et al., 2019).

ConvGQR The experiments are conducted on one Nvidia A100 40G GPU. For generative PLMs training, we use Adam optimizer with 1e-5 learning rate and set the batch size as 8. The loss balance weight α is set to 0.5, which is the best according to the hyper-parameter selection of our experiments. For training ConvGQR on QReCC, we use its provided human-rewritten query q^* and gold answer r^* as generation ground-truth for two PLMs. We discard the samples without positive passages for both training and inference as Wu et al. (2021). For TopiOCQA, as it does not provide human-rewritten query q^*, we only use the ground-truth answer r^* to train one generative model for query expansion, and the rewritten query is generated by the model trained on QReCC. Aiming for a fair comparison, we set the maximum generation length (32) the same as CONQRR, which is the current state-of-the-art. The zero-shot evaluation is also based on the generative models trained on QReCC. Following the previous works (Yu et al., 2021; Lin et al., 2021b; Mao et al., 2022a), we set the relevance judgment threshold at 1 and 2 for CAsT-19 and CAsT-20, respectively.

Baselines We implement baselines based on our experimental setting and their open-source code and material. For the normal evaluation, we train QuReTeC, GPT2QR, and T5QR on the corresponding datasets rather than using external resources. Since CONQRR has not released the code and its experimental setting is similar to ours, we directly quote their experimental results on QReCC. The human-rewritten queries are provided in the datasets as annotations but are not available for TopiOCQA. For the zero-shot setting, the Query Rewriter is quoted from the original paper (Yu et al., 2021), and the T5QR is implemented on our own as the query rewriting part. The reformulated queries by Transformer++ and QuReTeC are provided in Vakulenko et al. (2021b).

B Additional Case Study

We provide two additional cases in Table 8 for analysis. The first one is a successful case where the generated expansion terms “motor”, “object”, “kinetic”, and “potential energy” occur in the relevant passage. Thus, they can further boost the retrieval performance although the model has already rewritten the query as the human-rewritten one. The second one is a failure case where the generated answer cannot act as useful expansion terms and even hurt the retrieval results. The possible reason is that the PLM generated a redundant answer and there are no co-occurring and semantic related terms contained in the relevant passage. Thus, the expansion terms are harmful. This is a case that we should improve in the future.

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

Table 7: Statistics of conversational search datasets.

Successful Case	Failure Case
Context: (QReCC Session 17) q_1: What are the different forms of energy? r_1: Examples of these are: light energy, heat energy, mechanical energy, gravitational energy, electrical energy, sound energy, chemical energy, nuclear or atomic energy and so on. q_2: How can it be stored? r_2: Batteries, gasoline, natural gas, food, water towers, a wound up alarm clock, a Thermos flask with hot water and even pooh are all stores of energy. They can be transferred into other kinds of energy. q_3: What type of energy is used in motion? r_3: Motion energy – also known as mechanical energy – is the energy stored in moving objects. As the object moves faster, more energy is stored. q_4: Tell me about mechanical energy. r_4: Mechanical energy is the sum of kinetic and potential energy in an object that is used to do work. In other words, it is energy in an object due to its motion or position, or both. **Current Query**: q_5: Give me some examples. **Human-Rewritten**: $q_5{}^*$: Give me some examples of mechanical energy. **ConvGQR Reformulated Query**: $\hat{q_5}$: Give me some examples of mechanical energy. The energy in a motor is the sum of kinetic and potential energy in an object that is used to do work. **Relevant Passage**: p^*: Objects have mechanical energy if they are in motion ... *A few examples are: a moving car possesses mechanical energy due to its motion(kinetic energy) and a barbell ... its vertical position above the ground(potential energy).* **Dense Score**: 0.33 (Human-Rewritten) **1.00 (Ours)** **Sparse Score**: 0.03 (Human-Rewritten) **0.17 (Ours)**	**Context**: (QReCC Session 5) q_1: What are the best ways to cook a turkey? r_1: Heat the oven to 450°F to preheat and then drop the temperature to 350°F when putting the turkey into the oven. The turkey is done when it registers a minimum of 165° in the thickest part of the thigh. q_2: Should I brine a turkey before smoking it? r_2: Use a brine before smoking to help keep meat moist while cooking and to add flavor. **Current Query**: q_3: How much salt do I use to brine it? **Human Oracle Rewrite**: $q_3{}^*$: How much salt do I use to brine a turkey? **ConvGQR Reformulated Query**: $\hat{q_3}$: How much salt do I use to brine a turkey? Salt: 1 teaspoon per pound of turkey breast, 1 teaspoon per pound of ground turkey breast, 1 teaspoon per pound of ground turkey breast, **Relevant Passage**: p^*: How To Brine a Turkey ... Salt Solution *The basic ratio for turkey brine is two cups of kosher salt to two gallons of water. Some recipes include sweeteners or acidic ingredients to balance the saltiness.* **Dense Score**: **1.00** (Human-Rewritten) 0.5 (Ours) **Sparse Score**: 0.13 (Human-Rewritten) 0.00 (Ours)

Table 8: Two additional concrete examples about different effectiveness of expanding generated query. The blue tokens and the orange tokens stand for the rewritten query and the expanded query of ConvGQR. The expansion terms and the gold answer are underline and *italicized* in the relevant passage.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
section 6 (Limitation)

□ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1 (Introduction)

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
No response.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

Section 4 (Experiments)

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4 and appendix

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4 and appendix

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.