NII@TREC IKAT 2024:LLM-Based Pipelines for Personalized Conversational Information Seeking

Xiao Fu
University College London
London, UK
xiao.fu.20@ucl.ac.uk

Navdeep Singh Bedi
Università della Svizzera italiana
Lugano, Switzerland
navdeep.singh.bedi@usi.ch

Praveen Acharya
Dublin City University
Dublin, Ireland
praveen.acharya2@mail.dcu.ie

Noriko Kando
National Institute of Informatics
Tokyo, Japan
Noriko.Kando@nii.ac.jp

***Abstract*—In this paper, we propose two novel pipelines—Retrieve-then-Generate (RtG) and Generate-then-Retrieve (GtR)—to enhance conversational information seeking (CIS) systems, evaluated within the TREC iKAT 2023 framework. The RtG pipeline emphasizes brevity in rewriting user utterances and generates multiple query groups to maximize the retrieval of relevant documents. This approach leads to improved recall in the final results compared to the best submission in 2023. Additionally, it incorporates a chain-of-thought methodology through a two-stage response generation process. In a zero-shot setting, the GtR pipeline introduces a hybrid approach by ensembling state-of-the-art Large Language Models (LLMs), specifically GPT-4o and Claude-3-opus. By leveraging the strengths of multiple LLMs, the GtR pipeline achieves high recall while maintaining competitive precision and ranking performance in both document retrieval and Personal Task Knowledge Base (PTKB) statement classification tasks. Our experimental results demonstrate that both pipelines significantly enhance retrieval effectiveness, offering robust solutions for future CIS systems.**[1]

I. Introduction

Information retrieval is the process of fetching documents or passages from a large corpus based on their relevance to a user's query. Traditionally, this involves calculating relevance scores using metrics such as BM25, vector similarity scores, or evaluation metrics like nDCG and P@K. Retrieval systems can operate through single-stage pipelines, where documents are directly retrieved based on these scores, or multi-stage pipelines that refine results through additional processing stages.

With the advent of conversational interfaces, information retrieval has evolved to accommodate dynamic interactions where users engage in a sequence of questions and answers. In these conversational settings, users may shift between different topics or contexts, making it essential for systems to retain and utilize contextual information from previous interactions to provide coherent and relevant responses.

The *Text REtrieval Conference (TREC)* is a series of workshops that promote research in information retrieval by providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies. Organized by the National Institute of Standards and Technology (NIST), TREC facilitates collaboration among researchers by offering standardized datasets and evaluation protocols.

Listing 1 One turn in iKAT 2023

```
{
"number": "2-1",
"title": "Visiting Doha",
"ptkb": {"1": "I like sightseeing.",
    ...},
"turns": [{
        "turn_id": 1,
        "utterance": "I have...?",
        "resolved_utterance": "...?",
        "response": "There are...",
        "ptkb_provenance": [9,...],
        "response_provenance": ["id1",...]
    },...}
```

Building on this foundation, the *TREC Interactive Knowledge Assistant Track (iKAT)* focuses on the task of personalized retrieval and answer generation within conversational interactions. The central objective of iKAT is to develop interactive knowledge assistants capable of engaging users in dynamic and personalized dialogues, offering comprehensive and contextually relevant information. Unlike previous Conversational Assistant Tracks (CAsT), which primarily considered context originating from the conversation itself, iKAT introduces a user personalization component. This component links conversations to specific personas that embody certain characteristics or preferences—such as dietary restrictions or educational interests.

To address complex information needs—for example, assisting a user in planning to visit Doha must have access to personal information and preferences, conversational capabilities to elicit and convey information effectively, and retrieval capabilities to access and integrate relevant information from various sources. Personalization is achieved by providing a Personal Textual Knowledge Base (PTKB), which contains natural language descriptions of a user's characteristics and preferences. Consequently, the same user question might require different responses depending on the personal information provided in the PTKB, leading to different conversational trajectories originating from the same initial query.

Figure 1 illustrates a sample turn in a conversation of

[1] Work done while at NII.

iKAT, where the user wants to create a visiting plan for Doha. At each turn, the agent should respond based not only on retrieved documents (*response_provenance*) but also on the user's personal preferences (*ptkb_provenance*).

The competition includes topics encompassing conversations with different users with varying personas. IKAT 2024 consists of 16 conversations. For the passage retrieval task, participants rely on a document collection comprising a subset of ClueWeb22-B [4] provided by the competition organizers.

Given the complexity of personalized retrieval and answer generation—which requires multiple reasoning, retrieval, and language capabilities—the task is divided into three subtasks to facilitate evaluation and system development: (i) PTKB Provenance, (ii) Passage Provenance, and (iii) Response Generation.

II. Tasks

Task Description

In the second year of the iKAT track, participants are provided with specific inputs at each turn of the conversation to simulate a personalized conversational assistant. These inputs include:

- **PTKB**
- **Conversation History**
- **The latest user utteranced**

As mentioned in the above section, due to the complexity of the task of creating a personalized conversational information retrieval agent and the multi-reasoning involved, this task is divided into 3 main sub-tasks:

1) **PTKB Statement (provenance) Classification** ' This is a binary classification task, in which given the context (previous conversation between the user and the system) and the current user utterance we need to find the subset of PTKB statements relevant to the system for responding to the user. This can also be an empty set.
2) **Passage (provenance) Ranking** This step includes passage retrieval and ranking in which given the context (previous conversation between the user and the system), personal information (PTKB statements), and the current user utterance, we need to retrieve and rank the relevant passages from the corpus (Clueweb22B), which can help the system to respond to the user. In this task, we can retrieve up to 1000 documents, but all these documents cannot be used to generate the response by the system due to the limitations of the LLMs, so we need to flag the documents that we used to generate the response to the user.
3) **Response Generation** This is the final task in which given the relevant PTKB statements selected in the first sub-task, the chosen passages in the second task, the context (previous conversation between the user and the system), and the current user utterances, the system needs to generate the response to the user.

III. Pipeline

The process begins by generating the indices of the passages in the Clueweb22b corpus using *Pyserini* [2], which can later help us to extract the documents from the corpus easily using sparse retrieval methods.

A. Retrieve-then-Generate (RtG)

Our first pipeline follows the classic rewrite-retrieve-rerank approach, as Figure 1 illustrates.

Utterance Rewriting: The pipeline begins by rewriting the user's original utterances using GPT-4 [3] and the conversation log, as detailed in Appendix A1. An additional instruction in the prompt—"keep only the question part with only related information and make it a clear question"—ensures that the rewritten utterances are concise and focused, facilitating their use with PTKBs in subsequent steps.

PTKB Ranking: The rewritten utterances are then employed to rank PTKBs using GPT-4, as demonstrated in Appendix A2.

Query Generation: To retrieve documents effectively, we utilize BM25. To achieve comprehensive retrieval coverage, queries are generated using GPT-4, as detailed in Appendix A3. For each PTKB, the rewritten utterance is used to generate a set of queries.

Document Retrieval and Reranking: Based on the generated queries, documents are retrieved and subsequently reranked using the rewritten utterance to enhance relevance. For each query, the top 50 hits are retrieved. This reranking employs the *cross-encoder/ms-marco-MiniLM-L-12-v2*[3].

Response Generation: Finally, we apply the chain-of-thought methodology for response generation. A preliminary response suggestion is generated before crafting the final response. This final response is produced by integrating the suggestion with the rewritten utterance, conversation history, PTKB, and candidate documents, as illustrated in Appendix A4.

B. Generate-then-Retrieve (GtR)

We employ an information retrieval and generative model based on the GtR pipeline to solve the subtasks mentioned in the previous section. The subtasks are executed sequentially, with the first and second providing components for the third. The pipeline is illustrated in Figure 2 and is explained below.

Initial Response Generation: We generate an initial response to the user's utterance using the conversation context (the dialogue between the system and the user up to the previous turn), the PTKB statements, and the current user utterance. This response is generated by prompting the above information with suitable instructions to LLMs, specifically GPT-4o and Claude-3-opus [2], as shown in Appendix A5.

PTKB Selection: Next, we retrieve relevant PTKB statements from the PTKB list using the conversation context, the PTKB statements, and the current user utterance. The selection is performed by prompting this information with

[2] https://github.com/castorini/pyserini?tab=readme-ov-file

[3] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2

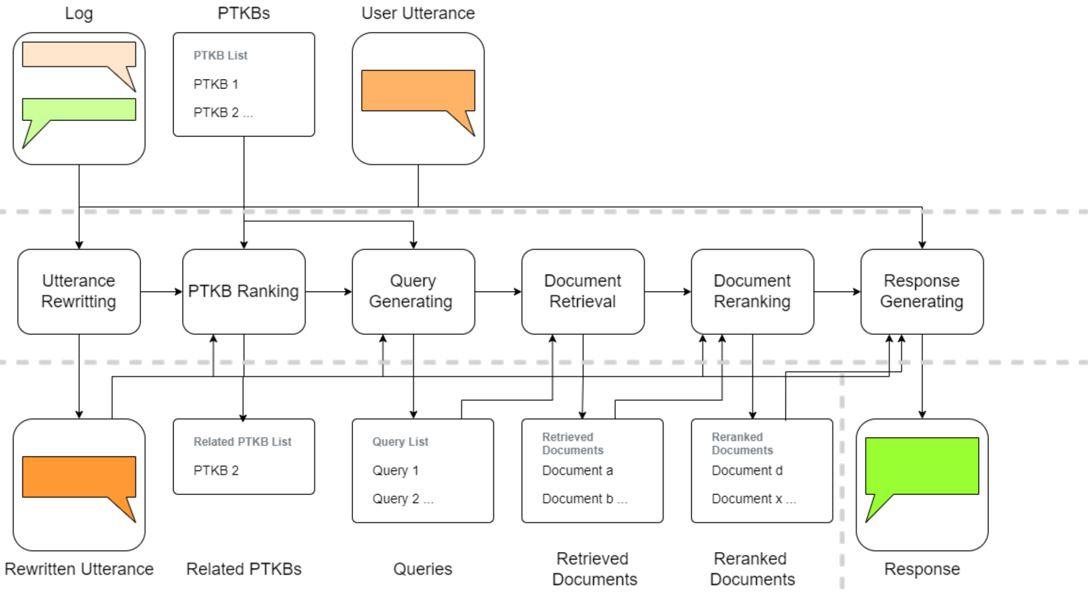


Fig. 1: Illustration of our RtG pipeline.

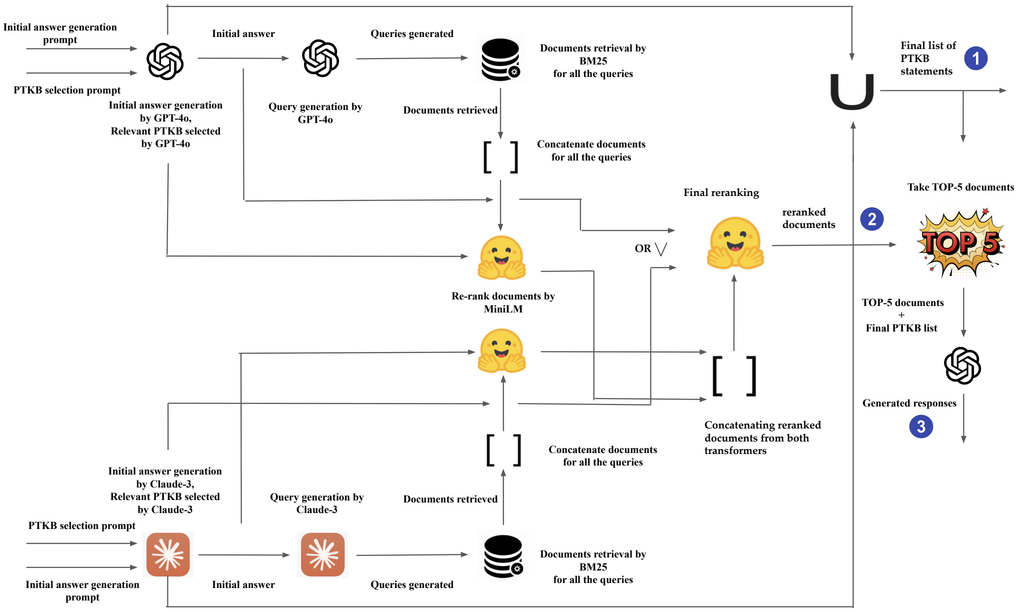

Initial answer generation prompt
PTKB selection prompt
Initial answer
Queries generated
Documents retrieval by BM25 for all the queries
Initial answer generation by GPT-4o, Relevant PTKB selected by GPT-4o
Query generation by GPT-4o
Documents retrieved
Concatenate documents for all the queries
Final reranking
OR ⋁
Re-rank documents by MiniLM
reranked documents
Final list of PTKB statements
1
2
Take TOP-5 documents
TOP 5
TOP-5 documents + Final PTKB list
Generated responses
3
Concatenating reranked documents from both transformers
Concatenate documents for all the queries
Initial answer generation by Claude-3, Relevant PTKB selected by Claude-3
Query generation by Claude-3
Documents retrieved
PTKB selection prompt
Initial answer generation prompt
Initial answer
Queries generated
Documents retrieval by BM25 for all the queries

Fig. 2: Illustration of our GtR pipeline.

suitable instructions to GPT-4o and Claude-3-opus via APIs from OpenAI and Anthropic, as detailed in Appendix A6.

Query Generation: Using the initial response generated in the first step, the LLMs are prompted to generate five queries that could be potential questions related to the initial response, as shown in Appendix A7.

Sparse Retrieval: We retrieve 300 passages per query using the sparse retrieval method implemented with the BM25 function in Pyserini. At this stage, we have 1,500 documents retrieved from the ClueWeb22B dataset for each LLM. Each document is then scored using the BM25 scoring function relative to each query.

Initial Re-ranking: For each LLM, we concatenate the results from all queries, remove duplicate documents, and rerank them using the initial generated system utterance. This reranking employs the *cross-encoder/ms-marco-MiniLM-L-6-v2*[4] from the Hugging Face library, selecting the top 1,000 documents.

Final Re-ranking: We then combine the results from both LLMs, remove duplicates, and perform a final reranking with respect to the initial response generated by the Claude-3-

[4] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2

TABLE I: Automatic evaluation of passage retrieval results of TREC iKAT 2023. Evaluation at retrieval cutoff of 1000.

Group	NDCG	NDCG@5	Recall
2023 Best	0.3479	**0.4396**	0.3456
2023 Baseline	0.0277	0.0450	0.0257
Ours RtG	**0.3492**	0.2686	**0.4991**
Ours GtR(Claude)	0.2473	0.1507	0.4006
Ours GtR(GPT)	0.2507	0.1569	0.3917
Ours GtR(Claude+GPT)	0.3087	0.2193	0.4781

TABLE II: Performance of automatic runs on the PTKB provenance task based on NIST assessment of TREC iKAT 2023.

Group	NDCG@3	P@3	Recall@3
2023 Best	**0.7254**	0.4626	0.6964
2023 Baseline	0.3434	0.2687	0.3099
Ours RtG	0.6960	0.4762	0.7037
Ours GtR(Claude)	0.6755	0.4762	0.6996
Ours GtR(GPT)	0.6910	0.4830	0.7046
Ours GtR(Claude+GPT)	0.6915	**0.4864**	**0.7166**

opus LLM using the same cross-encoder model. The top five documents are extracted from the reranked corpus.

Final PTKB Statements: To obtain the final list of PTKB statements, we take the union of the selected PTKB statements from both LLMs.

Generating Final Response: Finally, we use the top five documents, the selected PTKB statements, the conversation context, and the current user utterance to generate the system's final response using GPT-4o, as shown in Appendix A8. These steps are repeated for each user utterance, and the final results are presented in Algorithm 3 in Appendix B.

IV. Further Settings

1) Evaluation Metric: Our paper mainly focuses on the retrieval and ranking part rather than the text generation, we utilize Recall evaluation metrics to show our results for document retrieval. We also used the nDCG@5 metric to show the ranking accuracy of our results. Whereas, in the case of PTKB results we use Recall@3, Precision@3, and nDCG@3 to present our results, as for PTKB statements there are approximately 10 PTKB statements for each user, so calculating the results at a cut-off of 3 is a decent choice.

2) Dataset and Baselines: We perform our experiments on the TREC iKAT 2023 dataset explained in the Section III-B. Our model is compared with the best models of each team from the TREC iKAT 2023 submission [1].

V. Results

A. TREC IKAT 2023

This section presents the automatic evaluation of our proposed RtG and GtR pipelines on the TREC iKAT 2023 dataset. We report the performance on both the passage retrieval task and the PTKB provenance task. Tables I and II summarize the results of our systems compared to the 2023 Best and Baseline systems.

B. Passage Retrieval Results

Table I shows the evaluation of passage retrieval results at a retrieval cutoff of 1000. The evaluation metrics include nDCG, nDCG@5, and Recall.

Our **RtG pipeline** achieved an nDCG of **0.3492**, slightly surpassing the 2023 Best system's nDCG of 0.3479. Moreover, it attained the highest Recall of **0.4991**, significantly outperforming the 2023 Best system's Recall of 0.3456. This indicates that our RtG pipeline is more effective at retrieving a larger proportion of relevant documents.

However, the nDCG@5 for the RtG pipeline is 0.2686, which is lower than the 2023 Best system's nDCG@5 of **0.4396**. This suggests that while our RtG pipeline retrieves more relevant documents overall, the top-ranked documents are not as highly relevant as those retrieved by the 2023 Best system.

For the **GtR pipelines**, the combined GtR (Claude+GPT) approach achieved an nDCG of 0.3087 and a Recall of 0.4781, demonstrating competitive performance. Individually, the GtR (Claude) and GtR (GPT) pipelines achieved nDCG values of 0.2473 and 0.2507, and Recall values of 0.4006 and 0.3917, respectively. These results indicate that ensembling multiple LLMs in the GtR pipeline enhances retrieval effectiveness.

C. PTKB Provenance Results

Table II presents the performance on the PTKB provenance task based on NIST assessments. The evaluation metrics include nDCG@3, P@3, and Recall@3.

Our **RtG pipeline** achieved an nDCG@3 of 0.6960, closely approaching the 2023 Best system's nDCG@3 of **0.7254**. It attained a P@3 of 0.4762 and a Recall@3 of 0.7037, slightly surpassing the 2023 Best system's Recall@3 of 0.6964. This indicates that our RtG pipeline is effective in retrieving relevant PTKB statements with high recall.

For the **GtR pipelines**, the combined GtR (Claude+GPT) approach achieved an nDCG@3 of 0.6915, a P@3 of **0.4864**—the highest among all methods—and a Recall@3 of **0.7166**, exceeding the 2023 Best system's performance. Individually, the GtR (Claude) and GtR (GPT) pipelines achieved nDCG@3 values of 0.6755 and 0.6910, respectively, with competitive P@3 and Recall@3 scores.

Overall, both our RtG and GtR pipelines demonstrate strong performance in both the passage retrieval and PTKB provenance tasks for conversational information seeking. The RtG pipeline excels in recalling a larger number of relevant documents, which is crucial for comprehensive information retrieval systems. Although the nDCG@5 for RtG is lower than the 2023 Best system, indicating that the top-ranked documents are less relevant, the overall retrieval effectiveness is higher.

The GtR pipeline, particularly when combining outputs from multiple LLMs (Claude+GPT), shows that ensembling LLMs enhances retrieval effectiveness and PTKB statement classification. The combined GtR approach achieved higher precision and recall in the PTKB provenance task, surpassing the 2023 Best system's performance.

These results underscore the efficacy of our proposed approaches in improving document and PTKB retrieval and classification in conversational information seeking systems, offering robust solutions for future CIS systems.

D. TREC IKAT 2024

TABLE III: Performance of passage retrieval results of TREC iKAT 2024. Released by NIST.

Runs	NDCG	NDCG@5	Recall
Auto_RtG	0.4184	0.3914	0.5218
Manu_RtG	**0.5066**	**0.5055**	0.5921
Auto_GtR	0.4639	0.4190	**0.6032**

1) Passage Retrieval Results: Table III presents our best performances in retrieving documents for TREC iKAT 2024. For the RtG pipeline, we used both raw user utterances (*Auto_RtG*) and manually rewritten utterances (*Manu_RtG*) as initial inputs. For the GtR pipeline, we used only raw utterances (*Auto_GtR*).

Our results show that the *Manu_RtG* run achieved the highest performance across all metrics, with an NDCG of **0.5066**, NDCG@5 of **0.5055**, and a Recall of 0.5921. This indicates that manually rewriting the user utterances significantly improves retrieval effectiveness compared to using raw utterances. The *Auto_GtR* run also performed well, achieving an NDCG of 0.4639 and the highest Recall of **0.6032** among all runs. This suggests that the GtR pipeline is effective at retrieving a larger number of relevant documents when using raw utterances. The *Auto_RtG* run, while using raw utterances, demonstrated reasonable effectiveness but lower performance than its manually rewritten counterpart.

These results indicate that both our RtG and GtR pipelines are effective in retrieving relevant documents. The superior performance of the *Manu_RtG* run highlights the importance of the quality of initial user utterances in the overall performance. Manually rewriting utterances helps in capturing the user's intent more accurately, leading to better retrieval outcomes. Conversely, the high Recall achieved by the *Auto_GtR* run demonstrates the GtR pipeline's capability to retrieve a broader set of relevant documents, which is crucial for comprehensive information retrieval systems.

TABLE IV: Performance of PTKB ranking results of TREC iKAT 2024. Released by NIST.

Runs	NDCG@5	P@5	Recall@5
Auto_RtG	0.5244	0.3649	0.5338
Manu_RtG	**0.5249**	**0.3719**	**0.5453**
Auto_GtR	0.4953	0.3596	0.5187

2) PTKB Provenance Results: Table IV presents the performance of our PTKB ranking results for TREC iKAT 2024, as released by NIST. For the RtG pipeline, we evaluated both raw user utterances (*Auto_RtG*) and manually rewritten utterances (*Manu_RtG*) as initial inputs. For the GtR pipeline, we used only raw utterances (*Auto_GtR*). Our results show that the *Manu_RtG* run achieved the highest performance across all metrics, with an NDCG@5 of **0.5249**, Precision@5 (P@5) of **0.3719**, and Recall@5 of **0.5453**. This indicates that manually rewriting the user utterances improves PTKB ranking effectiveness, likely due to better capturing of user intent and contextual nuances.

The *Auto_RtG* run also performed well, with an NDCG@5 of 0.5244 and a Recall@5 of 0.5338, which are close to the *Manu_RtG* results. This suggests that the RtG pipeline is robust even when using raw utterances, although manual rewriting offers a slight advantage.

The *Auto_GtR* run achieved an NDCG@5 of 0.4953, a P@5 of 0.3596, and a Recall@5 of 0.5187. While these metrics are slightly lower than those of the RtG runs, they demonstrate that the GtR pipeline is still effective in PTKB ranking using raw utterances.

These findings highlight the effectiveness of both our RtG and GtR pipelines in PTKB ranking tasks. The superior performance of the *Manu_RtG* run emphasizes the value of manual utterance rewriting in enhancing retrieval performance. Overall, our approaches contribute to the advancement of conversational information seeking systems by improving PTKB statement retrieval and ranking.

VI. Conclusion

In this work, we propose two pipelines—the **RtG** and the **GtR**—for the TREC iKAT 2024 challenge.

For the **RtG pipeline**, our key features include:

1) *Prioritizing brevity in utterance rewriting*: Instead of providing detailed information, we focus on concise rewriting of user utterances for effective combination with PTKBs during query generation.
2) *Generating groups of queries and performing multiple retrievals*: In the query generation phase, we generate multiple queries and perform multiple retrievals from the corpus.

These features are designed to retrieve as many related documents as possible, resulting in improved recall in the final results compared to the best submission in 2023. Additionally, the final response generation in the RtG pipeline introduces the chain-of-thought methodology and employs a two-stage response generation process.

In the **GtR pipeline**, we present a hybrid approach that ensembles multiple LLMs for the conversational information search task. We utilize two state-of-the-art LLMs in a zero-shot setting: GPT-4o and Claude-3-opus. The task was carried out on the TREC-iKAT 2023 dataset, measuring the quality of retrieval and classification of documents to generate the results. We observe that in retrieving relevant documents, we achieved a remarkable recall value while maintaining competitive precision and ranking performance. Our approach also yielded notably good results in the classification of PTKB statements. Thus, we conclude that by combining multiple LLMs, our GtR pipeline demonstrates its utility in complex information retrieval tasks such as TREC iKAT, offering a robust solution for future CIS systems.

References

[1] Mohammad Aliannejadi, Zahra Abbasiantaeb, Shubham Chatterjee, Jeffrey Dalton, and Leif Azzopardi. Trec ikat 2023: A test collection for evaluating conversational and interactive knowledge assistants. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 819–829, New York, NY, USA, 2024. Association for Computing Machinery.

[2] Anthropic. The claude 3 model family: Opus, sonnet, haiku. 2024.

[3] OpenAI and Others. Gpt-4 technical report, 2024.

[4] Arnold Overwijk, Chenyan Xiong, and Jamie Callan. Clueweb22: 10 billion web documents with rich information. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 3360–3362, New York, NY, USA, 2022. Association for Computing Machinery.

Appendix

A. Prompts

1) Prompt For RtG Rewriting utterance:

Prompt

Instruction: Rewrite the following utterance to keep only the question part with only related information and make it a clear question: {User utterance}
Context: {context}
User utterance: u_t
Conversational context: $\{(u_1, s_1), ..., (u_{t-1}, s_{t-1})\}$

2) Prompt For RtG Ranking PTKB:

Prompt

Instruction: You are an expert, and a user is asking: {Rewritten utterance} Which of the following personal statements of this user should be considered if you want to answer:{ptkb} write the index of your answer as a JSON list
Rewritten utterance: u_r
PTKB: $\{ptkb_1, ..., ptkb_n\}$

3) Prompt For RtG Query Generating:

Prompt

Instruction: Write a list of queries to retrieve documents that respond to the question from a BM25 searcher: '{Single PTKB + Rewritten Utterance}'. Reply with a list in JSON only, where each element is a string.
Single PTKB + Rewritten Utterance: $ptkb_n + u_r$

4) Prompt For RtG Response Generating:

Prompt

Instruction: given context:[{context}], what elements are necessery if I want to answer the query: "{Rewritten Utterance}" If that question is from my friend, is there any key preference I should know from my friend to address the query? make this within 300 words
Conversational context: $\{(u_1, s_1), ..., (u_{t-1}, s_{t-1})\}$
Rewritten Utterance: u_r

Output

Suggestions: $suggestion$

Prompt

Instruction: Suggestions (If I have): [{suggestion}] This is what I got from internet (if I got):{documents}. And here are some user's statements (if my friend have): {ptkb}. Context:[{context}] Query: {Rewritten Utterance} Pick useful information fron these text and write a response following the suggestions, in conversation style, friendly and no more than 250 words, act as an expert to reply this query.
Suggestions: $suggestion$
PTKB: $\{ptkb_1, ..., ptkb_n\}$
Context: $\{(u_1, s_1), ..., (u_{t-1}, s_{t-1})\}$
Rewritten Utterance: u_r
Documents: $d_1, d_2, ..., d_5$

5) Prompt For GtR Initial Response Generation:

Prompt

Instruction: I will give you a conversation between a user and a system. Also, I will give you some background information about the user as the PKTB statements listed below. You should answer the last utterance of the user given in the Conversation below. Please remember that your answer to the last question of the user shouldn't be more than 200 words.
PTKB: $\{ptkb_1, ..., ptkb_n\}$
Conversational context: $\{(u_1, s_1), ..., (u_{t-1}, s_{t-1})\}$
Current user utterance: u_t

Output

Initial system response: s'_t

6) Prompt For GtR PTKB Selection:

Prompt

Instruction: I will give you a conversation between a user and a system. Also, I will give you some background information about the user under PTKB. You should select the relevant background information about the user and rank them in order. Please remember that your answer should contain the relevant background information only from the PTKB given also list the PTKB statements with proper numbering and don't write any sentence before or after it, write just the statements selected.
PTKB: $\{ptkb_1, ..., ptkb_n\}$
Conversational context: $\{(u_1, s_1), ..., (u_{t-1}, s_{t-1})\}$
Current user utterance: u_t

Output

Selected PTKB statements: $ptkb_{i_1}, ..., ptkb_{i_k}$

7) Prompt For GtR Query Generation:

Prompt

Instruction: Can you generate the unique queries that can be used for retrieving your previous answer to the user? (Please write each query in one line and don't generate more than 5 queries).
Previous utterance: u_t

Output

Generated queries: $q_1, ..., q_5$

8) Prompt For GtR Generating Final Response:

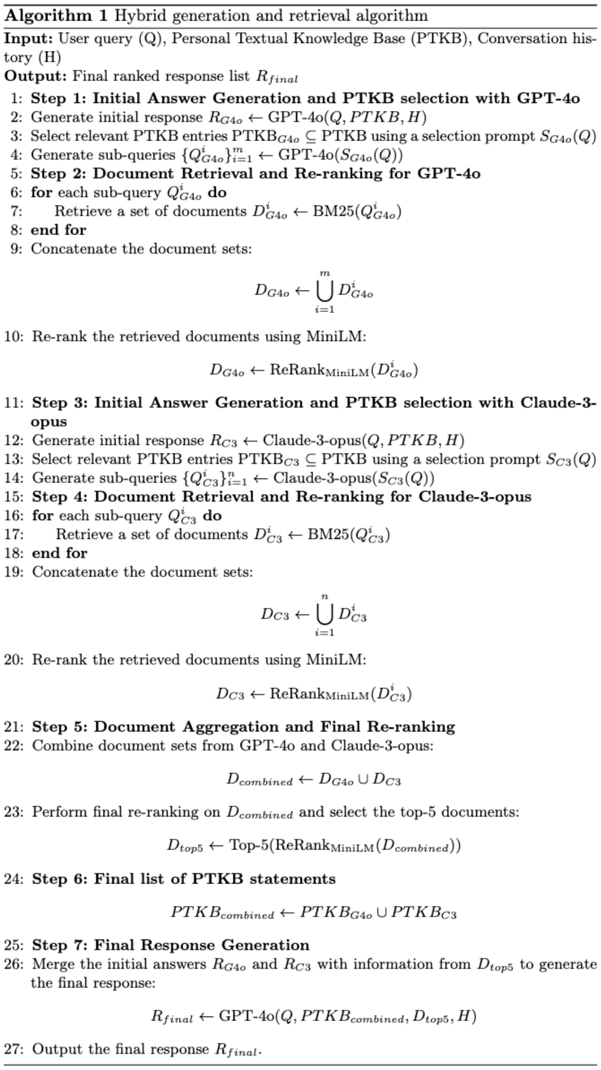
Prompt

Instruction: I will give you a conversation between a user and a system. Also, I will give you some background information about the user as PTKB statements. You should answer the last utterance of the user by providing a summary of the relevant parts of the documents given below. Please remember that your answer should not be more than 200 words.
PTKB: $\{ptkb_{i_1}, ptkb_{j_1}, ..., ptkb_{i_k}, ptkb_{j_m}\}$
Conversational context: $\{(u_1, s_1), ..., (u_{t-1}, s_{t-1})\}$
Relevant documents: $d_1, d_2, ..., d_5$
Current user utterance: u_t

Output

Final system response: s_t

B. Flow of GtR



```
**Algorithm 1** Hybrid generation and retrieval algorithm
**Input:** User query (Q), Personal Textual Knowledge Base (PTKB), Conversation history (H)
**Output:** Final ranked response list $R_{final}$
**Step 1: Initial Answer Generation and PTKB selection with GPT-4o**
Generate initial response $R_{G4o} \leftarrow \text{GPT-4o}(Q, PTKB, H)$
Select relevant PTKB entries $\text{PTKB}_{G4o} \subseteq \text{PTKB}$ using a selection prompt $S_{G4o}(Q)$
Generate sub-queries $\{Q^i_{G4o}\}^m_{i=1} \leftarrow \text{GPT-4o}(S_{G4o}(Q))$
**Step 2: Document Retrieval and Re-ranking for GPT-4o**
**for** each sub-query $Q^i_{G4o}$ **do**
    Retrieve a set of documents $D^i_{G4o} \leftarrow \text{BM25}(Q^i_{G4o})$
**end for**
Concatenate the document sets:
    $D_{G4o} \leftarrow \bigcup_{i=1}^{m} D^i_{G4o}$
Re-rank the retrieved documents using MiniLM:
    $D_{G4o} \leftarrow \text{ReRank}_{\text{MiniLM}}(D^i_{G4o})$
**Step 3: Initial Answer Generation and PTKB selection with Claude-3-opus**
Generate initial response $R_{C3} \leftarrow \text{Claude-3-opus}(Q, PTKB, H)$
Select relevant PTKB entries $\text{PTKB}_{C3} \subseteq \text{PTKB}$ using a selection prompt $S_{C3}(Q)$
Generate sub-queries $\{Q^i_{C3}\}^n_{i=1} \leftarrow \text{Claude-3-opus}(S_{C3}(Q))$
**Step 4: Document Retrieval and Re-ranking for Claude-3-opus**
**for** each sub-query $Q^i_{C3}$ **do**
    Retrieve a set of documents $D^i_{C3} \leftarrow \text{BM25}(Q^i_{C3})$
**end for**
Concatenate the document sets:
    $D_{C3} \leftarrow \bigcup_{i=1}^{n} D^i_{C3}$
Re-rank the retrieved documents using MiniLM:
    $D_{C3} \leftarrow \text{ReRank}_{\text{MiniLM}}(D^i_{C3})$
**Step 5: Document Aggregation and Final Re-ranking**
Combine document sets from GPT-4o and Claude-3-opus:
    $D_{combined} \leftarrow D_{G4o} \cup D_{C3}$
Perform final re-ranking on $D_{combined}$ and select the top-5 documents:
    $D_{top5} \leftarrow \text{Top-5}(\text{ReRank}_{\text{MiniLM}}(D_{combined}))$
**Step 6: Final list of PTKB statements**
    $PTKB_{combined} \leftarrow PTKB_{G4o} \cup PTKB_{C3}$
**Step 7: Final Response Generation**
Merge the initial answers $R_{G4o}$ and $R_{C3}$ with information from $D_{top5}$ to generate the final response:
    $R_{final} \leftarrow \text{GPT-4o}(Q, PTKB_{combined}, D_{top5}, H)$
Output the final response $R_{final}$.
```

Fig. 3: Illustration of GtR.