# Generate then Retrieve: Conversational Response Retrieval Using LLMs as Answer and Query Generators

**Zahra Abbasiantaeb** and **Mohammad Aliannejadi**
University of Amsterdam
Amsterdam
The Netherlands

## Abstract

Conversational information seeking (CIS) is a prominent area in information retrieval (IR) which focuses on developing interactive knowledge assistants. These systems must adeptly comprehend the user's information requirements within the conversational context and retrieve the relevant information. To this aim, the existing approaches model the user's information needs with one query called rewritten query and use this query for passage retrieval. In this paper, we propose three different methods for generating multiple queries to enhance the retrieval. In these methods, we leverage the capabilities of large language models (LLMs) in understanding the user's information need and generating an appropriate response, to generate multiple queries. We implement and evaluate the proposed models utilizing various LLMs including GPT-4 and Llama-2 chat in zero-shot and few-shot settings. In addition, we propose a new benchmark for TExt Retrieval Conference (TREC) Interactive Knowledge Assisstance Track (iKAT) based on GPT-3.5 judgments. Our experiments reveal the effectiveness of our proposed models on TREC iKAT dataset.

## 1 Introduction

Conversational information seeking (CIS) is a well-established topic in information retrieval (IR) (Zamani et al., 2022), where a knowledge assistant interacts with the user to fulfill their information needs. While conversations can be complex (Radlinski and Craswell, 2017), involving various types of interactions such as revealment and clarification, one of the main goals of the system is to provide accurate responses to users' queries during the conversation. The TExt Retrieval Conference (TREC) Conversational

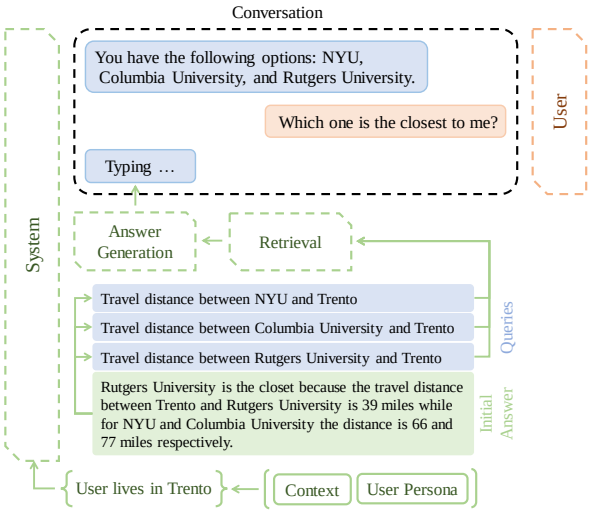


Figure 1: An example conversation with a complex user request. The system in this case generates three distinct queries from the initial answer and searches for every query in the passage collection. Then it reads and reasons over the top passages to generate the final grounded answer.

Assistance Track (CAsT) 2019–2022 (Dalton et al., 2020) focuses on the development of conversational knowledge assistants, while the main focus of this track is on evaluating passage retrieval and dialogue context modeling. TREC Interactive Knowledge Assistance Track (iKAT) (Aliannejadi et al., 2024) evolves CAsT into emphasizing the development of personalized conversational knowledge assistants, where each dialogue is coupled with a knowledge base describing the user. This knowledge base contains various types of information, ranging from personal details to past experiences and exchanges with the system (Aliannejadi et al., 2024). It is assumed that the user has revealed this information in their previous interactions with the system, thereby impacting the system's response to the same request based on the user's background.

To ensure the groundedness and accuracy

of responses (Semnani et al., 2023), existing methods follow a **retrieve-then-generate (RG)** pipeline, breaking the task into several subtasks: dialogue context modeling, retrieval, and answer generation (Voskarides et al., 2019; Yu et al., 2021). The user's information need is often represented either by a single rewritten query (Voskarides et al., 2019; Yu et al., 2020) or by a single representation in the query embedding space (Yu et al., 2021; Hai et al., 2023).

Representing complex information needs using only one query leads to several limitations, especially in cases where the information cannot be answered using a single passage and requires complex reasoning over multiple facts in a chain-of-thought scenario (Aliannejadi et al., 2024; Lyu et al., 2023). Take the user query of Figure 1 as an example. Clearly, it is unlikely to find a passage that has distance information about all these universities compared to the user's address. Therefore, the system would need to gather relevant information from different sources (i.e., issue multiple queries) and reason over the gathered evidence to generate the final response. Existing RG methods often fail to answer such complex queries (Aliannejadi et al., 2024). This is because existing ranking methods rely solely on semantic similarity between a query and a passage, without high-level reasoning or control over the set of retrieved passages. For example, they do not ensure that the top results contain address information about all three universities the user is interested in.

To address these challenges, in this work, we propose to rely on the knowledge and capabilities of large language models (LLMs) to respond to complex user queries in a dialogue. Specifically, inspired by the relevant literature (Shuster et al., 2022b), we study several methods based on the **generate-then-retrive (GR)** pipeline. We first prompt the LLM to generate an answer for a given dialogue, followed by various retrieval strategies to ground the generated answer in the passage collection. These strategies include: (i) using the generated answer as a query to rank passages (Gao et al., 2023); and (ii) a novel approach of breaking the answer into several searchable queries. We hypothesize that leveraging an LLM's knowledge and reasoning capacity leads to a more accurate response, enabling the model to generate more effective queries based on the generated answer. For example, in Figure 1, the LLM's answer that compares the distance of the three universities helps it generate more effective queries, as shown in the figure.

This leads us to our first research question: **(RQ1)** *Can we leverage LLMs' internal knowledge and reasoning capacity to enhance conversational passage retrieval in a GR pipeline?* To answer this question, we propose various GR retrieval pipelines based on GPT-4 and LLaMA. Regarding the GR-based methods, we address our next research question: **(RQ2)** *Can we leverage LLMs's reasoning capabilities to generate related searchable queries to enhance retrieval?* We answer this question by comparing our proposed query generation method with three alternatives: (i) using the generated answer as a query, (ii) directly prompting the LLM to generate queries before generating an answer, and (iii) prompting the LLM to generate the query rewrite. We conduct extensive experiments on the iKAT 2023 dataset,[1] where we find that GR pipelines significantly outperform their RG counterparts by a large margin, and generating queries based on the LLM's answer leads to more effective queries and retrieval.

Given that only a handful of GR runs were evaluated in the official iKAT assessment pool, we find that the released relevance assessments do not accurately reflect the effectiveness of our proposed methods, leading to a very high rate of unjudged passages in the top 10 results (72.78%). Recent work shows that LLMs can accurately predict passage relevance (Faggioli et al., 2023; MacAvaney and Soldaini, 2023), leading to high agreement with human assessors. We follow the same approach and leverage GPT-3.5 in a one-shot manner to assess all the passages in our new pool. To ensure our assessment is not biased by the choice of the LLM, we conduct extensive experiments and study the correlation and agreement with human labels (see appendix).

We summarize our contributions as follows:

- We propose three GR-based conversa-

[1] https://trecikat.com

tional passage retrieval models, leveraging the LLM's internal knowledge to generate an answer and break it down into multiple queries.
- We implement our method using commercial and open-source LLMs, demonstrating its effectiveness in both settings.
- We investigate the usability of existing relevance judgments in evaluating GR-based models. We collect and release GPT-3.5-based relevance judgments, incorporating the new models into the judgment pool.[2]
- We conduct extensive experiments, showcasing the effectiveness of our proposed approach under various conditions.

## 2 Related Work

**Conversational information seeking.** Recently, CIS has gained significant popularity in both IR and natural language processing (NLP) communities (Anand et al., 2020). Similar to knowledge-intensive dialogues (Dinan et al., 2019; Feng et al., 2021; Li et al., 2022), a key challenge in CIS is to model the dialogue context to better understand the user information need and perform effective retrieval (Zamani et al., 2022). As outlined by Radlinski and Craswell (2017), a CIS dialogue could consist of numerous user–system interactions, such as revealment, information request, and feedback, making it more complex for the existing systems to model dialogue context and user information need. TREC CAsT 2019–2022 (Dalton et al., 2020) and iKAT 2023 (Aliannejadi et al., 2024) aim at addressing these challenges through a common evaluation framework where complex and knowledge-intensive dialogues were provided to the participants, as well as several passage collections. The goal was to retrieve relevant passages for each turn in a dialogue and generate a response synthesizing several passages.

**Retrieval pipeline.** Most existing methods follow the two-step pipeline for retrieval which includes first-stage retrieval and re-ranking. First-stage retrieval aims to extract as many relevant passages from the collection. In this step, the recall is more important, and models like BM25 are usually used. The goal of re-ranking is to improve the precision by bringing the most relevant passages to the top of the list of passages returned by retrieval. The Cross-encoder model is one of the widely used re-ranker models.

**Modeling dialogue context.** Most existing methods tackle this problem by query rewriting where the goal is to address the ambiguity and dependence of a user utterance by resolving its dependencies and making it self-contained (Voskarides et al., 2019, 2020; Lin et al., 2021b). The rewritten query is supposed to be a self-contained and context-independent query that represents the user's information needs per turn. CRDR (Qian and Dou, 2022) forms the rewritten query by modifying the query by disambiguation of the anaphora and ellipsis. The existing work trains GPT-2 (Yu et al., 2020; Vakulenko et al., 2021) and T5 (Dalton et al., 2020) models on CANARD (Elgohary et al., 2019) to generate the rewritten query. The input of these models is created based on different combinations of the previous user turns and system responses. The CANARD dataset (Elgohary et al., 2019) is widely used for training such models, including the manually rewritten queries of QuAC (Choi et al., 2018), which is a conversational machine reading comprehension dataset that is collected manually by pairing a teacher and a student annotator. Another line of research aims at learning to represent the dialogue context directly for passage retrieval (Yu et al., 2021; Hai et al., 2023), where a distillation loss learns to map the representation of the whole dialogue context to the one of the gold resolved query, hence improving the dense retrieval performance.

Our work distinguishes itself from these works by leveraging the power of LLMs to model dialogue context and uses the LLMs' knowledge to answer the user's question directly first, and then try to ground its answer on the passage collection.

**Grounding LLM-generated responses.** IR is a common approach to ground LLM-generated responses and build grounded chatbots. BlenderBot 2–3 (Shuster et al., 2022b) utilizes the results of web search for

[2]The code and data will be released upon acceptance.

this purpose, while SeeKer (Shuster et al., 2022a) employs a three-step approach in which it generates search queries, extracts useful knowledge from the top passages, and generates the final answer. Semnani et al. (2023) learns to avoid hallucination and improve the factuality of the generated responses by searching and grounding the responses on Wikipedia articles. Other approaches such as GenRead (Yu et al., 2023) study the effectiveness of LLMs in generating multiple passages for a given question and use the generated passages to generate the final response.

Differently, in this work, we focus on modeling the dialogue context through LLMs' response and query generation and use that to enhance passage retrieval performance. Our goal is to leverage the LLM's response as a means to model and expand the user utterance, enhancing retrieval performance.

## 3 Methodology

**Task definition.** Each conversation revolves around a topic $t$ and starts with a user utterance. A conversation includes several turns, where a turn starts with a user utterance $u_i$, followed by a system response called $r_i$. A conversation, which comprises a set of turns, is represented as $(u_1, r_1), ..., (u_n, r_n)$. The TREC iKAT dataset also contains the user's persona. The user's persona is a knowledge base, consisting of a set of statements shown as $PTKB = \{s_1, ..., s_l\}$, where each statement $s_i$ is a natural language sentence. The task of conversational assistants is defined as follows: (i) retrieving relevant passages to the current user utterance from the collection $D = \sum_{i=1}^{|D|} d_i$, and (ii) generating the response $r_i$ given the user utterance $u_i$ and previous turns $(u_1, r_1), ..., (u_{i-1}, r_{i-1})$, grounded on the retrieved passages.

**Retrieve then generate.** The raw user utterance cannot be used as a query for the passage retrieval step, as it is not a self-contained query and depends on the context (Dalton et al., 2020). A common practice is to form the rewritten query $q_i$ for the current user utterance $u_i$ (Voskarides et al., 2019; Yu et al., 2020). The rewritten query is used in a first-stage retrieval model such as BM25 (Robertson and Zaragoza, 2009). The top passages returned by the first-stage retriever are then passed to the re-ranker model. The top-k passages returned by the re-ranker are finally passed to the response generation model to generate the response $r_i$.

**Generate then retrieve.** Inspired by existing work (Gao et al., 2023), we propose a set of approaches that rely on the answer generated by the LLM. Our goal is to leverage the LLM's internal knowledge and reasoning capability to improve passage retrieval, by generating a set of $n$ queries $\{q_1^i, \ldots, q_n^i\}$ given the current user utterance $u_i$ and the context of the conversation:

- **AD**: In this approach, we treat the LLM's answer $r_i'$ as a single long query and pass it to both the first-stage retrieval and re-ranker models.
- **QD:** We prompt the LLM to directly generate a maximum of five queries to search for the answer. Each query of turn $i$, denoted as $q_j^i$, is then passed to both the first-stage retrieval model and the re-ranker model. The output of the re-ranker for each query is demonstrated as $d_1^{i,j}, d_2^{i,j}, \ldots, d_m^{i,j}$. Subsequently, we interleave the results of all the queries to obtain the final ranking: $d_1^i, d_2^i, \ldots, d_k^i$.
- **AQD:** Here, we combine two other approaches: (i) prompting the LLM to generate an initial response $r_i'$, and (ii) prompting the LLM to generate up to five queries to refine its own generated answer. Similar to the QD model, we then pass these generated queries to the first-stage retrieval and re-ranking model and interleave the ranked list of all queries.
- **AQD$_\mathbf{A}$:** Interleaving the results of the generated queries is suboptimal, as some of the generated queries may be of low quality. To tackle this problem, we propose a variant to the AQD approach, where we re-rank the final ranking list, based on the predicted relevance to the generated response $r_i'$.

**Prompts:** The GPT-4 model is used as a zero-shot learner for AQD and QD approaches. The prompts used for these approaches are shown in Table 5 and 6, respectively. The LLaMA model is given a few-shot prompt for

query generation in AQD and QD approaches as is shown in Table 9 and 10, respectively. The examples of few-shots are from the pruned turns of the same dataset and output of the GPT-4 model. For answer generation in AQD we design a zero-shot prompt. The same answer generated in AQD approach is used for AD approach. Also, the AQD and $AQD_A$ approaches use the same prompt

## 4 Experimental Setup

**Hyper-parameters.** For the first-stage retrieval we employ the BM25 model from Pyserini (Lin et al., 2021a) using the default values for the parameters. For the re-ranker, we use the pre-trained Cross-Encoder model `ms-marco-MiniLM-L-6-v2` from the `sentence_-transformers` library with a maximum length of 512. We use LLaMA-chat 13B with the following parameters: top_k= 10, top_p= 0.9, temperature= 0.75. For the AQD approach, we design a two-shot prompt for LLaMA. For AD and QR, we use LLaMA as a zero-shot learner and for QD we design a one-shot in-context learning prompt for LLaMA. We use the GPT-4 model as a zero-shot learner using the default values of parameters for all approaches. We utilize the GPT-3.5 with default parameters for prediction.

**Dataset.** We report the results on the TREC iKAT dataset, consisting of 25 conversations over 13 different topics and a total number of 133 turns. The average length of conversations in iKAT is 13.04 which makes the context modeling task more challenging. We evaluate our methods on iKAT as it is one of the few datasets that features complex dialogues where single-query rewriting is not effective.

**Metric.** We evaluate passage retrieval task using the official metrics of iKAT, namely, nDCG@5, P@20, Recall@20, Recall, and mAP. nDCG@5 evaluates the scenarios where the top passages are intended to be presented to the user, while P@10 and Recall@20 measure the performance in cases where the top results are going to be used by the LLM to generate an answer, assuming that it would read and ground its answer on the top 20 passages.

**GPT-3.5-based assessment.** As mentioned earlier, the original pool of iKAT runs contains only a handful of AQD models, resulting in a significant number of unjudged passages among the top results of our proposed methods. For instance, we observe that 82.40% of the top passages in AQD approach using LLaMA model are unjudged (see Figure 2 for more details). Inspired by (Faggioli et al., 2023; MacAvaney and Soldaini, 2023), we create a new pool of passages and prompt GPT-3.5 to judge the relevance of passages. To make it a fair comparison, we take the following considerations: (i) we judge all the query–passage pairs even if they are already assessed in the original data, and (ii) include the top 10 retrieved passages by all of our proposed models and the baselines. The one-shot prompt used for relevance judgment is shown in Table 8. In this prompt, we give the ground truth answer from the dataset as an example of a passage with a score of 4 to the model. Overall, we judge the relevance of 19,413 query–passage pairs.

**Assessment bias.** We argue that using GPT-3.5 as an assessor will not favor GPT-4-based models (Liu et al., 2023). We assess query–passage pairs that are ranked by BM25 and the cross-encoder, while GPT-4 only generates the query in such pipelines. Therefore, it is very unlikely that the judgments are favoring any models, as the passages are not directly generated or ranked by GPT-4.

**Baselines.** Below we describe the baselines that were submitted as TREC runs, as well as others that we implement in this work.

- *Human* refers to using the resolved utterance provided in the gold dataset for retrieval and re-ranking.
- *InfoSense* refers to the TREC iKAT submission called *georgetown_infosense_ikat_-run_3* (Patwardhan and Yang, 2023) and follows the GR pipeline. In this run, relevant PTKB statements for each turn are determined. We give the LLaMA-chat 13B model a one-shot prompt to generate an initial answer given the context of the conversation and relevant PTKB statements. The top passages are identified using the BM25 model.
- *LLaMA10* refers to the *llama2_only_10_-docs* model (Aliannejadi et al., 2024), a

baseline model proposed by the organizers of the track and is an RG model. It prompts the LLaMA 7B model to generate the rewritten query. They use BM25 to retrieve the top 1000 passages using this rewritten query. The top 10 passages are re-ranked based on the judgment of the LLaMA model.

- *ConvGQR* is another TREC baseline that uses a pre-trained model for query rewriting and query expansion on the QReCC dataset (Anantha et al., 2021).
- *monoT5* refers to TREC's *run_automatic_dense_monot5* model that uses a pre-trained BART model for query rewriting. The BART model is fine-tuned on the SAMSum (Gliwa et al., 2019) and CANARD (Elgohary et al., 2019) datasets. They also employ a T5-based model for re-ranking.
- *AD* baselines are also based on the GR pipeline. As described earlier, the LLM is prompted to generate the answer given the context of the conversation and the persona of the user. The generated response is passed to BM25 for first-stage retrieval. The top 1000 passages returned by BM25 are re-ranked using the pre-trained Cross-Encoder model.
- *QR* baseline rewrite the query given the context of the conversation and persona of the user. The GPT-4 and LLaMA-chat 13B models are used for this approach. The rewritten query is then passed to the BM25 and Cross-Encoder model to retrieve the relevant passages. The prompts designed for the zero-shot QR model are shown in Table 7. This prompt is used for both GPT-4 and LLaMA models.

## 5 Results and Discussions

**Passage Ranking Results.** The performance of the proposed models and the baseline models are shown in Table 1. These results are based on the pool generated by GPT-3.5 model (we provide the result of passage retrieval based on the pool assessment of iKAT in Table 3). As can be seen, the AQD and $\text{AQD}_\text{A}$ models outperform the QR baseline using the LLaMA model (addressing RQ1). In addition, the $\text{AQD}_\text{A}$ approach outperforms the QR approach using human query rewrites and the AD approach in terms of nDCG, Recall, and mAP. In addition, the $\text{AQD}_\text{A}$ model achieves a better performance in terms of all metrics compared to the organizer's baseline and InfoSense (addressing RQ2).

According to Table 1, the LLaMA model in the AQD approach achieves a better performance given the answers of GPT-4 model for query generation. This result indicates the importance of the quality of the initially generated answer. Given a wrong answer, the error will propagate to the query generation step and reversely impact the quality of the generated queries.

Using the GPT-4 model as our LLM and the $\text{AQD}_\text{A}$ approach, we have achieved the best performance by outperforming the best baseline model based on GR pipeline which is AD approach with GPT-4 as our LLM. (addressing RQ1, RQ2) The AQD approach achieves a better performance in terms of mAP, Recall@20, and P@20 compared to the AD approach. The results indicate that using multiple queries for retrieval can help to retrieve more relevant documents compared to using the initially generated answer as they cover different aspects of the initially generated answer. By generating queries from an answer, we are breaking the answer into several aspects, and by using the output of the retrieval model for each of these queries, we are giving an equal chance for these aspects to appear in the ranking list. The second stage of re-ranking based on the initially generated answer is more effective than interleaving the output of the re-ranker for each query. This makes sense because the final goal here is to reconstruct the initial answer by grounding it in the collection.

The superior performance of the AQD approach, compared to the QR approach demonstrates the importance of having multiple queries rather than one query for the complex user queries. The better performance of models with GR pipeline compared to the models with RG pipeline represents the effectiveness of using the knowledge and power of LLMs for the retrieval task.

The AQD model generates the queries in two steps, first doing reasoning and generat-

Table 1: The results of passage retrieval task iKAT dataset on the GPT-3.5 benchmark for different pipelines (P.) and approaches (A.). [H]: using human-rewritten queries, [G]: using answers generated by GPT-4. The superscripts indicate the results of models that are significantly different, as determined by a two-sided paired t-test with a Bonferroni correction, at a significance level of $p < .05$. Given the space limit, we run the test only on the runs that belong to the same block in the table (except for AD-GPT-4).

| P. | A. | Model | nDCG@5 | nDCG | P@20 | Recall@20 | Recall | mAP |
|---|---|---|---|---|---|---|---|---|
| GR | QD | (a) GPT-4 | .5982$^{(cbd)}$ | .3839$^{(cbd)}$ | .6143$^{(b)}$ | .1259$^{(b)}$ | .3275$^{(d)}$ | .1942$^{(d)}$ |
| GR | AQD | (b) GPT-4 | .6300$^{(ad)}$ | .4020$^{(ad)}$ | **.6414**$^{(ad)}$ | **.1313**$^{(ad)}$ | .3384$^{(d)}$ | .2018 |
| GR | AQD | (c) GPT-4 [H] | .6594$^{(a)}$ | .4023$^{(ad)}$ | .6282 | .1298 | .3255$^{(d)}$ | .1964$^{(d)}$ |
| GR | $AQD_A$ | (d) GPT-4 | **.6826**$^{(ab;n)}$ | **.4799**$^{(acb;n)}$ | .6124$^{(b;n)}$ | .1239$^{(b;n)}$ | **.4771**$^{(acb;n)}$ | **.2079**$^{(ac;n)}$ |
| GR | QD | (e) LLaMA | .3925$^{(fghi)}$ | .2161$^{(fghi)}$ | .4184$^{(fghi)}$ | .0822$^{(fghi)}$ | .1947$^{(fghi)}$ | .0959$^{(fghi)}$ |
| GR | AQD | (f) LLaMA | .5033$^{(ei)}$ | .3004$^{(egi)}$ | .5421$^{(ei)}$ | .1092$^{(ei)}$ | .2657$^{(ei)}$ | .1458$^{(ei)}$ |
| GR | AQD | (g) LLaMA [G] | .5279$^{(e)}$ | .3184$^{(efi)}$ | .5331$^{(e)}$ | .1082$^{(e)}$ | .2743$^{(ei)}$ | .1503$^{(ei)}$ |
| GR | AQD | (h) LLaMA [H] | .4851$^{(ei)}$ | .2974$^{(ei)}$ | .5124$^{(e)}$ | .1048$^{(e)}$ | .2562$^{(ei)}$ | .1380$^{(ei)}$ |
| GR | $AQD_A$ | (i) LLaMA | .5571$^{(efh)}$ | .3906$^{(efgh)}$ | .5474$^{(ef)}$ | .1108$^{(ef)}$ | .4092$^{(efgh)}$ | .1771$^{(efgh)}$ |
| | | **Baselines** | | | | | | |
| RG | QR | (j) Human | .6193$^{(lm)}$ | .3422$^{(klm)}$ | .5658$^{(klm)}$ | .1173$^{(klm)}$ | .2877$^{(klm)}$ | .1593$^{(klm)}$ |
| RG | QR | (k) GPT-4 | .5983$^{(lm)}$ | .2373$^{(jm)}$ | .5274$^{(jlm)}$ | .1071$^{(jlm)}$ | .1464$^{(jlm)}$ | .1164$^{(jm)}$ |
| RG | QR | (l) LLaMA | .4521$^{(jkm)}$ | .2431$^{(jm)}$ | .4538$^{(jkm)}$ | .0903$^{(jkm)}$ | .2211$^{(jkm)}$ | .1145$^{(jm)}$ |
| RG | QR | (m) LLaMA10 | .3667$^{(jkl)}$ | .1081$^{(jkl)}$ | .3203$^{(jkl)}$ | .0621$^{(jkl)}$ | .0621$^{(jkl)}$ | .0512$^{(jkl)}$ |
| GR | AD | (n) GPT-4 | .6433$^{(opqr;d)}$ | .4038$^{(opqr;d)}$ | .5793$^{(opqr;d)}$ | .1177$^{(opqr;d)}$ | .3739$^{(opqr;d)}$ | .1721$^{(opqr;d)}$ |
| GR | AD | (o) LLaMA | .5599$^{(nqr)}$ | .2504$^{(npq)}$ | .5342$^{(npqr)}$ | .1076$^{(npqr)}$ | .1705$^{(npqr)}$ | .1253$^{(npqr)}$ |
| GR | AD | (p) InfoSense | .5274$^{(nqr)}$ | .2135$^{(nor)}$ | .4602$^{(noq)}$ | .0919$^{(noqr)}$ | .1407$^{(noqr)}$ | .1004$^{(noq)}$ |
| RG | QR | (q) ConvGQR | .3978$^{(nopr)}$ | .2115$^{(nor)}$ | .3853$^{(nopr)}$ | .0686$^{(nopr)}$ | .2112$^{(nopr)}$ | .0835$^{(nopr)}$ |
| RG | QR | (r) monoT5 | .4557$^{(nopq)}$ | .2676$^{(npq)}$ | .4316$^{(noq)}$ | .0770$^{(nopq)}$ | .2790$^{(nopq)}$ | .0986$^{(noq)}$ |

ing the desired answer, and second generating queries from this answer. In the QD approach the queries are directly generated from the context of the conversation. The better performance of the AQD approach compared to the QD approach indicates the importance of relying on the power of LLMs in context modeling and reasoning for generating the answer. The answer generated by LLM is a more helpful source for generating queries compared to the context of the conversation.

The performance of the AQD approach using resolved utterance is presented in Table 1 to study the power of LLMs in query understanding. Note that the manual AQD-GPT-4 model performs slightly better than the automatic AQD-GPT-4 model in terms of lower cutoffs of Precision, Recall, and nDCG. While the automatic AQD-GPT-4 model achieves a better performance in terms of Recall and mAP. The result of all the metrics for the manual run of the LLaMA model is lower than the automatic run where the LLaMA model relies on its own knowledge to understand the context of the conversation and generate an answer.

Performance of different QR models shows that the LLaMA and GPT-4 models as zero-shot learners cannot outperform humans in query rewriting tasks.

**Pooling.** To demonstrate the extent of lacking relevance judgments in the runs, here we report the Judged@10 of some of the runs. The results of all the runs are available in Table 3 in the appendix. Judged@10 simply reports the average number of judged passages (between 0 and 10) in the top 10 passages of each run. We see in Figure 2 that the number of judged passages in the top 10 passages returned by our proposed models is very low (∼2), compared to the models that participated at iKAT (AQD-GPT-4 and InfoSense), leading to unfair evaluation. This is evident in Table 3 where we report the performance of all the models using the official iKAT relevance labels, where we see that all our proposed approaches are inferior to the iKAT runs, which is expected given the very low rate of judged passages. Therefore, we generated a new assessment pool using GPT-3.5 model, containing a total number of 19,413 relevance judgments over 133 conversational turns. To demonstrate the extent to which

Table 2: Confusion matrix comparing binary relevance judgments made by TREC assessors and one-shot GPT-3.5 on TREC iKAT 2023.

| | | TREC iKAT 2023 | |
|---|---|---|---|
| | | Relevant | Irrelevant |
| GPT-3.5 | Relevant | 12,885 | 450 |
| | Irrelevant | 7146 | 2,203 |

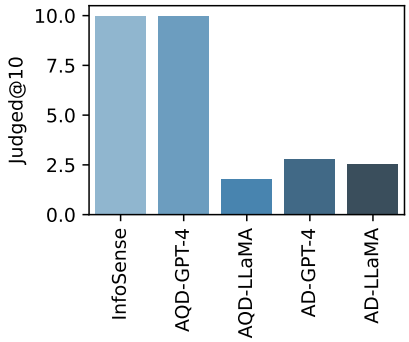


Figure 2: Judged@10 metric in TREC iKAT official relevance assessments for the first 10 passages returned by each model.

the GPT-3.5-judgments agree with human labels, we re-generate the pool of TREC iKAT benchmark and compare the performance of GPT-3.5 one-shot model as an assessor with humans. According to Table 2, the GPT-3.5 and human assessors have 66.84% agreement in terms of binary relevance judgment. Additionally, the GPT-3.5 tends to give a higher relevance score compared to humans.

**Analysis.** Performance of the proposed models per depth of the conversation and topic are shown in Figures 4 and 3. As can be seen, as the conversation continues and the context modeling becomes more challenging (turn>14), the AQD and $AQD_A$ approaches represent a better retrieval performance compared to the AD model.

## 6 Conclusion

We propose two different models based on the GR pipeline for enhancing the retrieval performance of CIS. The proposed models work by generating multiple queries rather than using one rewritten query. We rely on the power of LLMs to understand the user's information needs and generate an appropriate response. In AQD approach, we prompt LLM to answer the user's question and then generate multiple

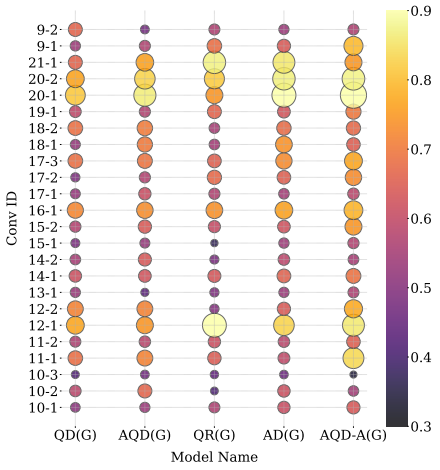


Figure 3: Performance of the proposed models per each conversation ((G): GPT-4).

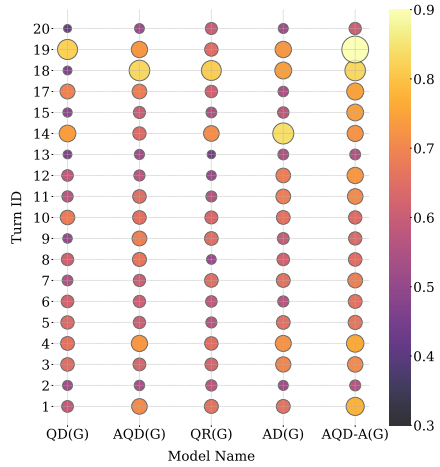


Figure 4: Performance of the proposed models per each conversational depth ((G): GPT-4)

to refine that answer. We do retrieval and re-ranking for all queries and interleave the ranking output for all queries. In $AQD_A$ approach, which is a variant of AQD model, after retrieving the relevant passages for each query, we re-rank the top passages for these queries using the initial answer. In QD approach we directly prompt the LLM to generate the queries given the context of the conversation and the user utterance. We conduct extensive experiments to study the performance of the proposed approaches. The experiments demonstrate the effectiveness of generating multiple queries compared to one rewritten query. Additionally, we propose a new assessment pool for iKAT dataset generated by GPT-3.5.

## 7 Limitations

We present two Generate-then-Retrieve-based methods for improving the retrieval performance. Our proposed methods rely on the answer generated by LLM and try to ground the generated answer to the given collection in different ways. So, the retrieval is biased toward the answer generated by LLM, and any error or limitation in the answer generation would influence the retrieval and re-ranking. For example, if the LLM does not know about a specific topic, it cannot generate a correct and complete answer. The low-quality answer will result in generating low-quality and non-relevant queries which decreases the performance of retrieval. Our main focus in this work is on improving the performance of retrieval and we have not studied the quality of the answer generation using our proposed models for generating the final answer. In addition, we prompt the LLM to generate a maximum number of 5 queries. The impact of the number of queries is not studied in this work. The impact of different numbers of queries on the performance of retrieval and generating the final answer to the user question remains an interesting future work.

## 8 Ethical considerations

Stressing the need to study and measure biases in Language Models (LLMs) when generating data, we think it could cause unexpected ethical issues. Consequently, we need to study the potential biases that exist in the data and formalize their impact on the final output of the model. While in this study we propose to use the answers and queries generated by LLMs for retrieval models, we think these methods should be used carefully in real-world retrieval systems, and designers should consider these biases.

## References


Mohammad Aliannejadi, Zahra Abbasiantaeb, Shubham Chatterjee, Jeffery Dalton, and Leif Azzopardi. 2024. Trec ikat 2023: The interactive knowledge assistance track overview. *arXiv preprint arXiv:2401.01330*.

Avishek Anand, Lawrence Cavedon, Hideo Joho, Mark Sanderson, and Benno Stein. 2020. Conversational search (Dagstuhl Seminar 19461). In *Dagstuhl Reports*, volume 9. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

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

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*.

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

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *ICLR (Poster)*. OpenReview.net.

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.

Guglielmo Faggioli, Laura Dietz, Charles L. A. Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, and Henning Wachsmuth. 2023. Perspectives on large language models for relevance judgment. In *ICTIR*, pages 39–50. ACM.

Song Feng, Siva Sankalp Patel, Hui Wan, and Sachindra Joshi. 2021. Multidoc2dial: Modeling dialogues grounded in multiple documents. In *EMNLP (1)*, pages 6162–6176. Association for Computational Linguistics.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *CoRR*, abs/1911.12237.

Nam Le Hai, Thomas Gerald, Thibault Formal, Jian-Yun Nie, Benjamin Piwowarski, and

Laure Soulier. 2023. Cosplade: Contextualizing SPLADE for conversational information retrieval. In *ECIR (1)*, volume 13980 of *Lecture Notes in Computer Science*, pages 537–552. Springer.

Yu Li, Baolin Peng, Yelong Shen, Yi Mao, Lars Liden, Zhou Yu, and Jianfeng Gao. 2022. Knowledge-grounded dialogue generation with a unified knowledge representation. In *NAACL-HLT*, pages 206–218. Association for Computational Linguistics.

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

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

Yiqi Liu, Nafise Sadat Moosavi, and Chenghua Lin. 2023. Llms as narcissistic evaluators: When ego inflates evaluation scores. *CoRR*, abs/2311.09766.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. *CoRR*, abs/2301.13379.

Sean MacAvaney and Luca Soldaini. 2023. One-shot labeling for automatic relevance estimation. In *SIGIR*, pages 2230–2235. ACM.

Quinn Patwardhan and Grace Hui Yang. 2023. Sequencing matters: A generate-retrieve-generate model for building conversational agents.

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.

Filip Radlinski and Nick Craswell. 2017. A theoretical framework for conversational search. In *CHIIR*, pages 117–126.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Sina Semnani, Violet Yao, Heidi Zhang, and Monica Lam. 2023. WikiChat: Stopping the hallucination of large language model chatbots by few-shot grounding on Wikipedia. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2387–2413, Singapore. Association for Computational Linguistics.

Kurt Shuster, Mojtaba Komeili, Leonard Adolphs, Stephen Roller, Arthur Szlam, and Jason Weston. 2022a. Language models that seek for knowledge: Modular search & generation for dialogue and prompt completion. In *EMNLP (Findings)*, pages 373–393. Association for Computational Linguistics.

Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, Morteza Behrooz, William Ngan, Spencer Poff, Naman Goyal, Arthur Szlam, Y-Lan Boureau, Melanie Kambadur, and Jason Weston. 2022b. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *CoRR*, abs/2208.03188.

Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2021. Question rewriting for conversational question answering. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 355–363.

Nikos Voskarides, Dan Li, Andreas Panteli, and Pengjie Ren. 2019. Ilps at trec 2019 conversational assistant track. In *TREC*.

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*. ACM.

Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. Few-shot generative conversational query rewriting. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 1933–1936.

Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-shot conversational dense retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 829–838.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. In *ICLR*. OpenReview.net.

Hamed Zamani, Johanne R Trippas, Jeff Dalton, and Filip Radlinski. 2022. Conversational information seeking. *arXiv preprint arXiv:2201.08808*.

# A Appendix

## A.1 iKAT Benchmark

The retrieval performance of the proposed models using the benchmark of iKAT is presented at Table 3. The percentage of unjudged passages among the first 10 retrieved passages for each model is written in this table.

Table 3: The results of passage retrieval task for iKAT dataset using the iKAT benchmark. [H]: using human-rewritten queries, [G]: using answers generated by GPT-4.

| P. | A. | Model | nDCG@5 | nDCG | P@20 | Recall@20 | Recall | mAP | Judged@10 |
|---|---|---|---|---|---|---|---|---|---|
| GR | QD | GPT-4 | .2122 | .2572 | .1820 | .1041 | .3176 | .0954 | 3.42 |
| GR | AQD | GPT-4 | .4213 | .3469 | .3327 | .1778 | .3584 | .1706 | 10.00 |
| GR | AQD | GPT-4 [H] | .2175 | .2633 | .1951 | .1199 | .3304 | .0990 | 3.22 |
| GR | $AQD_A$ | GPT-4 | .2023 | .3082 | .1553 | .0900 | .4914 | .0920 | 2.77 |
| GR | QD | LLaMA | .0492 | .1002 | .0571 | .0294 | .1525 | .0220 | 1.45 |
| GR | AQD | LLaMA | .1089 | .1434 | .0966 | .0542 | .1893 | .0411 | 1.76 |
| GR | AQD | LLaMA [G] | .1432 | .1912 | .1312 | .0709 | .2546 | .0616 | 2.69 |
| GR | AQD | LLaMA [H] | .1151 | .1707 | .1147 | .0681 | .2293 | .0514 | 2.06 |
| GR | $AQD_A$ | LLaMA | .1149 | .1970 | .1162 | .0609 | .3196 | .0575 | 1.97 |
| | | | | **Baselines** | | | | | |
| RG | QR | Human | .3010 | .3538 | .2726 | .1590 | .4979 | .1476 | 5.05 |
| RG | QR | GPT-4 | .2124 | .1853 | .1921 | .1005 | .1863 | .0828 | 3.79 |
| RG | QR | LLaMA | .1176 | .1580 | .1056 | .0508 | .2441 | .0506 | 2.31 |
| RG | QR | LLaMA10 | .1466 | .0756 | .1192 | .0553 | .0553 | .0376 | 10.00 |
| RG | QR | ConvGQR | .1623 | .1518 | .1421 | .0611 | .2034 | .0551 | 10.00 |
| RG | QR | monoT5 | .2206 | .2147 | .1831 | .0812 | .3058 | .0754 | 10.00 |
| GR | AD | GPT-4 | .1441 | .2420 | .1244 | .0771 | .4041 | .0699 | 2.28 |
| GR | AD | LLaMA | .1304 | .1479 | .1395 | .0738 | .1675 | .0577 | 2.53 |
| GR | AD | InfoSense | .3109 | .2097 | .2519 | .1168 | .1862 | .1042 | 10.00 |

Table 4: Confusion matrix comparing graded relevance judgments made by TREC assessors and one-shot GPT-3.5 on TREC iKAT 2023.

| | | TREC iKAT 2023 | | | | |
|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 |
| GPT-3.5 | 0 | 8,743 | 227 | 112 | 39 | 9 |
| | 1 | 3,515 | 400 | 209 | 70 | 11 |
| | 2 | 3,189 | 668 | 450 | 189 | 47 |
| | 3 | 1,625 | 630 | 476 | 284 | 52 |
| | 4 | 620 | 414 | 369 | 284 | 52 |

### A.2 GPT-3.5 Pool

The comparison between the graded relevance scores given by TREC assessors and GPT-3.5 is provided in Table 2.

### A.3 Prompts

The prompt used for AQD and AD approach using GPT-4 is shown in Table 5. We use the same prompt for $AQD_A$ approach. The prompt used for zero-shot QD using GPT-4 model is shown in Table 6. The term *ctx* in the prompts designed for GPT-4 includes all of the previous user utterances and system responses.

For QR model the prompt shown in Table 7 is designed. This prompt is used for both LLaMA and GPT-4 models. We pass all the previous user and system interactions as *ctx* in this prompt. The prompt designed for relevance judgement using GPT-3.5 model is shown in Table 8. In this prompt the resolved utterance, relevant PTKB statements, and the canonical answer from gold data are used.

The two-shot prompt designed for LLaMAAQD approach is shown in Table 9. In this prompt the answer generated by LLaMA itself is passed as the *response*. For the prompts of LLaMA model, all the previous user utterances with the last system response are passed as context to the model.

The one-shot prompt designed for LLaMAQD approach is shown in Table 10.

Table 5: The prompt designed for AQD and AD approaches using GPT-4 as zero-shot learner.

**(1) Initial Answer Generation and (2) Query Generation in AQD approach.**

(1) # Instruction:
*I will give you a conversation between a user and a system. Also, I will give you some background information about the user. You should answer the last question of the user.*

Table 6: The prompt designed for QD approach using GPT-4 as zero-shot learner.

**Query Generation in QD approach.**

# Instruction:
*I will give you a conversation between a user and a system and some background information about the user. Imagine you want to find the answer to last user question by searching the google. You should generate the search queries that you need to search in google. Please don't generate more than 5 queries and write each query in one line.*

# Background knowledge: {*ptkb*}
# Context: {*ctx*}
# User question: {*user utterance*}

# Generated queries:

Table 7: The prompt designed for QR using GPT-4 and LLaMA models as zero-shot learner.

| **Query re-writing (QR).** |
|---|
| # Instruction:<br>*I will give you a conversation between a user and a system. Also, I will give you some background information about the user. You should rewrite the last question of the user into a self-contained query.*<br><br># Background knowledge: {*ptkb*}<br># Context: {*ctx*}<br># Please rewrite the following user question: {*user utterance*}<br># Re-written query: |

Table 8: The prompt designed for relevance judgement using GPT-3.5 as one-shot learner.

| **Relevance judgement.** |
|---|
| # Instruction:<br>*I will give you a user question and a passage, you should say to what extent the given passage is relevant for answering the question by giving an integer rate between 0-4. The relevance score lower than 2 means the document is not relevant. I will also give some background information about the user who asked this question.*<br><br># Background knowledge:<br>{*relevant ptkb statements*}<br># User question: {*resolved utterance*}<br><br># Document 1: {*canonical response*}<br># Score: 4<br><br># Document 2: {*document*}<br># Score: |

Table 9: The prompt designed for two-shot AQD using LLaMA model.

**The two-shot prompt for query generation in AQD.**

# Instruction:
*Generate the unique queries to search them in a search engine to retrieving the last response of the system to the user. (Please write each query in one line and don't generate more than 5 queries)*

# Example 1
# Background knowledge: 1: My sister is following the 'West Worl', but I don't like it, 2: Johnny Depp made the Pirates of the Caribbean excellent, 3: My friend suggested to me the 'Now you see me' movie, ...
# Context:
user: Can you tell me what the Golden Globe Awards is?
user: What is it?
user: Is it different from the Oscars?
user: What is the difference between them?
user: No, I mean Academy Awards and Golden Globe Awards.
user: What else?
system: The Hollywood Foreign Press Association, a group of 93 journalists from around 55 countries, ....
# User question: Did any of my favorite actresses win any of them?
# System response: Yes, both Jennifer Aniston and Lisa Kudrow, who you enjoyed in the Friends series, have won Golden Globe Awards. Jennifer Aniston won the Golden Globe in 2003 for Best Actress in a Television Series – Musical or Comedy for Friends. Lisa Kudrow also won this category in 1998 for the same series. Aniston's performance in The Morning Show also earned her another nomination in 2020. On the other hand, neither of the actresses have won an Academy Award as of yet.
# Generated queries:
1. Has Jennifer Aniston ever won a Golden Globe or an Academy Award?
2. Has Lisa Kudrow ever won a Golden Globe or an Academy Award?
3. List of Golden Globe winners in Best Actress in a Television Series – Musical or Comedy category for 1998 and 2003.
4. Did Jennifer Aniston win any awards for 'The Morning Show'?

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

# Example 2
# Background knowledge: 1: I am diagnosed with diabetes type 2, 2: My husband is a light drinker, 3: We have pasta twice a week! It is my favorite dish,....
# Context:
user: Can you tell me about different types of alcoholic drinks?
user: What is the difference between them?
user: No, the other category.
user: Ok. Can you compare base liquors and the third one?
user: Interesting, but I was mainly looking for ingredients and flavor! FYI, I don't drink myself.
user: How is it different from Liqueurs?
user: Why are their names so similar?
user: How about the percentage of alcohol?
system: Liquor is hard (the hardest) alcohol product made by distillation, often clocking ....
# User question: Which types are suitable for my husband?
# System response: Considering your husband is a light drinker, he might enjoy lower-alcohol content beverages such as wine, beer, or certain liqueurs. Specifically, since he enjoys seafood like salmon, white wine such as chardonnay, pinot grigio, or a light-bodied beer might be a good match. However, always remember to consume alcohol in moderation, and if there are any health concerns, consult with a physician.
# Generated queries:
1. Which alcoholic beverages are suitable for a light drinker who likes salmon?
2. What types of alcohol go well with seafood?
3. What types of alcohol are preferred for people who drink lightly?
4. What alcoholic drinks have lower alcohol content?
5. Recommendations for alcoholic beverages for light drinkers.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

# Example 3
# Background knowledge: {*ptkb*}
# Context: {*ctx*}
# User question: {*user utterance*}
# System response: {*response*}
# Generated queries:

Table 10: The prompt designed for one-shot QD using LLaMA model.

**The one-shot prompt designed for QD.**

# Instruction:
*Please generate self-contained questions that should be searched in a search engine to answer the user's LAST utterance. (Please write each query in one line and don't generate more than 5 queries)*

# Example 1
# Background knowledge: 1: My sister is following the 'West Worl', but I don't like it, 2: Johnny Depp made the Pirates of the Caribbean excellent, 3: My friend suggested to me the 'Now you see me' movie, it was fantastic, 4: I went on a biking trip last year, 5: I usually like to drink coffee in the morning, 6: I watched the proposal and enjoyed it. Ryan Reynolds is my favorite!, 7: The 'Friends' series was terrific, Jennifer Aniston and Lisa Kudrow were the best stars!

# Context:
user: Can you tell me what the Golden Globe Awards is?
user: What is it?
user: Is it different from the Oscars?
user: What is the difference between them?
user: No, I mean Academy Awards and Golden Globe Awards.
user: What else?
system: The Hollywood Foreign Press Association, a group of 93 journalists from around 55 countries, are the committee for the Globes. On the contrary, the voting body of the Academy Awards; the Academy of Motion Picture Arts and Sciences (AMPAS) consists of 6,000 voting members. The Golden Globes Award recognizes the excellence of artists in both the film and television industry in the United States as well as in other countries. However, Academy Awards only recognize the excellence of artists in their cinematic achievements, primarily in Hollywood or the American film industry. There are 25 categories for Golden Globes; 14 in motion pictures and 11 in television. At present, the Academy Awards has 24 categories. Usually, the Golden Globes ceremony is held in January of each year while the Academy Awards ceremony is held in February of each year.

# User question: Did any of my favorite actresses win any of them?

# Generated queries:

1. Has Jennifer Aniston ever won a Golden Globe or an Academy Award?
2. Has Lisa Kudrow ever won a Golden Globe or an Academy Award?
3. Did Jennifer Aniston win any awards for 'The Morning Show'?
4. List of Golden Globe winners in Best Actress in a Television Series – Musical or Comedy category for 1998 and 2003.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

# Example 2
# Background knowledge: {*ptkb*}
# Context: {*ctx*}
# User question: {*user utterance*}
# Generated queries: