Building Open-Retrieval Conversational Question Answering Systems by Generating Synthetic Data and Decontextualizing User Questions

Anonymous ACL submission

Abstract

We consider open-retrieval conversational question answering (OR-CONVQA), an extension of question answering where system responses need to be (i) aware of dialog history and (ii) grounded in documents (or document fragments) retrieved per question. Domain-specific OR-CONVQA training datasets are crucial for real-world applications, but hard to obtain. We propose a pipeline that capitalizes on the abundance of plain text documents in organizations (e.g., product documentation) to automatically produce realistic OR-CONVQA dialogs with annotations. Similarly to real-world human-annotated OR-CONVQA datasets, we generate in-dialog question-answer pairs, self-contained (decontextualized, e.g., no referring expressions) versions of user questions, and propositions (sentences expressing prominent information from the documents) the system responses are grounded in. We show how the synthetic dialogs can be used to train efficient question rewriters that decontextualize user questions, allowing existing dialog-unaware retrievers to be utilized. The retrieved information and the decontextualized question are then passed on to an LLM that generates the system's response.

1 Introduction

Retrieval-Augmented Generation (RAG) is used to ground large language models (LLMs) to knowledge outside of their training data, and limit hallucinations (Lewis et al., 2020b). RAG is especially applicable to conversational agents, enabling them to provide responses grounded in retrieved documents. We focus on open-retrieval conversational question answering (OR-CONVQA), an extension of question answering where system responses need to be (i) aware of the dialog history and (ii) grounded in the retrieved documents retrieved per question.

Compared to conventional Information Retrieval (IR) (Manning et al., 2008), OR-CONVQA introduces two challenges. Firstly, the system needs to account for the additional context of the dialog (Mao et al., 2022a), mostly the dialog history (previous system and user turns). Solely relying on the last user question to query the document repository may result in sub-optimal answers, since discourse phenomena like ellipsis and co-reference are prevalent in dialogs (Jurafsky and Martin, 2000; Dalton et al., 2022; Zaib et al., 2022; Zamani et al., 2022). Thus the dialog history has to be considered jointly with the last user question, which becomes an issue when the history includes information irrelevant to the last question, or the history is too long. Alternatively, a separate model may produce a self-contained ('decontextualized') version of the last user question (e.g., with no ellipsis, anaphora) (Li and Gaussier, 2024; Yu et al., 2020; Lin et al., 2020), allowing the use of existing dialog-unaware retrievers, which expect a stand-alone query. This approach, *query reformulation*, either rewrites the user question to include all relevant information or appends relevant tokens from the dialog history (Mo et al., 2023a); we adopt the former method.

A second challenge in OR-CONVQA is the lack of domain-specific data and annotations (Mo et al., 2024), which are crucial to train real-life systems. Collecting and, especially, manually annotating new dialog data for specific domains is particularly cumbersome. Alternatives, such as Dialog Inpainting (Dai et al., 2022; Hwang et al., 2023; Wu et al., 2024) or synthesizing dialogs from scratch (Kim et al., 2022), generate synthetic data from domain-specific documents, which are abundant in practice (e.g., product documentation, recommendation guidelines). However, previous alternatives of this kind make unrealistic assumptions, like presuming a one-to-one correspondence between document sentences and possible user questions, and/or assuming that additional manually annotated domain-specific dialogs are available to fine-tune system components (Dai et al., 2022; Hwang et al., 2023).

Motivated by such issues, we propose a pipeline to generate synthetic, document-grounded OR-CONVQA dialogs and annotations. Like previous approaches, the pipeline uses domain-specific documents, but without requiring *any* additional training data and without assuming a one-to-one mapping between document sentences and user questions.

The pipeline firstly prompts an LLM, to generate *propositions* from the documents of the repository. Similarly to Chen et al. (2024), we require each proposition to be a stand-alone simple sentence (e.g., no compound sentences, no anaphora, ellipsis) expressing information from a document. Unlike Chen et al. (2024), however, we require each proposition to convey information important enough to be requested by a user question. Some document sentences may not be used in any of our propositions (hence, they may not be used to answer any question), and some questions may require information from multiple propositions. The retrieval pool may then contain the propositions, not the original documents or document fragments, making it easier to retrieve the information needed by a user question, without irrelevant information.

The pipeline then prompts the same LLM to generate OR-CONVQA dialogs from sampled propositions. Each dialogic pair (user and system turn) includes a contextualized (dependent on dialog history) and decontextualized (self-contained) version of the user question, the corresponding system response (answer), and the propositions used to generate the question and response.

We experimentally show the superiority of dialogs generated through our propositions, compared to using directly document sentences, by measuring the coherence of the dialogs, their relevance to the knowledge they are grounded in, and improvements in retrieval scores. To demonstrate the usefulness of the generated synthetic dialogs, we show how they can be used to fine-tune light models as *question rewriters*. The rewriters generate self-contained (decontextualized) forms of the user questions, which allows utilizing existing (dialog-unaware) retrieval systems. The retrieved information and the self-contained question are then given to an LLM that produces the system's response. We verify the effectiveness and efficiency of our question rewriters on both synthetic and real-world test data, comparing against rewriting questions by prompting larger LLMs, or using the last user query (with or without concatenating the dialog history). We also propose a new mechanism to detect questions that are already self-contained and not require rewriting, improving inference speed further. We leave for future work the question of how to use synthetic data to fine-tune lighter response generation models too, instead of prompting larger LLMs for response generation.

Overall, our main contributions are: (1) We propose a pipeline to create high-quality synthetic annotated OR-CONVQA dialogs from domain-specific documents, without requiring *any* manually annotated training data. (2) We demonstrate the superiority of synthetic dialogs generated by first converting the documents to propositions that capture important information, compared to directly using document sentences. (3) We show how the generated synthetic dialogs can be used to fine-tune light question rewriters, which allow utilizing existing (dialog-unaware) retrievers. (4) We make publicly available our source code and a synthetic OR-CONVQA dataset to facilitate future research.

2 Related Work

2.1 Conversational Question Answering

In the simplest case, Conversational Question Answering (CONVQA) systems answer a *sequence* of questions about a *single* given (always the same) document, by identifying spans of the document that answer each question. The difference from machine-reading comprehension datasets like SQUAD (Rajpurkar et al., 2016) is that the context includes not only the document, but also the previous questions and answers. Choi et al. (2018) and Reddy et al. (2019) concatenate the document with the last k dialog turns, and fine-tune an encoder to predict the document span that answers the last user question. In similar work, Huang et al. (2018), Yeh and Chen (2019), Zhu et al. (2018), Qu et al. (2019), Campos et al. (2020) also use representations from intermediate layers of the encoder.

In OR-CONVQA, the system again needs to take into account the dialog history, but it also needs to retrieve relevant documents for each user question, and compose an answer, typically by feeding the retrieved information to an LLM. For retrieval, one may again concatenate the last k dialog turns, to obtain queries that include the dialog history, and fine-tune a retriever to handle queries of this kind (Qu et al., 2020; Anantha et al., 2021). Fine-tuning the retriever, however, typically requires training data with ground truth, which are difficult to obtain. Thus zero-shot (Krasakis et al., 2022) and approaches with limited supervision (Qu et al., 2021;

Voskarides et al., 2020; Li and Gaussier, 2024; Mao et al., 2022a) have also been proposed.

2.2 Query reformulation

Instead of fine-tuning the retriever to handle queries that include the dialog history, it is computationally cheaper and requires less data (Wu et al., 2022; Zhang et al., 2024) to train a question rewriter to decontextualize (make self-contained) the user questions. This allows utilizing existing dialog-unaware retrievers, which expect stand-alone questions as queries, without fine-tuning them.

Question rewriting is the dominant approach to cope with the dialog history in OR-CONVQA and, more generally, CONVQA, to the point that it is treated as a task of its own (Elgohary et al., 2019). Most question rewriting approaches leverage Transformers (Vaswani et al., 2017) fine-tuned on datasets like those of Anantha et al. (2021), Elgohary et al. (2019), Ren et al. (2021), which include user questions and ground truth rewrites (Li and Gaussier, 2024; Yu et al., 2020; Lin et al., 2020; Vakulenko et al., 2020). Cheng et al. (2024) propose a multitask approach for both retrieval and query rewriting. Mo et al. (2023a) perform both question rewriting and query expansion (§1). Mo et al. (2023b) train their model to identify dialog turns complementary to the last user question.

Query reformulation can also be achieved implicitly. Yu et al. (2021) use BERT (Devlin et al., 2019) to encode the last user question concatenated with the dialog history. They also encode the ground truth query reformulation using the query encoder of an ad-hoc retriever. They fine-tune BERT to minimize the mean squared error loss of the two encodings, in addition to the ranking loss of the BERT encoding of the user question and dialog history. Reinforcement learning has also been leveraged for question rewriting (Wu et al., 2022; Ma et al., 2023). Finally, rewrites can also be generated via prompting LLMs using few or no examples (Mao et al., 2023; Ye et al., 2023; Yoon et al., 2024).

2.3 Synthetic data generation for ConvQA

There is a plethora of manually annotated CONVQA datasets (Elgohary et al., 2019; Anantha et al., 2021; Choi et al., 2018; Qu et al., 2020; Ren et al., 2021; Reddy et al., 2019; Campos et al., 2020; Feng et al., 2020, 2021), but such volumes of annotated data are expensive to compile and scarce in practice when moving to new application domains.

A promising direction to alleviate this issue in OR-CONVQA is to leverage domain-specific documents. In Dialog Inpainting, consecutive sentences of a document are considered an answer to a user question that an LLM tries to generate (Dai et al., 2022; Hwang et al., 2023; Wu et al., 2024). Contrary to our work, this approach assumes that every sequence of sentences is an answer to a possible user question; in practice, however, some document parts may not convey information users would be interested in. In the original Dialog Inpainting, a question generation model also needs to be trained, which requires additional annotated data.

Huang et al. (2023) generate synthetic questions by prompting an LLM. They feed, however, the LLM with ground truth passages (answering user questions) from existing datasets, which are again difficult to obtain in new application domains. They also consider only retrieval, not response generation. Mao et al. (2022b) generate dialog questions from existing web searches. Mo et al. (2024) instruct an LLM to generate dialogs around certain topics, which results in dialogs not grounded in specific documents. In similar work, Bitton et al. (2023) utilize user questions from publicly available QA datasets, instead of topic descriptions.

Closer to our pipeline is the work of Kim et al. (2022) and Liu et al. (2024). The former identifies document fragments that may provide answers to possible user questions, from which synthetic questions and answers are extracted. Contrary to our work, however, their pipeline requires additional annotated data to train their question-answer extractors. Liu et al. (2024) provide a *single* document to an LLM and instruct it to generate a dialog. By contrast, our synthetic dialogs can be grounded on propositions from multiple documents.

3 Methodology

3.1 Domain-specific documents

We hypothesize that our pipeline will be especially beneficial in scenarios revolving around domain-specific documents, rich in knowledge, as is call centers. Hence, we collect 1,036 documents from call centers, henceforth *proprietary documents*, which cover four domains: software, finance, insurance, miscellaneous (misc). We also leverage the 488 publicly available documents of DOC2DIAL (Feng et al., 2020) and MULTIDOC2DIAL (Feng et al., 2021); both datasets use the same documents, hereafter DOC2DIAL or *public documents*, which are similar to the proprietary ones in quantity, ori-

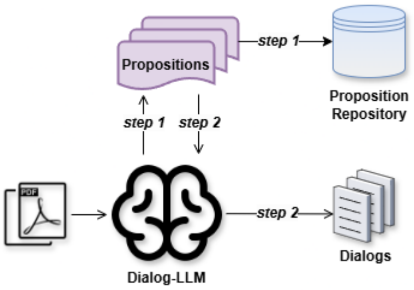

Figure 1: Our synthetic dialog generation pipeline.

gin, and domains. DOC2DIAL documents originate from government websites covering insurance (VA), student financial support (STUDENTAID), car rental (DMV), and social security services (SSA). The DOC2DIAL dataset includes 69,820 dialog turns across 4,470 dialogs, while MULTIDOC2DIAL includes 61,078 turns across 4,796 dialogs, all grounded in the documents provided. All dialogs were created via crowd-sourcing. The main difference between MULTIDOC2DIAL and DOC2DIAL is that the former's dialogs may be grounded in more than one document. We use *only the test dialogs* of both datasets, to measure the performance of our methods on real user questions.

3.2 Synthetic dialog generation pipeline

Step 1: Following Chen et al. (2024), our synthetic dialog generation pipeline (Fig. 1) first prompts an LLM, hereafter 'Dialog-LLM' (Claude 3.5[1] in our experiments), to extract propositions from the documents (§1). The full prompt is provided in Appendix B.1. Specifically, we instruct the Dialog-LLM to split compound sentences into simple ones, separate information into standalone sentences, and decontextualize them to remove references from one proposition to another, taking care to generate propositions only for information users are likely to ask about, unlike the original propositions of Chen et al. (2024). We obtain 11,566 and 14,443 propositions from proprietary and DOC2DIAL documents, respectively. The propositions of all documents are inserted into a single list, keeping propositions from the same document adjacent. We split the list into non-overlapping sublists of size n ($n = 30$ in our experiments), maintaining the original order. Each sublist may, thus, contain propositions from one or more documents.

Step 2 generates synthetic dialogs and annotations (Fig. 1) by prompting the same LLM (Dialog-LLM) with three separate prompts (Appendix B.2).

[1] www.anthropic.com/news/claude-3-5-sonnet

Prompt 2.1 (dialog generation): This prompt instructs the LLM to generate a user-system dialog, grounded in a sampled sublist of n propositions (Step 1). Each sublist is used only once, to generate a single dialog. In this step, we instruct the Dialog-LLM to ensure the user questions are decontextualized, i.e., that they include all the necessary information from the dialog context. We find that generating decontextualized questions first and then contextualizing them (using Prompt 2.2) leads to dialogs where more user turns have been decontextualized correctly, instead of the opposite.
Prompt 2.2 (contextualized questions): The second prompt of Step 2, instructs the Dialog-LLM to create contextualized versions of the user questions, taking into account the dialog context (e.g., inserting pronouns when entities have been mentioned in the dialog history). An example can be seen in Appendix A. Hence, there are two versions of each user question, the contextualized and the decontextualized one, along with the system response.
Prompt 2.3 (ground truth propositions): The third prompt of Step 2 feeds the Dialog-LLM with each sublist of propositions and the corresponding generated dialog, and instructs it to identify the propositions each question-answer pair is grounded on. Thereafter, each pair will contain two versions of the user question (contextualized and decontextualized), the system response, and the corresponding propositions. The Dialog-LLM is also instructed to generate an additional token ('accepted' or 'not_accepted') for each pair, signifying whether the pair is indeed grounded in the selected propositions or not. We remove pairs marked as 'not_accepted' and replace each subsequent user question with its decontextualized version, to avoid referring to a removed pair. This seldom happens.

3.3 Building domain-specific systems

To build an OR-CONVQA system for a new application domain, we first apply our synthetic dialog generation pipeline to the domain-specific documents the user questions will be answered from. This also converts the documents to propositions, stored in the proposition repository (Fig. 1). The synthetic data are also used to fine-tune a light query rewriter to decontextualize user questions. Then, in real-life dialogs, each user question is decontextualized and fed to an off-the-shelf (not fine-tuned) retriever to obtain relevant propositions from the proposition repository. In some of our experiments, we also investigate retrieving document passages. The retrieved information and the decontextualized question are then given to an LLM (Response-LLM) instructed (with the prompt of Appendix B.3) to generate the system response. In our experiments, we use LLAMA-8B (Dubey et al., 2024) as the Response-LLM without fine-tuning it. As already noted, we leave for future work the possibility of fine-tuning a lighter response generator on synthetic data (as we do for question rewriting), instead of prompting a larger Response-LLM.

4 Experiments

4.1 Experimental setup

We experiment with dense retrieval, sparse retrieval, and Reciprocal Rank Fusion (Cormack et al., 2009) (RRF). We always feed the Response-LLM with the top 20 retrieved propositions. For dense retrieval, we use MiniLM (Wang et al., 2020) to embed the propositions of the proposition repository (Fig. 1) and the user questions. For each query we retrieve the top-20 propositions with the highest cosine similarity. For sparse retrieval, we use BM25 (Robertson and Zaragoza, 2009). RRF fuses the scores of the two other retrievers as follows:

$$score_i = \frac{1}{rank_i^b + k} + \frac{1}{rank_i^d + k},$$

where $score_i$ refers to the new score assigned by RRF, i is the index of the propositions regardless of rank, b and d refer to BM25 and dense retrieval, respectively, and k is set to 60 as per usual practice (Cormack et al., 2009). We do not tune k further, nor do we assign weights to the two terms.

For every experiment involving synthetic dialogs, we split them into training and test sets using three different seeds, and report average scores on the test sets. The training sets are only used to train the question rewriter and tune the hyper-parameters of BM25.[2] We also use the original test sets of DOC2DIAL and MULTIDOC2DIAL, unchanged, and conduct the corresponding experiments only once; the training sets of these datasets are not used, since the question rewriter is always trained on synthetic data, to demonstrate that our approach requires no manually annotated training data. For each test set, we report results using the best retrieval method. To measure retrieval performance, we compute Mean Average Precision (MAP), and Recall at the top-k retrieved items (R@k). For response generation, we report SACREBLEU (SBLEU) (Post, 2018) measuring 4-grams, METEOR (Banerjee and Lavie, 2005), BERTSCORE (Zhang et al., 2020), and the perplexity (PL) of the Response-LLM. We also report additional experiments, each one considering a single domain (§3.1), in Appendix D.

4.2 Propositions vs. sentences

We hypothesize that converting the domain-specific documents to propositions leads to synthetic dialogs of higher quality, compared to dialogs generated directly from document sentences. To confirm this, we employ the pipeline of Fig. 1 to generate dialogs with both approaches (propositions, sentences), applying it to the proprietary and public (DOC2DIAL) documents (§3.1). To generate sentence-based dialogs, we split the documents into sentences and form chunks of 30 consecutive ones (maintaining their order), matching the size of the proposition sublists used to generate dialogs in Step 1 (§3.2).[3] From the proprietary documents, we extract 20,520 sentences, which the pipeline uses to generate dialogs; 36% of proposition-generated and 33% of sentence-generated user questions require rewriting. From DOC2DIAL documents, we extract 17,197 sentences; 27% and 28% of user questions require rewriting, respectively.

We compare the quality of proposition-based and sentence-based dialogs, by measuring the relevance of the dialogs to the knowledge they are grounded in, dialog coherence, and retrieval performance.

Relevance: We employ QRELSCORE (Wang et al., 2022) to measure the relevance of each synthetic user question to the corresponding ground-truth propositions (Prompt 2.3) or document chunks, and we then average over user questions. QRELSCORE ranges in $[0, 1]$. It is the harmonic mean of two terms. For the first term, the user question is concatenated with its ground-truth propositions or document chunks, and it is fed to an off-the-shelf BERT. For every layer of BERT, the cosine similarities between each token embedding of the question and each token embedding of the ground truth are calculated and averaged across all layers. The second term measures the difference between the likelihood of an off-the-shelf GPT2 (Radford et al., 2019) generating the context with, and without conditioning on the corresponding question.

Coherence: To measure dialog coherence, we use QUANTIDCE (Ye et al., 2021), which considers the dialogs themselves (not the ground-truth proposi-

[2] In BM25, $k_1 = 0.05$, $b = 5$. The best rewriter checkpoint is selected on development data held out from the training set.

[3] We use NLTK (www.nltk.org/) for sentence splitting.

tions or document chunks). QUANTIDCE employs a BERT model fine-tuned for dialog coherence evaluation on a large dialog corpus. It ranges in $[1, 5]$.

For relevance (QRELSCORE), we consider both contextualized and decontextualized user questions. For dialog coherence (QUANTIDCE), we only consider the contextualized questions, as they better mimic real-world dialogs. Table 1 reports the QRELSCORE and QUANTIDCE scores obtained. When using the proprietary documents, proposition-based dialogs are clearly better than sentence-based ones in relevance (QRELSCORE). When using public (DOC2DIAL) documents, however, both approaches are on par. In dialog coherence (QUANTIDCE), sentence-based dialogs are slightly better, both with proprietary and public documents, but the differences are minute (recall that QUANTIDCE ranges in $[1, 5]$). Overall, we conclude so far (Table 1) that proposition-based dialogs are on par or better than sentence-based dialogs, and we turn to retrieval performance to obtain a clearer winner between the two approaches.

Docs		QRELSCORE↑ (co)	QRELSCORE↑ (de)	QUANTIDCE↑ (co)
PR	Prop	**0.36**	**0.41**	3.16
	Sent	0.25	0.27	**3.18**
PU	Prop	**0.33**	**0.36**	3.08
	Sent	**0.33**	0.35	**3.14**

Table 1: **Relevance** (QRELSCORE) and **Coherence** (QUANTIDCE) results for **synthetic dialogs** generated through **propositions** (Prop) or **sentences** (Sent) using proprietary (PR) and public documents (PU). (co): contextualized questions, (de): decontextualized questions.

Retrieval: We finally compare proposition-based to sentence-based synthetic dialogs by comparing retrieval performance. We use RRF to retrieve either propositions or sentences, and compare three query types: concatenation of the dialog history with the last contextualized user question (Context), contextualized user question alone (Query_{co}), decontextualized user question alone (Query_{de}). We use the previous question-answer pair only as the dialog history, as it led to the best Context results. Note that the decontextualized user questions used here are the 'ground-truth' ones generated by Dialog-LLM (in Step 2). Table 2 shows that proposition-generated dialogs lead to substantially higher retrieval performance, compared to sentence-generated dialogs, which can be attributed to the clearer and more prominent information propositions express. We consider the superior retrieval performance of proposition-generated dialogs as an indication of higher-quality synthetic data, since ground truth decontextualized questions should lead to high retrieval scores. Consequently, we use proposition-based synthetic dialogs in subsequent experiments. Table 2 also shows that concatenating the dialog history with the last user question leads to substantially worse retrieval performance (for off-the-shelf retrievers), probably due to the noise that previous utterances may introduce, as pointed out by Mao et al. (2022a).

PR	Query	MAP↑	R@5↑	R@10↑	R@20↑
Prop	Context	0.19	0.31	0.44	0.56
	Query_{co}	0.46	0.55	0.63	0.77
	Query_{de}	**0.50**	**0.60**	**0.67**	**0.73**
Sent	Context	0.09	0.13	0.20	0.27
	Query_{co}	0.20	0.25	0.30	0.35
	Query_{de}	0.21	0.26	0.31	0.37
PU	**Query**	**MAP↑**	**R@5↑**	**R@10↑**	**R@20↑**
Prop	Context	0.18	0.30	0.44	0.56
	Query_{co}	0.49	0.59	0.65	0.72
	Query_{de}	**0.54**	**0.63**	**0.71**	**0.77**
Sent	Context	0.12	0.19	0.27	0.36
	Query_{co}	0.30	0.37	0.43	0.50
	Query_{de}	0.31	0.39	0.46	0.52

Table 2: **RRF retrieval results** in **synthetic dialogs** generated through **propositions** (Prop) **or sentences** (Sent) using proprietary (PR) and public (DOC2DIAL) documents (PU). Context: concatenated last user question and history, Query_{co}: contextualized question only, Query_{de}: ground-truth decontextualized question only.

4.3 Retrieval in synthetic dialogs

Next, we fine-tune three lightweight rewriters to decontextualize user questions (§3.3): MAMBA 370M (Gu and Dao, 2024), GPT-2 350M, T5 220M (Raffel et al., 2020). To our knowledge, no previous work explores Mamba for query rewriting; we use it, because of its linear complexity and, thus, bet-

	Query	MAP↑	R@5↑	R@10↑	R@20↑
PR	GPT2	0.47	0.56	0.64	0.71
	Mamba	0.48	**0.57**	0.64	0.71
	T5	**0.49**	**0.57**	**0.65**	**0.72**
PU	GPT2	0.51	0.60	0.67	0.74
	Mamba	0.51	0.60	0.67	0.74
	T5	**0.52**	**0.62**	**0.69**	**0.76**

Table 3: **Additional RRF retrieval results** in **synthetic dialogs** generated via **propositions**, using proprietary (PR) and public (PU) documents, and the decontextualized user questions of **lightweight fine-tuned rewriters** as queries. Results comparable to those of Table 2.

ter performance on long sequences, compared to Transformers. We fine-tune the rewriters separately on the training sets of the synthetic dialogs obtained from the proprietary and DOC2DIAL documents. Again, we use RRF retrieval, so the new results (Table 3) are comparable to those of Table 2.

Table 3 shows that lightweight rewriters perform similarly to each other and better than using contextualized user questions, with or without concatenating the dialog context (Context, Query_{co}). Naturally, lightweight rewriters cannot outperform the 'ground truth' decontextualized queries (Query_{de}) generated by Dialog-LLM (produced in Step 2). Additional results with BM25 and dense retrieval are presented in Appendix C. Although smaller in size, T5 has the best performance among the three lightweight rewriters, in dialogues generated from both proprietary and public documents. Hence, we used only the T5 rewriter in subsequent experiments. We also do not experiment further with Context, given its poor results (Tables 2–3).

4.4 Response generation in synthetic dialogs

Continuing our experiments with synthetic dialogs, we now use the contextualized user questions (Query_{co}), the decontextualized user questions of the T5 rewriter, or the 'ground-truth' decontextualized questions (Query_{de}) as queries to the RRF retriever. We then feed the Response-LLM (LLAMA-8B) with the top-20 retrieved propositions and instruct it to generate the system response (prompt in Appendix B.3). Table 4 compares the generated responses to the 'ground truth' system responses (generated by Dialog-LLM in Step 2). The decontextualized questions of T5 clearly lead to better responses, compared to using the contextualized questions, though again the best results are obtained using the 'ground-truth' decontextualized questions, as in Tables 2–3.

	Query	SBLEU↑	METEOR↑	BSC↑	PL↓
PR	Query_{co}	40.57	55.81	93.24	3.78
	T5	42.79	58.67	93.65	3.34
	Query_{de}	**44.52**	**59.72**	**93.99**	**3.25**
PU	Query_{co}	44.92	58.99	93.39	3.03
	T5	47.33	62.11	93.80	2.74
	Query_{de}	**48.73**	**63.46**	**94.08**	**2.68**

Table 4: **Response generation results** in **synthetic dialogs** generated via **propositions** from proprietary (PR) and public (PU) documents, using RRF retrieval.

4.5 Retrieval in real-world dialogs

We now provide evaluation scores in the *real-world* DOC2DIAL and MULTIDOC2DIAL datasets. We do not use any of the training dialogs of these datasets, to demonstrate the value of our synthetic dialog generation pipeline in new application domains *without any* manually annotated dialogs.

We use the T5 question rewriter fine-tuned on the synthetic dialogs we had generated from the DOC2DIAL documents (§4.3). Alternatively, we rewrite user questions by prompting an LLM (CLAUDE-SONNET, also used as Dialog-LLM in the synthetic data generation pipeline). Note that ground-truth question rewrites are not available in the DOC2DIAL and MULTIDOC2DIAL datasets. For retrieval, these datasets provide ground-truth *passages*, thus we also use RRF for passage retrieval.

Table 5 shows that the T5 question rewriter substantially improves the retrieval performance in both datasets, compared to using contextualized questions (Query_{co}). Obtaining question rewrites by prompting an LLM (CLAUDE) leads to further substantial improvements, as with the similar 'ground-truth' rewrites (Query_{de}) of Table 4.2, at the expense of invoking an LLM to rewrite each user question. For reference, Table 5 also includes the reported results of Feng et al. (2020) and Feng et al. (2021), who *fine-tune a dense retriever* (different from our question rewriting approach) on the training sets of the two datasets, thus *requiring manually annotated domain-specific data*. Hence, their results cannot be fairly compared to ours. Since both real-world datasets also include ground-truth documents (not just passages), we also present *document* retrieval results in Table 6. We reach the same conclusions as in passage retrieval.

4.6 Response generation in real-world dialogs

Table 7 shows response generation results, similar to those of Table 4, but now using the real-world dialogs of the test set of MULTIDOC2DIAL. We do not show response generation results for DOC2DIAL, as it concerns generating a system response from a single given document, which is incompatible with our synthetic data generation pipeline and our focus on OR-CONVQA. We now retrieve propositions, since entire documents or passages confuse the Response-LLM (LLAMA-8B) with redundant information. Again, the T5 rewriter (fine-tuned on synthetic data) improves performance, compared to using contextualized questions (Query_{co}). Inter-

D2D/Method	Query	MAP↑	R@5↑	R@10↑	R@20↑
–	Query_{co}	0.17	0.26	0.34	0.40
Rewr-synFT	T5	0.21	0.31	0.41	0.48
Rewr-prompt	Claude	**0.25**	**0.36**	**0.47**	**0.56**
Retr-FT*	Context	n/a	0.85	0.90	n/a
MD2D/Method	**Query**	**MAP↑**	**R@5↑**	**R@10↑**	**R@20↑**
–	Query_{co}	0.17	0.26	0.34	0.40
Rewr-synFT	T5	0.21	0.31	0.40	0.48
Rewr-prompt	Claude	**0.23**	**0.34**	**0.44**	**0.53**
Retr-FT*	Context	n/a	n/a	0.69	0.79

Table 5: **RRF passage retrieval results** in **real-world dialogs** from DOC2DIAL (D2D) and MULTIDOC2DIAL (MD2D). T5/Claude: question rewritten by T5/Claude. Rewr-synFT: rewriter fine-tuned on synthetic data, Rewr-prompt: the rewriter is a prompted LLM, Retr-FT: retriever fine-tuned on *manually annotated domain-specific data* (not comparable to our work). Starred results from Feng et al. (2020) and Feng et al. (2021).

D2D/Method	Query	MAP↑	R@5↑	R@10↑	R@20↑
–	Query_{co}	0.36	0.47	0.56	0.64
Rewr-synFT	T5	0.45	0.59	0.68	0.77
Rewr-prompt	Claude	**0.66**	**0.82**	**0.90**	**0.95**
MD2D/Method	**Query**	**MAP↑**	**R@5↑**	**R@10↑**	**R@20↑**
–	Query_{co}	0.37	0.49	0.57	0.65
Rewr-synFT	T5	0.45	0.58	0.68	0.77
Rewr-prompt	Claude	**0.59**	**0.75**	**0.85**	**0.91**

Table 6: **RRF document retrieval results** in **real-world dialogs** from DOC2DIAL (D2D) and MULTIDOC2DIAL (MD2D). We use the same notation as in Table 5.

estingly, decontextualizing questions by prompting CLAUDE does not necessarily lead to better response generation scores compared to the T5 rewriter. For completeness, we also include the method of Feng et al. (2021), who *fine-tune* BART (Lewis et al., 2020a) for response generation using *manually annotated training data*; hence, their results are not directly comparable. We also note that the ground-truth system responses of MULTIDOC2DIAL are often direct excerpts from the ground-truth documents, whereas the responses of our Response-LLM are more abstractive and are penalized by n-gram based measures (SBLEU, METEOR). By contrast, BERTSCORE, which is based on word embeddings, assigns reasonably high scores to the responses of our Response-LLM.

MD2D/Method	Query	SBLEU↑	METEOR↑	BSC↑	PL↓
–	Query_{co}	6.16	20.93	85.63	26.04
Rewr-synFT	T5	**6.54**	22.65	**85.74**	23.08
Rewr-prompt	Claude	6.52	**23.34**	85.47	**21.75**
Retr-FT*	Context	21.9	n/a	n/a	n/a

Table 7: **Response generation results** in **real-world dialogs** from MULTIDOC2DIAL, using RRF retrieval of **propositions**. Responses generated by LLAMA-8B in our (the first three) methods. Starred results from Feng et al. (2021). We use the same notation as in Table 5.

4.7 Conditional question rewriting

Finally, we propose a new joint question classification/rewriting approach to reduce the expected latency in real-world applications. Again, we use the T5 question rewriter (§4.3), but now during training we prepend each decontextualized question with the tokens 'rewrite', if it is different from the contextualized one, or 'no_rewrite' if not. At inference, we stop the generation procedure if the 'no_rewrite' token is generated, and replace the token with the input (contextualized) question as the prediction of the question rewriter. We find that the performance of the question classifier/rewriter is almost identical to that of the original T5 rewriter (Table 3), with differences noticeable from the third decimal and on, in favor of the original T5 rewriter. More importantly, the average generation time for proprietary dialogs is reduced from 0.19 seconds to 0.09 (53% reduction) and for public dialogs from 0.24 seconds to 0.09 (62% reduction). The reader is reminded (§4.2) that 36% of synthetic proprietary and 27% public user questions require rewriting.

5 Conclusions and Future Work

We presented a new pipeline that generates synthetic annotated document-grounded dialogs, to alleviate the lack of training data in new application domains. The pipeline requires only a set of relevant domain-specific documents. We highlighted the importance of using propositions, rather than document sentences, for dialog generation, and showed experimentally that they lead to synthetic dialogs that are clearly superior in retrieval performance, and on par or superior to dialogues generated from document sentences in coherence and relevance. Using only our synthetic data, we trained light question rewriters, which allow utilizing dialog-unaware retrievers without fine-tuning them. We showed that the rewriters substantially improve performance, compared to using the original questions with or without dialog history, and that their performance is comparable to obtaining rewrites by prompting LLMs. We also introduced a joint efficient question classification/rewriter.

In future work, we plan to use larger sets of documents, enabling us to generate more dialogs, thus facilitating fine-tuning a retriever or a lightweight response generator. Finally, as CONVQA dialogs in low resource languages are scarcer still, we plan to extend our pipeline to such languages, exploiting multilingual LLMs and/or machine translation.

6 Limitations

A limitation of our work is its dependence on LLMs (like CLAUDE-SONNET) for the creation of synthetic data. However costly these models may be, we deem their usage essential to ensure high quality synthetic data, as in previous work (Chen et al., 2024; Mo et al., 2024).

Our generated data, being purely synthetic, may contain errors. For instance, there are a few cases where the 'decontextualized' user question is not fully self-contained. Despite such errors, our fine-tuned question rewriters showed promising results in both synthetic and real-world dialogs. There is also the possibility that the generation pipeline may hallucinate information. For example, there is no guarantee that the generated propositions (Step 1 of the generation pipeline) perfectly reflect the knowledge in the documents; in our experiments, however, we have not encountered any such case.

The number of generated dialogs is limited by to the number and length of the provided documents. For this reason, we opted to train a question rewriter to decontextualize questions, which allows utilizing existing dialog-unaware retrievers, rather than fine-tuning a retriever, which usually requires more training data (Wu et al., 2022).

Finally, although we showed how the generated synthetic data can be used to train lightweight question rewriters, instead of employing costly LLMs for question rewriting, response generation still relies on prompting LLMs. Preliminary experiments (not reported) showed that light response generators (e.g., T5) fine-tuned on our current synthetic dataset severely under-perform compared to prompting LLMs as response generators, possibly because the synthetic datasets are not large enough. We, hence, left this direction for future work.

7 Ethical Considerations

A major concern regarding LLMs like CLAUDE-SONNET, which our generation pipeline leverages, is that sensitive data may be stored by third parties and may even be exposed publicly. In our case, we either used already publicly available documents, or documents that do not include sensitive information and their processing has been approved by qualified individuals. We advise potential users of the pipeline to take similar precautions.

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.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Yonatan Bitton, Shlomi Cohen-Ganor, Ido Hakimi, Yoad Lewenberg, Roee Aharoni, and Enav Weinreb. 2023. q2d: Turning questions into dialogs to teach models how to search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13661–13676, Singapore. Association for Computational Linguistics.

Jon Ander Campos, Arantxa Otegi, Aitor Soroa, Jan Deriu, Mark Cieliebak, and Eneko Agirre. 2020. DoQA - accessing domain-specific FAQs via conversational QA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7302–7314, Online. Association for Computational Linguistics.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2024. Dense X retrieval: What retrieval granularity should we use? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15159–15177, Miami, Florida, USA. Association for Computational Linguistics.

Yiruo Cheng, Kelong Mao, and Zhicheng Dou. 2024. Interpreting conversational dense retrieval by rewriting-enhanced inversion of session embedding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2879–2893, Bangkok, Thailand. Association for Computational Linguistics.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau 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.

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR*

Conference on Research and Development in Information Retrieval, SIGIR '09, page 758–759, New York, NY, USA. Association for Computing Machinery.

Zhuyun Dai, Arun Tejasvi Chaganty, Vincent Y Zhao, Aida Amini, Qazi Mamunur Rashid, Mike Green, and Kelvin Guu. 2022. Dialog inpainting: Turning documents into dialogs. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4558–4586. PMLR.

Jeffrey Dalton, Sophie Fischer, Paul Owoicho, Filip Radlinski, Federico Rossetto, Johanne R. Trippas, and Hamed Zamani. 2022. Conversational information seeking: Theory and application. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 3455–3458, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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

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

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

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*.

Chao-Wei Huang, Chen-Yu Hsu, Tsu-Yuan Hsu, Chen-An Li, and Yun-Nung Chen. 2023. CONVERSER: Few-shot conversational dense retrieval with synthetic data generation. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 381–387, Prague, Czechia. Association for Computational Linguistics.

Hsin-Yuan Huang, Eunsol Choi, and Wen-tau Yih. 2018. Flowqa: Grasping flow in history for conversational machine comprehension. *CoRR*, abs/1810.06683.

Yerin Hwang, Yongil Kim, Hyunkyung Bae, Hwanhee Lee, Jeesoo Bang, and Kyomin Jung. 2023. Dialogizer: Context-aware conversational-QA dataset generation from textual sources. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8806–8828, Singapore. Association for Computational Linguistics.

Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 1st edition. Prentice Hall PTR, USA.

Gangwoo Kim, Sungdong Kim, Kang Min Yoo, and Jaewoo Kang. 2022. Generating information-seeking conversations from unlabeled documents. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2362–2378, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Antonios Minas Krasakis, Andrew Yates, and Evangelos Kanoulas. 2022. Zero-shot query contextualization for conversational search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 1880–1884, New York, NY, USA. Association for Computing Machinery.

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.

Minghan Li and Eric Gaussier. 2024. Domain adaptation for dense retrieval and conversational dense retrieval through self-supervision by meticulous pseudo-relevance labeling. In *Proceedings of the*

2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 5247–5259, Torino, Italia. ELRA and ICCL.

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

Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. 2024. ChatQA: Surpassing GPT-4 on conversational QA and RAG. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

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.

Kelong Mao, Zhicheng Dou, and Hongjin Qian. 2022a. Curriculum contrastive context denoising for few-shot conversational dense retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 176–186, New York, NY, USA. Association for Computing Machinery.

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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Fengran Mo, Kelong Mao, Yutao Zhu, Yihong Wu, Kaiyu Huang, and Jian-Yun Nie. 2023a. 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.

Fengran Mo, Jian-Yun Nie, Kaiyu Huang, Kelong Mao, Yutao Zhu, Peng Li, and Yang Liu. 2023b. Learning to relate to previous turns in conversational search. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 1722–1732, New York, NY, USA. Association for Computing Machinery.

Fengran Mo, Bole Yi, Kelong Mao, Chen Qu, Kaiyu Huang, and Jian-Yun Nie. 2024. Convsdg: Session data generation for conversational search. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, page 1634–1642, New York, NY, USA. Association for Computing Machinery.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Chen Qu, Liu Yang, Cen Chen, W. Bruce Croft, Kalpesh Krishna, and Mohit Iyyer. 2021. Weakly-supervised open-retrieval conversational question answering. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 – April 1, 2021, Proceedings, Part I*, page 529–543, Berlin, Heidelberg. Springer-Verlag.

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. 2020. Open-retrieval conversational question answering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 539–548, New York, NY, USA. Association for Computing Machinery.

Chen Qu, Liu Yang, Minghui Qiu, W. Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019. Bert with history answer embedding for conversational question answering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 1133–1136, New York, NY, USA. Association for Computing Machinery.

Alec Radford, Jeff Wu, Rewon Child, D. Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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. *J. Mach. Learn. Res.*, 21(1).

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

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.

Pengjie Ren, Zhumin Chen, Zhaochun Ren, Evangelos Kanoulas, Christof Monz, and Maarten De Rijke. 2021. Conversations with search engines: Serp-based conversational response generation. *ACM Trans. Inf. Syst.*, 39(4).

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2020. Question rewriting for conversational question answering. *CoRR*, abs/2004.14652.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020. Query resolution for conversational search with limited supervision. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 921–930, New York, NY, USA. Association for Computing Machinery.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Xiaoqiang Wang, Bang Liu, Siliang Tang, and Lingfei Wu. 2022. QRelScore: Better evaluating generated questions with deeper understanding of context-aware relevance. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 562–581, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Fanyou Wu, Weijie Xu, Chandan Reddy, and Srinivasan Sengamedu. 2024. Synthesizing conversations from unlabeled documents using automatic response segmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8012–8026, Bangkok, Thailand. 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.

Fanghua Ye, Meng Fang, Shenghui Li, and Emine Yilmaz. 2023. Enhancing conversational search: Large language model-aided informative query rewriting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5985–6006, Singapore. Association for Computational Linguistics.

Zheng Ye, Liucun Lu, Lishan Huang, Liang Lin, and Xiaodan Liang. 2021. Towards quantifiable dialogue coherence evaluation. 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 2718–2729, Online. Association for Computational Linguistics.

Yi-Ting Yeh and Yun-Nung Chen. 2019. FlowDelta: Modeling flow information gain in reasoning for conversational machine comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 86–90, Hong Kong, China. Association for Computational Linguistics.

Chanwoong Yoon, Gangwoo Kim, Byeongguk Jeon, Sungdong Kim, Yohan Jo, and Jaewoo Kang. 2024. Ask optimal questions: Aligning large language models with retriever's preference in conversational search. *Preprint*, arXiv:2402.11827.

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

Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-shot conversational dense retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 829–838, New York, NY, USA. Association for Computing Machinery.

Munazza Zaib, Wei Emma Zhang, Quan Z. Sheng, Adnan Mahmood, and Yang Zhang. 2022. Conversational question answering: a survey. *Knowl. Inf. Syst.*, 64(12):3151–3195.

Hamed Zamani, Johanne R. Trippas, Jeff Dalton, and Filip Radlinski. 2022. Conversational information seeking. *CoRR*, abs/2201.08808.

Tianhua Zhang, Kun Li, Hongyin Luo, Xixin Wu, James R. Glass, and Helen M. Meng. 2024. Adaptive query rewriting: Aligning rewriters through marginal probability of conversational answers. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13444–13461, Miami, Florida, USA. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Chenguang Zhu, Michael Zeng, and Xuedong Huang. 2018. Sdnet: Contextualized attention-based deep network for conversational question answering. *CoRR*, abs/1812.03593.

Appendix

A Examples of Synthetic Dialogs

Below are snippets of two synthetic dialogs generated from the DOC2DIAL data. Each dialogic pair contains a contextualized and decontextualized user turn (USER, USER decont.) and the system response (SYSTEM). Notice that user turns 2–5 of Dialog 7 make references to user turn 1 ('it' refers to 'Board Appeal', as can be seen in the corresponding USER decont. turns). Moreover, Turn 8 mentions a ('Board Appeal') decision, omitting 'Board Appeal', which is mentioned in previous turns. Similarly, for Dialog 356, Turns 2–4 make references to 'HazMat endorsement', without explicitly mentioning it. Notice that users may ask multiple questions in their turns (Turn 3).

-- Synthetic Doc2Dial Dialog id 7 --

-- 1 --
USER : How can I submit a Decision Review Request for a Board Appeal?
USER decont. : How can I submit a Decision Review Request for a Board Appeal?
SYSTEM : To submit the Decision Review Request: Board Appeal VA Form 10182, you can apply by mail, in person, or by fax.

-- 2 --
USER : What are the steps to apply for it by mail?
USER decont. : What are the steps to apply for a Board Appeal by mail?
SYSTEM : To apply for a Board Appeal by mail, you need to send the completed VA Form 10182 to the address: Board of Veterans Appeals, PO Box 27063, Washington, D.C. 20038.

-- 3 --
USER : How can I apply for it in person?
USER decont. : How can I apply for a Board Appeal in person?
SYSTEM : To apply for a Board Appeal in person, you need to bring your completed VA Form 10182 to a regional benefit office.

-- 4 --
USER : Can I apply for it by fax?
USER decont. : Can I apply for a Board Appeal by fax?
SYSTEM : Yes, to apply for a Board Appeal by fax, you need to fax your completed VA Form 10182 to 844-678-8979.

-- 5 --
USER : How can I get a copy of VA Form 10182 to fill out for a Board Appeal?
USER decont. : How can I get a copy of VA Form 10182 to fill out for a Board Appeal?
SYSTEM : You can ask a regional benefit office for a copy of VA Form 10182 to fill out for a Board Appeal. You can also call the VA toll-free hotline at 800-827-1000, Monday through Friday, 8:00 a.m. - 9:00 p.m. ET, to request VA Form 10182 for a Board Appeal.

-- 6 --
USER : What is the time limit to request it?
USER decont. : What is the time limit to request a Board Appeal?
SYSTEM : You have 1 year from the date on your decision to request a Board Appeal. However, if you're one of multiple people claiming the same benefit (which is rare), you have 60 days to request a Board Appeal.

-- 7 --
USER : What can I do if I disagree with the Board's decision?
USER decont. : What can I do if I disagree with the Board's decision?
SYSTEM : If you disagree with the Board's decision and have new and relevant evidence that supports your case, you can file a Supplemental Claim after a Board decision. You can also appeal to the U.S. Court of Appeals for Veterans Claims after a Board decision.

-- 8 --
USER : Do I need to do anything while waiting for the decision?
USER decont. : Do I need to do anything while waiting for the Board Appeal decision?
SYSTEM : After you request a decision review like a Board Appeal, you don't need to do anything while you wait unless VA sends you a letter asking for more information. However, if VA schedules exams for you after you request a decision review like a Board Appeal, be sure not to miss them.

-- Synthetic Doc2Dial Dialog id 356 --

-- 1 --
USER : What is the validity period of a CDL with HazMat endorsement issued by the DMV if the background checks are not completed before the endorsement expires?
USER decont. : What is the validity period of a CDL with HazMat endorsement issued by the DMV if the background checks are not completed before the endorsement expires?
SYSTEM : The DMV will provide a CDL with the HazMat endorsement that is valid for 90 days only if you applied to renew a current NY State CDL with a HazMat endorsement and the background checks are not completed and approved before the endorsement expires.

-- 2 --
USER : What happens if I don't pass the written test or background checks for it?
USER decont. : What happens if I don't pass the written test or background checks for the HazMat endorsement?
SYSTEM : If you do not pass the written test or the background checks for the HazMat endorsement,

the DMV will issue a CDL that does not have a HazMat endorsement.

-- 3 --
USER : How will I be notified if I am approved for it? And what should I do after receiving the notification?
USER decont. : How will I be notified if I am approved for the HazMat endorsement? And what should I do after receiving the notification?
SYSTEM : The DMV will mail you notification of approval for the HazMat endorsement along with instructions to go into a DMV office to obtain a CDL with the HazMat endorsement.

-- 4 --
USER : What information will be displayed on my new CDL with it?
USER decont. : What information will be displayed on my new CDL with the HazMat endorsement?
SYSTEM : Your new CDL with the HazMat endorsement will display the expiration date of your CDL and the expiration date of your HazMat endorsement.

B Prompts Used to Generate Dialogs

B.1 Step 1 prompt

```
Read the document you will be given and look for questions and answers in it. Return propositions if the document includes information that could actually answer user questions. If the document only has links or vague information that can't answer questions, do not return propositions. Also, do not return propositions if the document only has questions. If the document does have questions and answers, break them down into simple and clear propositions that make sense on their own. Recognize the language of the document given below and provide the propositions in the original language as the given Document.
If you do not create propositions the reply must be an empty list such as [] and nothing else.

Here is a document:
<document>
{text}
</document>

To generate propositions you need to:
1. Split compound sentence into simple English sentences. Maintain the original phrasing from the input whenever possible.
2. For any named entity that is accompanied by additional descriptive information, separate this information into its own distinct proposition.
3. Decontextualize the proposition by adding necessary modifier to nouns or entire sentences and replacing pronouns (e.g., "it", "he", "she", "they", "this", "that") with the full name of the entities they refer to.
4. Present the results as a list of strings, formatted in JSON. Provide only the JSON and nothing else.
```

B.2 Step 2 prompts

Prompt 2.1

To maintain the dialog flow, we instruct the model to keep relevant (to each other) queries in adjacent turns. We also encourage the LLM to generate queries that are grounded in more than one proposition. The purpose of the first two instructions, which are related to turns where the user and system exchange greetings, is to mimic real dialogs, but can be skipped without affecting the quality of the dialogs. The output is a JSON dictionary of question-answer pairs, each containing the decontextualized query and its answer.

```
Your task is to read the given propositions and generate a dialog between a user and a system, where the user asks certain questions and the system tries to provide answers.
Follow these instructions:
1. Your response should be a JSON of the following format:

{
  "0" : {
    "<user>": ,
    "<system>":,
  },
  "1" : {
    "<user>": ,
    "<system>":,
  },
  ...
}

2. The dialog must start with the user greeting the system and the system replying politely.
3. The dialog must end with user thanking the system and the system replying politely.
4. In each dialog turn, the user asks a question based on a given proposition. The user question must be a self-contained, standalone question without the need to refer to previous dialog context.
5. A user may also ask complex questions, for which the answer can be two or more propositions.
6. In each dialog exchange the system answers the user question based on the propositions.
7. Make sure that the user questions referring to the same propositions are in adjacent turns.
8. Each system's answer must be a full sentence.

<propositions>
{}
</propositions>
```

Prompt 2.2

```
Your task is to read the given dialog. The dialog you will be given has a JSON format. The key <user> refers to user utterances, while the key <system> refers to the system utterances. Make the user utterances dependent on previous dialog turns taking into account the dialog context and using pronouns to replace already mentioned information only if such information is already mentioned in the previous dialog turns. Only return a JSON of the following format:

{
```

```
  "0" : {
    "<contextualized user>": ,
    "<system>":
  },
  "1" : {
    "<contextualized user>": ,
    "<system>":
  },
  ...
}

Here is the dialog:
<dialog>
{}
</dialog>
```

Prompt 2.3

```
I will give you a list of propositions and a text in JSON format of question and answer pairs generated from these propositions. I need you to act as a human annotator and evaluate the question and answer pairs provided following these instructions:
1. Provide a separate review and evaluation for each question and answer.
2. First check if the questions provided are correctly generated from the propositions provided.
3. The answer to each question should be reflecting the information provided in the propositions.
4. Note which propositions are used in each answer.
5. If a question and answer is generated from the provided propositions after your review, mark it as "accepted". If not, mark it as "not_accepted".
6. The first and last pairs should always be accepted.
7. Return only a dictionary in JSON format and nothing else. The key of each dictionary should be the same with each question answer pair given. Follow the example:

{
  "0": {
    "propositions_used":
    ,
    "explain_evaluation": ,
    "evaluation": ,

  },
}

Here are the propositions and the question-answer pairs:

<propositions>
{}
</propositions>

<question and answer pairs>
{}
</question and answer pairs>
```

B.3 Response-LLM prompt

For response generation, we prompt LLAMA-8B instruct, to generate a system response, conditioned on the top-20 retrieved propositions and the rewritten query (using our fine-tuned query rewriter). If the query cannot be answered using the provided propositions, the LLM is instructed to generate the token <cannot_answer>.

```
Your job is to answer user questions given a set of propositions in a list format. There may be irrelevant propositions included.
You only need to provided the answer. If the question cannot be answered using the provided propositions, generate the token <cannot_answer> only.

Here are the propositions: {}

Here is the user question: {}
```

C BM25 and Dense Retrieval Results

We present results of BM25 and dense retrieval for proprietary and DOC2DIAL documents, separately, in Table 8. Both retrievers exhibit the same behavior as the RRF retriever (Tables 2–3); T5 manages to outperform the contextualized query, but not the 'ground-truth' decontextualized one. Both retrievers perform worse than the RRF retriever.

PR		**MAP↑**	**R@5↑**	**R@10↑**	**R@20↑**
Dense	Query_{co}	0.41	0.51	0.57	0.63
	T5	0.46	0.54	0.61	0.66
	Query_{de}	**0.48**	**0.58**	**0.65**	**0.72**
BM25	Query_{co}	0.42	0.51	0.57	0.63
	T5	0.45	0.54	0.61	0.66
	Query_{de}	0.47	0.56	0.63	0.68
PU		**MAP↑**	**R@5↑**	**R@10↑**	**R@20↑**
Dense	Query_{co}	0.45	0.54	0.62	0.69
	T5	0.49	0.57	0.67	0.73
	Query_{de}	**0.51**	**0.60**	**0.68**	**0.74**
BM25	Query_{co}	0.45	0.53	0.60	0.66
	T5	0.48	0.56	0.63	0.70
	Query_{de}	0.49	0.59	0.65	0.71

Table 8: **Dense and BM25 retrieval results** in **synthetic dialogs** generated via **propositions**, using proprietary (PR) and public (PU) documents.

D Experiments in Separate Domains

In many real-life scenarios, the domain of each question is known during inference; the user may also explicitly request documents for a particular domain. Thus, the retriever only needs to consider information of the corresponding domain. To simulate such a scenario, we split propositions and questions based on their domains. Both proprietary and DOC2DIAL datasets include documents and questions from four distinct domains (§3.1). The results are presented in Tables 9– 10 for proprietary and DOC2DIAL dialogs, respectively. Regardless of dataset or domain, we reach the same conclusions regarding the performance of the question rewriter as in the main text (§4.3); the question rewriter outperforms using the original contextualized queries, but not the 'ground-truth' decontextualized queries.

In the proprietary data, we notice a drop in performance for the Miscellaneous (Misc) and Finance domains, compared to the main experiments (Table 3). This is mostly due to the more complex and much longer user questions of these two domains. A similar observation can be made for the SSA domain of DOC2DIAL. For the rest of the domains of both datasets, the performance is equal to, or better than the performance reported in the main experiments, which is to be expected, since the retriever has to consider fewer propositions.

	MAP↑	R@5↑	R@10↑	R@20↑
		Finance		
Query$_{co}$	0.38	0.44	0.51	0.57
T5	0.39	0.46	0.52	0.59
Query$_{de}$	**0.40**	**0.48**	**0.54**	**0.60**
		Software		
Query$_{co}$	0.45	0.56	0.63	0.71
T5	0.49	0.58	0.65	0.73
Query$_{de}$	**0.45**	**0.60**	**0.67**	**0.74**
		Insurance		
Query$_{co}$	0.59	0.70	0.75	**0.79**
T5	0.59	0.71	0.76	**0.79**
Query$_{de}$	**0.62**	**0.72**	**0.77**	**0.79**
		Misc		
Query$_{co}$	0.30	0.36	0.44	0.49
T5	0.30	0.36	0.45	0.46
Query$_{de}$	**0.33**	**0.41**	**0.49**	**0.50**

Table 9: RRF **retrieval results** in **synthetic dialogs** generated via **propositions** from **proprietary documents**, **separately for each domain** .

	MAP↑	R@5↑	R@10↑	R@20↑
		DMV		
Query$_{co}$	0.56	0.66	0.75	0.80
T5	**0.60**	0.68	0.76	0.82
Query$_{de}$	**0.60**	**0.69**	**0.77**	**0.83**
		VA		
Query$_{co}$	0.49	0.58	0.65	0.72
T5	0.54	0.65	0.72	0.79
Query$_{de}$	**0.57**	**0.69**	**0.77**	**0.84**
		SSA		
Query$_{co}$	0.44	0.55	0.62	0.68
T5	0.44	0.55	0.62	0.69
Query$_{de}$	**0.46**	**0.56**	**0.63**	**0.70**
		StudentAid		
Query$_{co}$	0.56	0.63	0.70	0.77
T5	0.57	0.65	0.72	**0.78**
Query$_{de}$	**0.58**	**0.66**	**0.73**	**0.78**

Table 10: RRF **retrieval results** in **synthetic dialogs** generated via **propositions** from **public documents** (DOC2DIAL), **separately for each domain**.