RAC: Retrieval-augmented Conversation Dataset for Open-domain Question Answering in Conversational Settings

Bonggeun Choi[1], Jeongjae Park[1], Yoonsung Kim[2], Jae-Hyun Park[2], Youngjoong Ko[1]*
[1]Sungkyunkwan University, Republic of Korea
[2]NCSOFT
{bonggeun.choi818, jeongjaepark97}@gmail.com, yjko@skku.edu
{yoonsungkim, jaehyunpark}@ncsoft.com

Abstract

In recent years, significant advancements in conversational question and answering (CQA) have been driven by the exponential growth of large language models and the integration of retrieval mechanisms that leverage external knowledge to generate accurate and contextually relevant responses. Consequently, the fields of conversational search and retrieval-augmented generation (RAG) have obtained substantial attention for their capacity to address two key challenges: query rewriting within conversational histories for better retrieval performance and generating responses by employing retrieved knowledge. However, both fields are often independently studied, and comprehensive study on entire systems remains underexplored. In this work, we present a novel retrieval-augmented conversation (RAC) dataset and develop a baseline system comprising query rewriting, retrieval, reranking, and response generation stages. Experimental results demonstrate the competitiveness of the system and extensive analyses are conducted to apprehend the impact of retrieval results to response generation.

1 Introduction

Conversational question answering (CQA), also known as interactive or sequential QA, focuses on answering questions within a conversational context (Webb, 2006; Saeidi et al., 2018; Reddy et al., 2019). However, existing studies often constrain questions and answers within predefined contexts, excluding the retrieval process (Reddy et al., 2019; Choi et al., 2018). This limitation creates a gap between the ideal and actual CQA environment. A more realistic scenario is to retrieve relevant passages related to a question each turn of the conversation and use these passages to provide answers. We refer this new task as *Retrieval-Augmented Conversation* (RAC).

The integration of retrieval fundamentally distinguishes RAC from conventional CQA. It is essential to construct proper search queries for retrieving external knowledge. Conversational search plays a pivotal role in addressing this challenge. It involves query reformulation based on understanding of conversational history, resolving coreference or anaphora across multiple turns, and expanding queries with supplementary terms to enhance retrieval performance (Kim et al., 2021; Qian and Dou, 2022; Wu et al., 2022; Mo et al., 2023; Mao et al., 2023). Another significant challenge in RAC lies in utilizing the retrieved knowledge to provide accurate responses. Recent advancements in large language models (LLMs) have led to the widespread use of generative models for open-domain QA tasks. These models, referred to as retrieval-augmented generation (RAG) models, offer superior performance and flexibility (Raffel et al., 2020; Min et al., 2020; Lewis et al., 2020b). Moreover, generative models are well-suited for answering questions in conversational settings. In summary, RAC is a mixture of conversational search and RAG that covers query reformulation, passage retrieval, and response generation. By addressing both retrieval and generation aspects, RAC aims to bridge the gap between the ideal and current CQA environments.

Despite its significance, no dedicated datasets for RAC exist. While Anantha et al. (2021) introduce the QReCC dataset that meets some conditions of RAC: requiring retrieval at each turn, query reformulation based on conversational history, and answering questions using retrieved passages, the gold answers in the dataset commonly consist of extracted sentences or phrases, which do not fully align with human-like responses suitable for conversational settings. To address this limitation, we introduce a new RAC dataset, derived from publicly available *knowledge-retrieval con-*

[*] Corresponding author

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 1477–1488
November 12-16, 2024 ©2024 Association for Computational Linguistics

versation dataset on AI-Hub[1], a prominent Korean data platform. The conversations include multiple utterances between a user question and its expert response. In each turn, supporting factors[2] used for the response are annotated, along with a referred document in the form of a URL. Details for the data construction are specified in Section 2.

Using this comprehensive dataset, we develop a strong baseline system that encompasses query rewriting, retrieval, reranking, and response generation. Query rewriting model is trained to rewrite queries from a current question with its conversation history. Furthermore, we train the model on the passage collection to enhance the ability of generating relevant terms inspired by the recent generative retrieval paradigm (Li et al., 2023, 2024). For passage retrieval, we adopt BM25 retriever due to its competitive performances, already demonstrated in other studies (Wu et al., 2022; Mo et al., 2023). Rather than excessively refining the retriever, we focus on reranking the retrieved passages. These passages are reranked based on the average probabilities that the query rewriting model generates the query used for retrieving them. Finally, following Fusion-in-Decoder (FiD) (Izacard and Grave, 2021), responses are generated using top-k retrieved passages that are fed into the encoder one-by-one and their last hidden states are concatenated to form the encoder hidden states for the decoder.

Experimental results demonstrate that training the query rewriting model on the entire passage collection and optimizing the reranking stages lead to remarkable performance improvements. In summary, our contributions are as follows:

- We introduce a novel RAC dataset bridging the gap between existing CQA and ideal RAC we aim to achieve, covering up the retrieval and generation aspects.
- Our RAC system establishes a robust baseline. In particular, the proposed learning method for query rewriting model and reranking approach enhance performance significantly.
- We conduct an empirical analysis on the baseline system, shedding light on the challenges faced by the entire RAC system.

2 Data Construction

To comprehensively address the requirement of RAC, a dataset must comprise conversations with referenced passages for response generation, as well as passage collections for retrieval purposes. However, existing CQA datasets are insufficient for the entire RAC because they provide questions and answers constrained on given contexts or do not cover an answering stage. Neither conversational search nor RAG datasets are also inadequate, as they primarily focus on query rewriting to improve retrieval performance and response generation using retrieved knowledge, respectively. To bridge the gap, we utilize the *knowledge-retrieval conversation* dataset and address its limitations. The dataset contains conversations between a user and an expert on several topics, including supporting factors configuring the responses by the expert and documents referenced for the supporting factors.

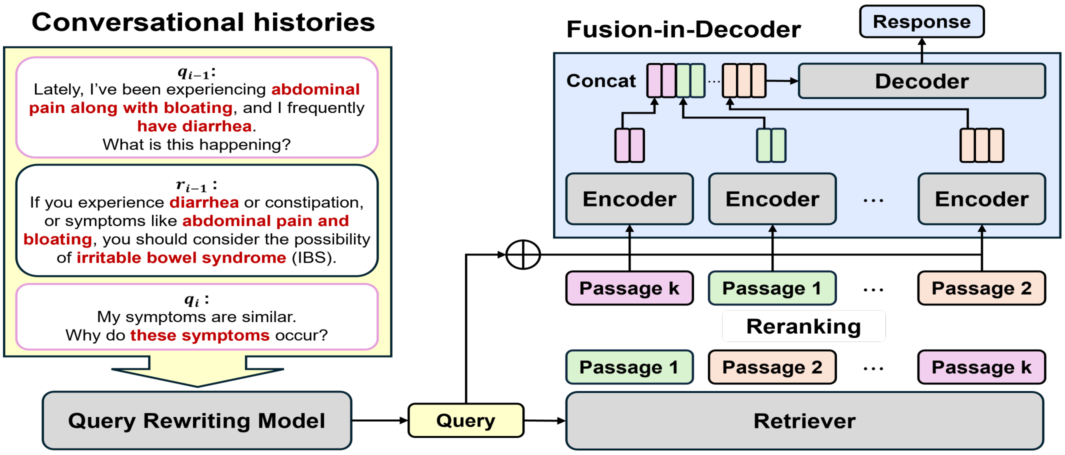
Passage collections The original dataset only provides document URLs that are referenced and hence it does not support retrieval stage. Therefore, we crawled whole Korean wikipedia pages and publicly opened news data over 20 years to reflect various eras. About 1M news were randomly selected from the overall news data and then the crawled data are chunked into passages of fixed length. It is worth to note that retrieval was not performed for the crawling because it may lead to a biased passage collections. Finally, a total of 1,345,209 passages were collected for incorporating the retrieval process.

Human-written query As colloquial questions often do not suit for retrieval purposes, proper queries are needed to deal with the query rewriting aspect of the RAC. To construct queries, we utilize questions and their conversational histories, excluding responses corresponding to the current questions because responses may contain key terms that simplify retrieval stage. For example, consider Figure 1, where the term *"irritable bowel syndrome"* in the response is difficult to be derived from the initial question, but it can be used to rewrite a query from the second question by utilizing the history. Likewise, relevant passages were not provided to prevent excessive paraphrasing. Eventually, 10,266 queries were written by human annotators.

Relevant passage annotation In real-world scenarios, multiple relevant passages may exist for a single input query, whereas the original dataset

[1] https://www.aihub.or.kr

[2] Note that the supporting factors are provided from the original dataset but we do not utilize them for developing baseline system.



q_{i-1}:
Lately, I've been experiencing **abdominal pain along with bloating**, and I frequently **have diarrhea**.
What is this happening?

r_{i-1} :
If you experience **diarrhea** or constipation, or symptoms like **abdominal pain and bloating**, you should consider the possibility of **irritable bowel syndrome** (IBS).

q_i :
My symptoms are similar.
Why do **these symptoms** occur?

Figure 1: The overview of the baseline system, consisting of query rewriting, retrieval, reranking, and response generation. The baseline system is constructed as pipeline so that each model is trained separately.

only offers one passage per a question. To address this discrepancy, we first retrieved passages using Elastic Search (elasticsearch, 2015) with the *human-written* queries and labeled the top-5 retrieved passages based on their relevance to the question. This process resulted in the annotation of 17,606 additional relevant passages.

Finally, the constructed dataset is available on our Github site[3].

3 Retrieval-Augmented Conversation

Although some studies about RAG construct end-to-end training systems that backpropagates loss of response generation to the retrieval model, we break the entire process into pipelines to alleviate the difficulties of systems as a beginning of RAC. Consequently, the overall system is divided up to query rewriting, retrieval, reranking, and response generation stages.

The encoder-decoder model, such as T5 (Raffel et al., 2020) or BART (Lewis et al., 2020a), was adopted as a backbone for both query rewriting and response generation because the architecture is particularly beneficial for the subtasks by cross-attending to given inputs after the self-attention layer and ensuring that input contexts well affect to the generated tokens. The details of the baseline system is specifically explained in the following subsections.

3.1 Training Method for Query Rewriting

The goal of query rewriting is to transform a question into a query for improving retrieval performance, as the original form of the question is often not suitable for retrieval purposes. It is crucial to make the query contain named entities or relevant nouns contained in a relevant passage as well as resolve coreference or anaphora in the question based on its conversational history.

Motivated by recent studies on generative retrieval (Li et al., 2023, 2024), we develop the query rewriting model utilizing a generative pretrained language model (PLM). The query rewriting model is trained to generate queries not only from questions but also from passages, enabling the model to memorize relevant passage-query pairs and hence implicitly incorporate pertinent terms when rewriting queries. Given that relevant passages accompanied by queries are a small subset of the overall passage collection, it is essential to assign queries to the remaining passages. Therefore, we structure the training process into multi-stages. Initially, the model is trained on relevant passages annotated in training data, along with questions and their conversational histories, to generate the *human-written* queries. Subsequently, pseudo queries are generated for the entire passage collection and used as targets in the following training stages. To elaborate the model, the generated pseudo queries are updated after the end of each training stage, except for the initial training queries. The proposed training method also brings a data augmentation effect, ensuring the model to progressively improve its query rewriting capabilities.

Specifically, when a question is used as an input, the current question and a previous conversational history are separated by a separation token '</s>', and questions and responses in the history are separated by a newline token '\n' as described in Ap-

[3] https://github.com/NLPlab-skku/rac

pendix B. Then, let q_i and $C^k = \{q_k,\ r_k\}_{k=0}^{i-1}$ be the current question and previous conversational history, the model is trained by the typical teacher forcing learning:

$$\mathcal{L} = -\sum_{t=1}^{T} \log\left(\Pr(w_t|\ w_{1:t-1},\ C^k,\ q_i)\right), \quad (1)$$

where w and T refer to a token and length of target query, respectively. Likewise, a passage can be used for an input context by replacing C^k. It is noteworthy that the trained query rewriting model is also used for reranking stage to leverage learned knowledge of relevant query-passage pairs.

3.2 Reranking with Query Rewriting Model

Although cross-encoder models usually have been employed for the reranking stage, training a cross-encoder retriever is cost-ineffective because they require sophisticated training setups, including carefully selected hard negative samples and extended training times compared to dense retrievers. To address these challenges, we utilize the query rewriting model for reranking the retrieved passages.

The query rewriting model, trained on the entire passages to generate queries, implicitly memorizes relevant query and passage pairs. Accordingly, the probability of the model generating a query from a relevant passage would be higher than from other passages, and thereby we can leverage this ability of the model for reranking stage. To infer a new score, s_i, of a passage retrieved by an input query, the passage is passed through the query rewriting model, which outputs probability distributions of query length over the vocabulary. The probabilities of tokens corresponding to the query are then averaged:

$$s_i = \frac{1}{T}\sum_{t=1}^{T} \Pr(w_t|\ w_{1:t-1},\ p_i), \quad (2)$$

where p represents the input passage. Finally, the retrieved passages are reranked based on the computed scores.

3.3 Response Generation

Given that the rank of relevant passages within retrieval results remains unknown, it is advisable to utilize multiple top-ranked retrieved passages.

Fusion-in-Decoder Attempting to encode all retrieved passages together may pose challenges resulting in obscure representations, as the model would attend to tokens from both relevant and irrelevant passages indiscriminately. To address this, we employ the FiD architecture, which independently encodes each passage. The input sequences for the encoder are constructed by concatenating the each retrieved passage with a question. Subsequently, all representations from the encoder are concatenated and passed to the decoder for cross-attention. The decoder is then trained by selectively attending to the representations necessary for generating accurate responses.

Large Language Model Although the FiD model can handle that a question is simultaneously attended to relevant and irrelevant passages through its unique architecture, recent LLMs have shown non-trivial performances in natural language processing fields. Therefore, we also generate responses using a LLM, GPT-4o-mini. The input prompts are as follows:

```
<s> Question: $q_0$
Passage 1: $p_0$
.
.
.
Passage k: $p_k$
Response: </s>
```

3.4 Retrieval Models

We evaluate the generated queries on the traditional BM25 using Pyserini (Yang et al., 2017). The hyperparameters are set to default, which are $k1 = 0.82$ and $b = 0.68$. In addition, a dense retriever is also employed for the comparison. Since there is no publicly available Korean dense retriever, we newly pretrain an encoder using a shallow decoder following prior studies (Shen et al., 2023; Zhang et al., 2023; Liu et al., 2023; Wang et al., 2023) and fine-tune the dense retriever with the contrastive learning (Karpukhin et al., 2020). The specific pretraining method for the encoder is described in Appendix C.

4 Experiments

4.1 Dataset

We preprocessed the original dataset to align with the RAC environment, including query rewriting, retrieval, and response generation. To this end, we excluded turns where retrieval was unnecessary and where relevant passages were either nonexistent or modified. In addition, since the original dataset

Splits	train	dev	test
# conversations	770	194	239
# turns	5,550	1,403	1,727
# relevant passages	3.47	3.47	3.49

Table 1: Statistics of the preprocessed dataset.

was divided into training and validation set, we merged the whole data and randomly split them into training, validation, and test sets. Finally, a total of 1,203 conversations were divided into training, validation, and test sets. Each conversation comprises approximately 10 turns, with an average of 3.47 relevant passages per turn. It is important to note that each turn retains its previous history and the excluded turns are also contained in the history to maintain the conversational context. The statistics of the preprocessed dataset are summarized in Table 1.

4.2 Implementation Details

We implemented the encoder-encoder model in PyTorch (Paszke et al., 2019) using a pre-trained Kobart-base-v2 initialization from the huggingface (Wolf et al., 2020) both for query rewriting and response generation. The details of selected hyperparameters are specified in Appendix D.

4.3 Main Results

Passage Retrieval The proposed reranking strategy significantly improved the first-stage retrieval results from the dense and BM25 retriever, reported in Table 2. As a result, more than half of the queries retrieved relevant passages within the top five results. Given that the query rewriting model trained to generate (pseudo) queries from passages is familiar to relevant query and passage pairs, the probability that a query generated from a relevant passage become higher than that generated from irrelevant passages.

The performance of BM25 generally exceeded that of the dense retriever. This can be attributed to the nature of *human-written* queries, which are constructed using a small number of terms derived from previous conversational histories or current questions. As a result, the model trained to generate such queries outputs that are well-suited to the BM25 retriever, which relies on the overlap of terms between a query and a passage. In contrast, the dense retriever, which is designed to capture the semantics of inputs, struggles to effectively capture context from those brief terms.

Following the competitive query reformulation method Mo et al. (2023), we additionally trained response generation model that uses only a user question (not a query) as an input without passage retrieval. Then, generated responses were used for expanding queries to enhance the semantics of input queries for the dense retriever. With the expanded queries, the retrieval performance of the dense retriever is significantly improved as shown in Table 3. However, the performance is still lower than that of BM25 (i.e., first-stage retrieval). The result demonstrates that dense retrievers do not always guarantee superior performances compared to BM25 in line with the retrieval results on other CQA datasets, such as QReCC (Anantha et al., 2021).

Response Generation with FiD We generated responses with diverse retrieval results to understand the correlation between retrieval and response generation. Although the retrieval performance of the dense retriever and BM25 exhibited some differences, the final responses generated using the retrieved passages were almost identical, as shown in Table 4. Moreover, responses generated from passages retrieved by the dense retriever scored higher than those generated using BM25 results, despite BM25's higher retrieval performance. Specifically, response generation performance increased in line with significant improvements in retrieval performance. However, there was no significant difference in response generation performance for similar levels of retrieval results. For instance, the overall results, *i.e.,* retireval and response generation, can be categorized into two groups: the results from first-stage retrieval and those from the reranked ones. These groups achieved similar intra-scores within the groups but showed different inter-scores between the groups. This indicates that minor differences in similar retrieval results can be attributed to fluctuated ranks of top-retrieved passages.

Response Generation with LLM We built the baseline system as a pipeline by separating the overall process into several subtasks: query rewriting, first-stage retrieval, reranking, and response generation. Actually, reranking stage is not a mandatory stage among the subtasks, but it is important to get passages more relevant to questions. Although modern LLMs may well generate human-like responses compared to fully fine-tuned model (i.e., FiD), the quality of the responses can be increased with respect to the quality of retrieved passages.

Retriever	Stages	Retrieval Metrics				
		MRR	Recall@5	MAP@5	NDCG@5	Hit@5
Dense	First-stage ret.	0.272	0.213	0.143	0.192	0.382
	+Reranking	0.439	0.393	0.293	0.359	0.575
BM25	First-stage ret.	0.332	0.272	0.192	0.249	0.460
	+Reranking	**0.453**	**0.414**	**0.310**	**0.377**	**0.595**
	HUMAN WRITTEN	0.512	0.436	0.322	0.404	0.681

Table 2: Retrieval results both on dense and sparse (BM25) retriever. The higher value indicates the better performance in all metrics. Since we train and evaluate the response generation model with top-5 retrieved passages, the metrics are also calculated with 5 passages ranked at top.

Query	Retrieval Metrics			
	MRR	R@5	MAP@5	NDCG@5
Rewritten	0.272	0.213	0.143	0.192
+expansion	**0.319**	**0.259**	**0.155**	**0.231**

Table 3: First-stage retrieval results of the dense retriever using the rewritten queries and expanded queries as inputs.

Retriever	Stages	Response Generation Metrics		
		ROUGE-L	BLEU	METEOR
Dense	First-stage ret.	0.076	0.054	0.221
	+Reranking	0.101	**0.066**	**0.244**
BM25	First-stage ret.	0.083	0.059	0.228
	+Reranking	**0.102**	0.065	0.241
Relevant-only		0.194	0.127	0.335

Table 4: Performances of the response generation with the FiD model across the retrieval results. We also generated responses with only relevant passages from the original dataset that provides one passage per question.

Generator	Ret. Stage	Response Generation Metrics		
		ROUGE-L	BLEU	METEOR
FiD	First-stage ret.	0.083	0.059	0.228
	+Reranking	0.102	**0.065**	0.241
LLM	First-stage ret.	0.134	0.056	0.309
	+Reranking	**0.154**	0.062	**0.324**

Table 5: Performance compariosn between the FiD model and LLM for response generation. The input passages are retrieved by BM25.

Stages	Retrieval Metrics			
	MRR	R@5	MAP@5	NDCG@5
First-stage ret.	**0.332**	**0.272**	**0.192**	**0.249**
-passage learning	0.310	0.265	0.186	0.239

Table 6: Comparison of the first-stage retrieval results using BM25 according to whether the query rewriting model learns passages or not.

Table 5 compares the response generation performances between the FiD model and LLM (i.e., GPT-4o-mini). As expected, the LLM generally performs better than the FiD model. Nevertheless, what we want to emphasize is that both models benefited from precisely reranked passages, stressing the importance of retrieval quality again.

4.4 Learning Passages for Query Rewriting

In Table 6, the retrieval results are reported from which queries are generated the query rewriting model trained with passages and that without passages. When the model learned questions only, without the passages, the performance declined 0.022%p in terms of Mean Reciprocal Rank (MRR). This degradation of performance demonstrates that the model, trained on passages to generate queries following the generative retrieval paradigm, is enhanced to effectively memorize the passages and implicitly generate terms contained in relevant passages, thereby aiding the term-based retriever.

4.5 Analysis on Generated Responses

Effect of the Number of Relevant Passages for Response Generation Given the uncertainty about the existence of relevant passages in the retrieval results, it is reasonable to utilize several passages ranked at the top. Consequently, the number of retrieved relevant passages may influence response generation. Figure 2 illustrates performance changes on two metrics of the generated responses both the FiD model and the LLM relative to the number of relevant passages among the retrieved ones. Generally, as the number of relevant passages in the retrieval results increased, performance steadily improved. However, the ROUGE-L score significantly dropped when all the retrieved passages were relevant. This occurred because the cases take a very small portion of the overall cases and the generated responses were typically shorter than the gold ones affecting to calculation of the

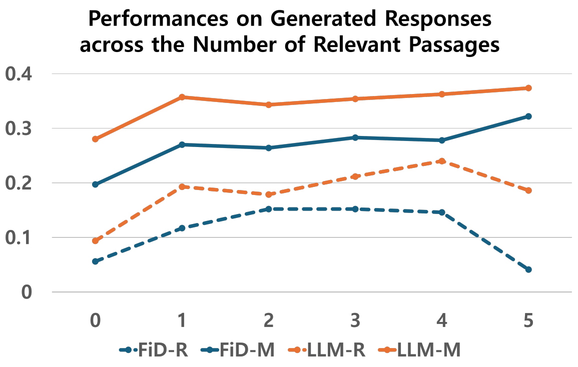


Figure 2: Performances on generated responses according to the number of retrieved relevant passages. -R and -M denote ROUGE-L and METEOR, respectively. The x-axis indicates the number of retrieved relevant passages.

metrics, leading to a sudden drop in the ROUGE-L metric both for the FiD model and LLM.

Furthermore, since performance does not show significant differences across the number of relevant passages except in cases where there are no relevant passages or all passages are relevant, it is crucial for the RAC system to retrieve passages that actually leverages response generation rather than to retrieve as many relevant passages, proved by response generation results using relevant-only passages in Table 4. To achieve this, integrating retriever and response generation models into an end-to-end system could be effective and it will be the future direction of our study.

Human Evaluation We conducted human evaluation on the responses generated by the FiD model based on four criteria: relevance to the question, partial relevance to the question, partial irrelevance to the question, and irrelevance to the question. The guidelines for the metrics are as follows:

- **Relevant to Question**: The response directly addresses the question, providing relevant information or a clear response.
- **Partially Relevant to Question**: The response contains some relevant information but may not fully answer the question or may include extraneous details.
- **Partially Irrelevant to Question**: The response contains somewhat relevant information but the core content is irrelevant or wrong to the question.
- **Irrelevant to Question**: The response does not address the question, providing irrelevant or off-topic information.

# Relevant	Human Evaluation			
	Rel.	**Partial rel.**	**Partial irrel.**	**Irrel.**
0	16.2%	**30.7%**	27.6%	**25.5%**
1	22.2%	26.7%	36.7%	14.4%
2	30.4%	22.4%	35.1%	12.1%
3	**34.6%**	22.6%	32.1%	10.7%
4	19.3%	29.8%	**45.6%**	5.3%
5	33.3%	8.3%	41.7%	16.7%
Total	22.3%	27.2%	32.7%	17.8%

Table 7: Human evaluation on generated responses. Each value represent the portion of the evaluated data out of the case.

Consistent with the analysis of the correlation between the numbers of retrieved relevant passages, the human evaluation discovered that the model does not always provide relevant responses, even when all retrieved passages were pertinent to the given questions. Furthermore, the responses exhibited the highest percentage of irrelevance. This typically occurred when past information appeared across all retrieved passages, leading to incorrect responses. Thus, it can be concluded that the generation model is weak for temporal questions, necessitating more sophisticated strategies to address time-dependent questions.

Another interesting observation is that the model provided (partially) relevant responses even when it did not use relevant passages in nearly half of the cases. Upon closer examination, it was noted that there are many scenarios where diverse responses are possible to questions. These types of responses are not well-addressed by existing evaluation metrics, indicating a need to develop better methods for evaluating generated responses.

5 Conclusion

In this work, we introduced RAC and presented the new dataset that satisfies its requirements. With the comprehensive dataset, a strong baseline system comprising query rewriting, retrieval, reranking, and response generation was constructed. Specifically, the query rewriting model was trained following the generative retrieval approach and also used for reranking stage by leveraging the ability of query generation from passages, resulting in significant improvement of the retrieval performance. Our empirical experiments and analyses discover the challenges of RAC and enlighten the future direction of the entire system.

Limitations

In this work, we utilized a small encoder-decoder model for query rewriting, which are weak in terms of parameterization compared to LLMs. As recent progress in natural language processing is largely contributed by LLMs, it would be interesting to employ larger and decoder-only models to get more effective queries.

In addition, the proposed dataset was constructed in Korean so that language specific features might influence the results. For language-agnostic generalization of the RAC, experiments on diverse languages are required. Hence, we are going to translate the dataset into English and publicly open it after verification to facilitate studies on RAC.

Acknowledgments

This research was supported by NCSOFT. This work was partly supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (RS-2019-II190421, AI Graduate School Support Program(Sungkyunkwan University)), Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT)(RS-2024-00343989, Enhancing the Ethics of Data Characteristics and Generation AI Models for Social and Ethical Learning), and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2024-00350379).

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 *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 520–534, Online. Association for Computational Linguistics.

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, Brussels, Belgium. Association for Computational Linguistics.

elasticsearch. 2015. elasticsearch/elasticsearch.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Gangwoo Kim, Hyunjae Kim, Jungsoo Park, and Jaewoo Kang. 2021. Learn to resolve conversational dependency: A consistency training framework for conversational 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 6130–6141, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2023. Multiview identifiers enhanced generative retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6636–6648, Toronto, Canada. Association for Computational Linguistics.

Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2024. Learning to rank in generative retrieval. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(8):8716–8723.

Zheng Liu, Shitao Xiao, Yingxia Shao, and Zhao Cao. 2023. RetroMAE-2: Duplex masked auto-encoder for pre-training retrieval-oriented language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2635–2648, Toronto, Canada. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Kelong Mao, Zhicheng Dou, Fengran Mo, Jiewen Hou, Haonan Chen, and Hongjin Qian. 2023. Large language models know your contextual search intent: A prompting framework for conversational search. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1211–1225, Singapore. Association for Computational Linguistics.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

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

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. *Preprint*, arXiv:2403.12968.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward 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*, volume 32. Curran Associates, Inc.

Hongjin Qian and Zhicheng Dou. 2022. Explicit query rewriting for conversational dense retrieval. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4725–4737, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Marzieh Saeidi, Max Bartolo, Patrick Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. 2018. Interpretation of natural language rules in conversational machine reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2087–2097, Brussels, Belgium. Association for Computational Linguistics.

Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Xiaolong Huang, Binxing Jiao, Linjun Yang, and Daxin Jiang. 2023. LexMAE: Lexicon-bottlenecked pretraining for large-scale retrieval. In *The Eleventh International Conference on Learning Representations*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2023. SimLM: Pre-training with representation bottleneck for dense passage retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2244–2258, Toronto, Canada. Association for Computational Linguistics.

Nick Webb, editor. 2006. *Proceedings of the Interactive Question Answering Workshop at HLT-NAACL 2006*. Association for Computational Linguistics, New York, NY, USA.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zeqiu Wu, Yi Luan, Hannah Rashkin, David Reitter, Hannaneh Hajishirzi, Mari Ostendorf, and Gaurav Singh Tomar. 2022. CONQRR: Conversational query rewriting for retrieval with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10000–10014, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 1253–1256, New York, NY, USA. Association for Computing Machinery.

Kai Zhang, Chongyang Tao, Tao Shen, Can Xu, Xiubo Geng, Binxing Jiao, and Daxin Jiang. 2023. Led: Lexicon-enlightened dense retriever for large-scale retrieval. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 3203–3213, New York, NY, USA. Association for Computing Machinery.

A Related Work

A.1 Conversational QA

CoQA is a dataset designed for building conversational question answering systems, containing 127k questions from 8k conversations across seven domains. The dataset emphasizes conversational questions and free-form text answers with highlighted evidence in the passages. The study shows that conversational questions pose unique challenges such as coreference and pragmatic reasoning, which are not present in traditional reading comprehension datasets. Evaluations reveal that current models significantly lag behind human performance, indicating substantial room for improvement. CoQA aims to stimulate advancements in conversational question answering (Reddy et al., 2019).

Anantha et al. (2021) presented a dataset for Question Rewriting in Conversational Context (QReCC), containing 14,000 conversations with 80,000 question-answer pairs. This is the first approach to incorporate information retrieval and reading comprehension as subtasks to answer the question within conversational histories. A strong baseline approach combining state-of-the-art models for question rewriting and competitive open-domain QA model is proposed. Nevertheless, there is a still limitation that the dataset does not provide rationale for answering questions which make it harder to analyze intermediate stages.

A.2 Conversational Search

Kim et al. (2021) addresses the challenge of resolving dependencies in conversational question answering (CQA). It introduces a consistency training framework to enhance model performance by ensuring that the model's answers remain consistent throughout a conversation. They introduced a novel training framework that leverages consistency training to handle conversational dependencies. Maintaining answer consistency across conversation turns results in improved performance on existing CQA datsets.

Qian and Dou (2022) presents a model called CRDR designed to handle query rewriting and context modeling within a unified framework for conversational search scenarios. The CRDR modifies only the necessary parts of the original query, enhancing both the accuracy and efficiency of query rewriting. This explicit rewriting helps highlight relevant terms, improving the contextualized query embedding.

Wu et al. (2022) focuses on improving conversational passage retrieval by rewriting queries using reinforcement learning. A query rewriting model (ConQRR) is optimized for passage retrieval performance rather than just human readability. Their experiments demonstrates that human-rewritten queries are precisely clear, but may omit context useful for retrieval, affecting performance. The proposed model significantly enhances retrieval effectiveness by aligning the query rewriting process with the retrieval task's requirements.

Mo et al. (2023) explores generative query reformulation to improve conversational search. A dual approach combining query rewriting and query expansion to address ambiguous queries and supplement them with additional context were proposed. The ConvGQR model integrates both rewriting and expansion techniques to produce more effective search queries. Emipirical results show that the combined approach outperforms traditional methods in generating queries that lead to better retrieval performance.

Mao et al. (2023) introduces LLM4CS, a framework leveraging large language models (LLMs) to interpret users' contextual search intent in conversational search scenarios. By generating multiple query rewrites and hypothetical responses, the framework creates an integrated representation of the user's search intent. Evaluations on conversational search benchmarks demonstrate the framework's effectiveness and robustness, outperforming existing methods and even human rewrites in some cases. The study underscores the potential of LLMs in enhancing conversational search systems.

A.3 Retrieval-augmented Generation

Lewis et al. (2020b) first introduced the word retrieval-augmented generation for knowledge-intensive NLP tasks. The paper introduces a RAG approach that combines retrieval mechanisms with generative models to handle knowledge-intensive NLP tasks. By incorporating retrieved information from knowledge bases, the model can generate more accurate and informed responses for tasks like question answering.

Izacard and Grave (2021) explores enhancing generative models for open-domain QA by incorporating passage retrieval, proposing Fusion-in-Decoder (FiD) architecture. Generative models have shown promise without external knowledge but require large parameters, making them costly. The authors investigate how these models can benefit from retrieving relevant text passages. The approach achieves state-of-the-art results on benchmarks like Natural Questions (NQ) and TriviaQA, showing significant performance improvement with more retrieved passages.

Pan et al. (2024) presents LLMLingua-2, a method for task-agnostic prompt compression to improve generalizability and efficiency in LLMs. Traditional prompt compression methods rely on information entropy, which may be suboptimal. LLMLingua-2 uses data distillation from an LLM and formulates prompt compression as a token classification problem to maintain the integrity of the original prompt. The approach employs a Transformer encoder to capture essential information using bidirectional context. The model shows significant performance gains and robust generalization across various datasets, achieving faster compression and reduced latency compared to existing methods.

REALM (Retrieval-Augmented Language Model) integrates a knowledge retriever with language model pre-training. This approach allows the model to retrieve and use external knowledge during both pre-training and fine-tuning. REALM significantly improves performance on open-domain question answering benchmarks by providing interpretability and modularity, outperforming state-of-the-art models by a large margin (Guu et al., 2020).

B Input Format of the Query Rewriting Model

When input is a question with previous histories, the input form for the model is as follows:

<s> History:
Question: q_0
Response: r_0
.
.
.
Question: q_{i-1}
Response: r_{i-1} </s>
Input: q_i </s>

C Training Dense Retriever

Although DPR demonstrates promising performance, pretraining the encoder enhances the model to be more advanced. Typically, the model is pretrained to improve the vector representation of input passages by employing a shallow decoder. In line with previous studies, we also incorporate a shallow decoder with an encoder exclusively for pretraining purposes.

Input tokens are separately constructed for the encoder and decoder by replacing some tokens in a passage with a mask token for language modeling. The ratio of masking remains consistent, but the positions where tokens are replaced differ between the modules. It is notable that the same token can be masked for both the encoder and decoder, as illustrated below:

$$x_e = \texttt{[CLS]}\ t_0\ \texttt{[MASK]}\ t_2\ \texttt{[MASK]}, \dots\ \texttt{[SEP]}, \quad (3)$$

$$x_d = \texttt{[CLS]}\ \texttt{[MASK]}\ t_1\ t_2\ \texttt{[MASK]}, \dots\ \texttt{[SEP]}, \quad (4)$$

where t denotes the tokens in the passage. The encoder and decoder are then trained to reconstruct the original tokens at the masked positions. Specifically, the last hidden state of the first token, [CLS], from the encoder is fed into the decoder, aligned with the word embeddings of other tokens. Consequently, the language modeling loss from the decoder is backpropagated to the encoder through the encoder's [CLS] hidden state. This process enhances the vector representation used for calculating vector similarity in the dense retriever, as it effectively memorizes input context to aid the decoder in reconstructing the original input.

After pretraining, the encoder is fine-tuned by following typical dense retrievers that maximize vector similarities between queries and their relevant passages through the contrastive learning:

$$\mathcal{L}_i^{Ret} = -\log \frac{e^{\mathrm{f}(q_i, p_i^+)/\tau}}{e^{\mathrm{f}(q_i, p_i^+)/\tau} + \sum_{j=1}^{B} e^{\mathrm{f}(q_i, p_{i,j}^-)/\tau}}, \quad (5)$$

where q_i, p_i^+, and $p_{i,j}^-$ refer to vector representations of query, positive passage, and negative passages, respectively. The score is inferred by the scoring function f that calculates cosine similarity between two vectors after divided by the temperature hyperparameter of τ. The impact of the pretraining is reported in Table 8.

Methods	Metrics				
	MRR	Recall@5	MAP@5	NDCG@5	Hit@5
Non-pretrained	0.242	0.189	0.122	0.167	0.357
Pretrained	**0.272**	**0.213**	**0.143**	**0.192**	**0.382**

Table 8: The retrieval results of pretrained and non-pretrained dense retriever. The pretrained retriever performed better than the non-pretrained one.

D Implementation Details

Query rewriting model For the first stage, the model was trained on questions and passages in training data during 100 epochs. For the remaining training stages, the models was trained during 5 epochs as the entire passages are used, which leaded to a load of training time.

Response generation model To implement FiD model, we slightly modified the code of BART model in the huggingface to make the model encode several passages at the time and then generate a response with the encoded passages. To train the model, it is essential to include multiple passages for the training to meet the test environment where top-k retrieved passages are processed by the encoder. Hence, we used k-1 top-ranked passages retrieved by BM25 using the *human-written* queries in addition to the relevant passage in the training data. To prevent the model learning the order of the input passages, they are fed in to the model in randomly shuffled orders. The model was trained during 20 epochs on the training data.

Both models are trained using AdamW (Loshchilov and Hutter, 2019) with a batch size of 256 and learning rate of 5e-5. Each training took about 3 hours on a RTX A6000 GPUs.