University of Tsukuba Team at the TREC 2023 Interactive Knowledge Assistance Track

LINGZHEN ZHENG, Graduate School of Comprehensive Human Sciences, University of Tsukuba, Japan
KAIYU YANG, Graduate School of Comprehensive Human Sciences, University of Tsukuba, Japan
HAITAO YU, Institute of Library, Information and Media Science, University of Tsukuba, Japan
SUMIO FUJITA, LY Research, LY Corporation, Japan
HIDEO JOHO, Institute of Library, Information and Media Science, University of Tsukuba, Japan

In this paper, we present our approach employed in the four automatic submission runs for the TREC 2023 Interactive Knowledge Assistance Track. This track comprises three subtasks: passage ranking, response generation, and Personal Text Knowledge Base (PTKB) statement ranking. Our comprehensive multi-stage pipeline for this task encompasses *query rewriting*, *PTKB statement ranking*, *passage retrieval and re-ranking*, and *response generation*. In particular, we employed fine-tuned pre-trained T5-CANARD for query rewriting, a combination of BERT, RankGPT, and MonoT5 for PTKB statement ranking, and Large Language Models (LLMs), RankGPT, and MonoT5 separately for passage re-ranking in four submission runs. For response generation, we adopted "mrm8488/t5-base-finetuned-summarize-news" from HuggingFace, which is a Text-to-Text Transfer Transformer (T5) based model that specially fine-tuned for summarization tasks.

1 INTRODUCTION

The TREC Interactive Knowledge Assistance Track (iKAT) builds upon the success of the Conversational Assistance Track (CAsT) and focuses on developing collaborative conversational agents that personalize responses based on user interactions. In its fourth year, CAsT introduced mixed initiatives, like clarifications and suggestions, to create dynamic multi-path, multi-turn conversations for each topic. iKAT extends this approach, emphasizing support for multi-path, multi-turn, multi-perspective dialogues. It aims to enhance system understanding of user knowledge and needs, adapting responses in real-time based on evolving context, personas, and user backgrounds. In short, iKAT advances research in conversational agents to provide more personalized and context-aware information-seeking experiences. To delve deeper into this task, we focus on the following three research questions:

- RQ1: Does incorporating PTKB statements alongside rewritten utterances in multi-turn conversational search improve passage ranking?
- RQ2: Are joint LLMs more effective in passage re-ranking compared to traditional pre-trained re-rankers?
- RQ3: Considering the computational cost and the impact on result quality when using LLMs, is it justifiable to employ LLMs for ranking tasks?

Authors' addresses: Lingzhen Zheng, s2221686@u.tsukuba.ac.jp, Graduate School of Comprehensive Human Sciences, University of Tsukuba, Japan; KAIYU YANG, s2321730@u.tsukuba.ac.jp, Graduate School of Comprehensive Human Sciences, University of Tsukuba, Japan; HAITAO YU, yuhaitao@slis.tsukuba.ac.jp, Institute of Library, Information and Media Science, University of Tsukuba, Japan; Sumio Fujita, sufujita@lycorp.co.jp, LY Research, LY Corporation, Japan; Hideo Joho, hideo@slis.tsukuba.ac.jp, Institute of Library, Information and Media Science, University of Tsukuba, Japan.

This paper outlines our approach and results of the TREC 2023 iKAT task. Our approach can be summarized as a robust four-stage pipeline, encompassing the following key components: (1) query rewriting, (2) PTKB statement ranking, (3) passage ranking, and (4) response generation. In the following section, we will provide a brief overview of each stage of our approach.

2 METHODOLOGY

In this section, we describe our four-stage pipeline for conversational search task, as depicted in Figure 1. Subsequently, we provide a detailed description of each of the four stages.

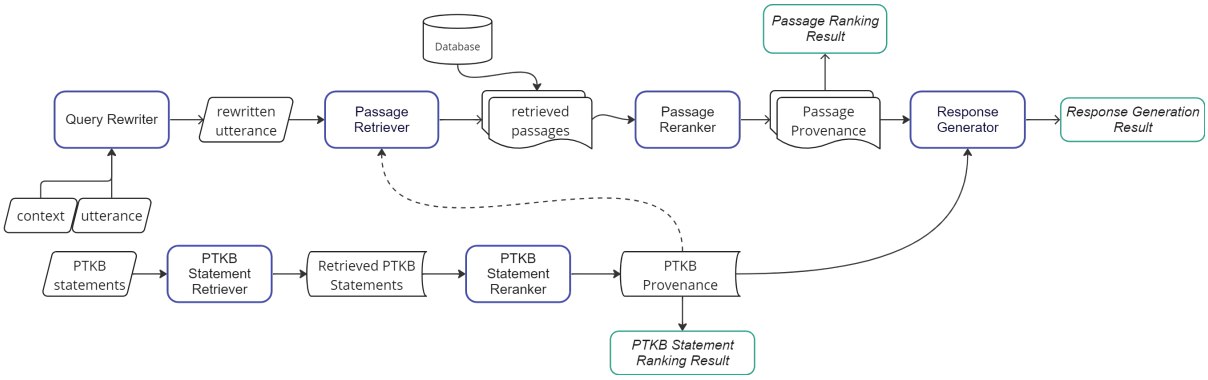

Fig. 1. Our iKAT submission overall framework.

2.1 Query Rewriting

The main objective of query rewriting is to rewrite the current turn's utterance with the context provided by previous turns' utterances and responses[6]. Ideally, this process should capture all contextual information from prior user-system interactions to effectively address any ambiguities or omissions underlying the current turn's utterance.

In our approach, we utilize a T5-based model that has been fine-tuned on CANARD[4] for query rewriting[1]. To work within length constraints, we concatenate up to the three preceding turns' user utterances and system responses as context, along with the current turn's utterance as input to the rewriter. We use u_i to denote the user utterance in the current ith conversation turn, $u_i^{'}$ represents the rewritten utterance in the current turn, and $C_i = \{u_k, r_k | (i-3) \leq k < i\}$ represents the utterances and responses from the previous three turns, encapsulating the context of up to three preceding rounds of conversation. The formula for the query rewriting process is represented as follows:

$$u_i^{'} = QR(u_i; C_i) \tag{1}$$

For fine-tuning the T5-CANARD model, we use data from the CAsT 2022 evaluation topics[2]. Subsequently, we employ the fine-tuned model to rewrite utterances within the TREC iKAT 2023 test topics.

[1]https://huggingface.co/castorini/t5-base-canard

[2]We used the evaluation topics in 2022_automatic_evaluation_topics_tree_v1.0.json from CAsT 2022 dataset.

This step enables us to automatically generate rewritten utterances for each turn in the conversation, a critical component in our subsequent processes. The rewritten queries are anticipated to encompass all essential information, better reflect user intent, and facilitate ad hoc search[6].

2.2 PTKB Statement Ranking

The second stage of our pipeline is as shown in Figure 2, which involves determining the relevance of PTKB (Personal Text Knowledge Base) statements to the current turn in the conversation. We utilized an ensemble learning method in this stage. Given that there are multiple PTKB statements associated with each "user" in a conversation, we first employ a pre-trained BERT model by HuggingFace[3]. This model classifies whether a PTKB statement is relevant to the rewritten utterance of the current turn, assigning "true" or "false" labels accordingly. After BERT's predictions, we get a ranked list of PTKB statements with their respective labels.

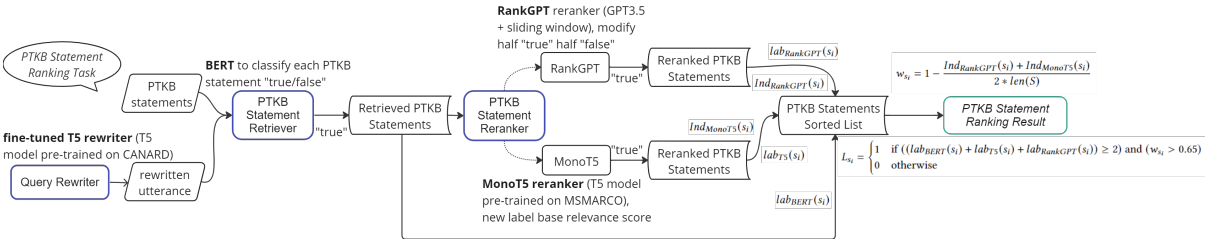

Fig. 2. PTKB Statement Ranking Pipeline.

Subsequently, we introduce a combined method to conduct a re-ranking and selection process. Both MonoT5[4] and RankGPT[5] are employed to independently re-rank and re-classify the "true" PTKB statements. To determine a final ranking score, we consider the positional weights assigned to each PTKB statement s labeled as "true" as determined by MonoT5 and RankGPT. We utilized $Ind_{RankGPT}(s_i)$ and $Ind_{MonoT5}(s_i)$ to represent each statement's two ranking indexes predicted by the two models, and $len(S)$ represent the length of PTKB statement list in each conversation. The linear interpolation formula of ith statement's average weight w_{s_i} is presented as follows:

$$w_{s_i} = 1 - \frac{Ind_{RankGPT}(s_i) + Ind_{MonoT5}(s_i)}{2 * len(S)} \tag{2}$$

We retain only those statements that have more than two "true" labels among the three predicted labels $lab_{BERT}(s_i)$, $lab_{T5}(s_i)$, and $lab_{RankGPT}(s_i)$, while at the same time having an average weight exceeding 0.65. This combination of conditions determines the final label L_{s_i} as 'true'.

$$L_{s_i} = \begin{cases} 1 & \text{if } ((lab_{BERT}(s_i) + lab_{T5}(s_i) + lab_{RankGPT}(s_i)) \geq 2) \text{ and } (w_{s_i} > 0.65) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

[3]https://huggingface.co/bert-base-uncased
[4]https://github.com/castorini/pygaggle#a-simple-reranking-example
[5]https://github.com/sunnweiwei/RankGPT

The final ranked PTKB statement provenance lists consist of the "true" statements, with "final_score"s, which are the average weights, arranged in descending order. This process ensures that only the most relevant statements are considered for subsequent stages of our approach.

2.3 Passage Ranking

In the passage ranking subtask, we drew from the experience of previous TREC CAsT competitions and adopted a retrieve-and-rerank approach. Specifically, we utilized the LuceneSearcher provided by Pyserini, along with the passage indexes supplied by the iKAT organizers, for passage retrieval in all four of our submission runs. This process involved retrieving a list of the top 1000 passages for each conversation turn's rewritten utterance.

Subsequently, we performed re-ranking on the top 5 retrieved passages using various models. We will introduce our submitted runs in the following paragraph. It's worth noting that all four of our methods are based on zero-shot learning approaches.

The main differences of our submission runs can be found in Table 1. We employed three passage re-ranking models and submitted four runs:

- Run_1: As illustrated in Figure 3, once we obtain the rewritten utterance and the two most relevant PTKB statements from the preceding stages, we concatenate them to serve as input for BM25 to retrieve 1000 passages for current conversation turn. Subsequently, we employ multiple LLMs, i.e., "stabilityai/stablelm-tuned-alpha-7b", "eachadea/vicuna-13b-1.1", "jondurbin/airoboros-7b", "TheBloke/koala-13B-HF"[6], to re-rank the top five passages retrieved during each turn. To adhere to the input length limitations of the LLMs, we instruct each LLM to perform a pair-wise comparison between every two passages within the top five retrieved passages. For each pair-wise comparison, we identify the passage ID that is deemed more relevant based on the LLMs' responses, and we tally the number of occurrences for each passage. Essentially, this process resembles a ranking vote facilitated by four LLMs. The final ranking of the top five passages is determined by the score, ordered from highest to lowest. It's important to note that, in this run, both the passage retrieval and re-ranking stages take into consideration the first two relevant PTKB statements generated by our automated runs.

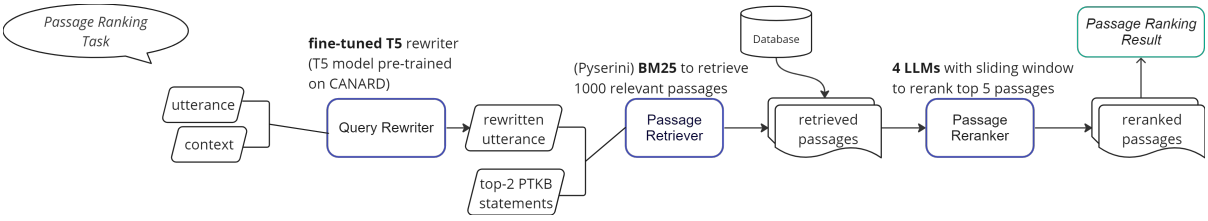

Fig. 3. Our Passage Ranking Pipeline for submission run 1.

- Run_2: Similar to Run 1, we used the same LLMs for passage re-ranking. However, in this run, neither the passage retrieval nor the re-ranking stages considered relevant PTKB statements. The ranking results simply rely on our rewritten utterance for each turn.

[6]We downloaded the pre-trained large language models from https://github.com/yuchenlin/LLM-Blender.

- Run_3: In this run, we utilized MonoT5 from pygaggle for the passage re-ranking process. Again, both the passage retrieval and re-ranking stages considered the top two relevant PTKB statements from our automatic PTKB statement ranking runs in each turn.
- Run_4: For this run, we employed a combination of the GPT-3.5 API with prompts and a sliding window approach to conduct the re-ranking process. Similar to Run 1, both the passage retrieval and re-ranking stages considered the top two relevant PTKB statements from our automatic PTKB statement ranking runs in each turn.

Submissions	Passage re-ranking method	Input of passage retrieval and re-ranking
Run_1	LLMs	With first two relevant PTKB statements
Run_2	LLMs	Without relevant PTKB statements
Run_3	MonoT5	With first two relevant PTKB statements
Run_4	RankGPT	With first two relevant PTKB statements

Table 1. Characteristics of Four Submission Runs

2.4 Response Generation

The fourth and final stage of our pipeline is response generation. We utilized a T5-based model provided by HuggingFace[7] for generating a summarization from specific input. This model is based on T5[3] architecture, and specially fine-tuned for summarization tasks.

As illustrated in Figure 4, for each turn in the conversation, we fed the model with the top-ranked passage from the passage provenance list and the top-2 PTKB statements from the PTKB provenance list as input. This stage completes our comprehensive approach, producing responses that are informed by the most relevant knowledge sources identified throughout the earlier stages of the pipeline.

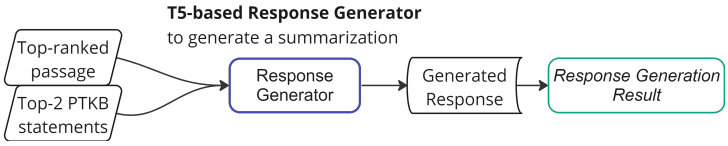

Fig. 4. Our Response Generation Pipeline.

3 RESULTS

In this section, we present the official evaluation results of our submitted runs, including the medians and dream_scores for this task. We also performed statistics and comparison of some data on the evaluation results of our four runs. As shown in Tables 2 and 3, we have highlighted the best scores in **bold font** and the second-best scores with underlines for each evaluation metric among our submitted runs.

[7]https://huggingface.co/mrm8488/t5-base-finetuned-summarize-news

3.1 Passage Ranking Evaluation Results

Passage_Run	success_1	ndcg_cut_5	ndcg_cut_10
Median	0.2074	0.1230	0.1266
Run_1	0.2727	0.1509	0.1484
Run_2	0.2614	**0.2020**	**0.2018**
Run_3	0.2607	0.1499	0.1475
Run_4	**0.2898**	0.1510	0.1470

Table 2. Passage Ranking Evaluation Results of Our Submitted Four Runs on TREC 2023 iKAT

From Table 2, we observe that all our submitted runs outperform the medians in terms of the evaluation scores for success_1, ndcg_cut_5, and ndcg_cut_10. Upon closer examination, we observe that the runs in which we did not consider the effects of relevant PTKB statements in passage retrieval and re-ranking (Run_2) achieved higher ndcg_cut scores. In contrast, the submitted runs that considered the effects of relevant PTKB statements in our framework (Run_1, Run_3, Run_4) achieved higher success_1 scores.

Passage_Run		success_1	ndcg_cut_5	ndcg_cut_10
Run_1	Better than median	23	52	53
	Worse than median	12	45	50
	Equal to best score (best!=median)	23	3	2
	Equal to worst score (worst!=median)	12	31	26
Run_2	Better than median	19	**70**	**70**
	Worse than median	11	40	43
	Equal to best score (best!=median)	19	**9**	**9**
	Equal to worst score (worst!=median)	11	21	17
Run_3	Better than median	21	51	51
	Worse than median	12	46	52
	Equal to best score (best!=median)	21	2	2
	Equal to worst score (worst!=median)	12	31	26
Run_4	Better than median	**26**	53	50
	Worse than median	12	45	52
	Equal to best score (best!=median)	**26**	3	3
	Equal to worst score (worst!=median)	12	30	26

Table 3. Passage Ranking Statistics Results of Our Submitted Runs' Evaluation Results

Table 3 presents statistical data for some metrics across our four submitted runs. This data includes the number of occurrences in each run's 176 predicted results where the success_1, ndcg_cut_5, and

ndcg_cut_10 scores are better than median, worse than median, equal to the best score, and equal to the worst score. From this table, we observe that the run (Run_2) which did not consider the impact of relevant PTKB statements on passage retrieval or re-ranking and utilized the LLM pair-wise comparison method for passage re-ranking achieved the best performance in terms of the ndcg_cut score. On the other hand, the run (Run_4) which considered the influence of relevant PTKB statements in the process and used only RankGPT as a passage re-ranker achieved the highest success_1 score.

We also noticed that the differences in ndcg_cut_5 and ndcg_cut_10 scores among our submitted runs were relatively small and did not reach high scores. We suspect that this might be due to our decision to re-rank only the top five passages retrieved from the same passage retriever, leading to minimal variations in ndcg_cut_5 and ndcg_cut_10 scores.

3.2 PTKB Statement Ranking Evaluation Results

PTKB_Run	ndcg	map	recip_rank
Median	0.61	0.54	0.73
Our_run	**0.67**	**0.61**	**0.75**

Table 4. PTKB Statement Ranking Evaluation Results of Our Submitted Runs

In this subsection, we present the evaluation results of our submitted runs in the PTKB statement ranking subtask. Table 4 provides the specific performance metrics, including NDCG, MAP, and Recip_Rank, for our submitted runs compared to the median score. Our run demonstrates notable improvements, achieving a higher NDCG, MAP, and Reciprocal Rank compared to the median.

PTKB_Run	ndcg	map	recip_rank
Better than or equal to median	70	70	77
Worse than or equal to median	61	60	71
Only worse than median	24	24	17
Equal to best score	38	38	62
Equal to worst score	15	14	14

Table 5. Statistics Results of Our PTKB Statement Ranking Evaluation Results

Table 5 presents statistical results, categorizing our PTKB statement ranking evaluation runs based on their performance relative to the median. It shows the distribution of runs that are better than or equal to the median, worse than or equal to the median, only worse than the median, equal to the best score, and equal to the worst score. We observed that our method showed outstanding results on metrics Equal to best score, but there are also a small number of results that get scores equal to the worst results. In the subsequent section, we will attempt to discuss these topics in depth.

Evaluation	PTKB_Run	Metrics			
		MRR	nDCG@3	P@3	Recall@3
NIST assessment	Baseline_Run	0.3844	0.3434	0.2687	0.3099
	Our_Run	0.7112	0.6594	0.4184	0.6213
iKAT organizer's assessment	Baseline_Run	0.3438	0.3200	0.1905	0.3720
	Our_Run	0.6890	0.6370	0.3512	0.6903

Table 6. More PTKB Statement Ranking Evaluation Results of Our Submitted Runs

Finally, Table 6 extends the evaluation to include metrics such as NDCG@3, P@3, Recall@3, and MRR in the context of NIST and iKAT organizer assessments. We compare the results of our runs with the baseline runs, highlighting the improvement in key metrics. Our runs demonstrate leading performance on this subtask, and achieved the highest score in MRR metric in iKAT organizer assessment among all submitted runs, showcasing the effectiveness of our approach in PTKB statement ranking.

We also received the official GPT-4 evaluation results for response generation task. Among the two evaluation metrics, Groundedness and Naturalness, our submissions obtained 0.67 (49/24) and 2.9178 respectively. Because this subtask lacks of human assessment, we will not discuss it much here.

These evaluation results underscore the success of our method in enhancing the relevance and ranking of PTKB statements, as demonstrated across various metrics and assessments.

3.3 Research Question Discussion

The results in above subsection help to answer the research questions stated in the introduction:

- RQ1: Based on the results presented in Table 2 and 3, we observe that among all submitted runs for this task, our approaches that considers both rewritten utterance and relevant PTKB statements, performs better in terms of the success_1 evaluation metric. However, they do not outperform the runs that solely focus on rewritten utterance in terms of ndcg_cut scores. We speculate that this performance difference is closely related to the input content during passage retrieval. We believe that considering PTKB statements in these processes is essential and beneficial, but it may require a designed allocation of attention to the two types of input texts. This is because the rewritten utterance and the response generated in the current turn of the multi-turn conversation are most highly relevant. The model might suffer from unsatisfying results due to an unreasonable attention distribution among the PTKB statements.
- RQ2: Through our statistical analysis of the four runs we submitted, it is evident that the runs utilizing LLMs (Run_1, 2, 4) clearly outperform the run that uses traditional pre-trained T5-based re-ranking models (Run_3). Notably, Run_2, which leverages joint LLMs, achieves the best performance in the ndcg_cut evaluation metric. Thus, we conclude that employing joint LLMs in the passage re-ranking task leads to superior results.
- RQ3: It is important to note that our experiments involving LLMs incur substantial computational costs, especially in the case of the joint LLMs task, which also required more time to execute. This limited our ability to perform in-depth analysis and hyperparameter tuning before submission.

Based on the evaluation results, we observe that the performance of using RankGPT alone for passage re-ranking is quite comparable to using joint LLMs for the same task. However, joint LLMs show a superior performance in more extensive ranking scenarios such as ndcg_cut_5 and ndcg_cut_10. Consequently, we acknowledge the need to further refine our approach. Nevertheless, the direction of employing LLMs for enhancing ranking results remains promising.

4 CONCLUSION

In this paper, we have introduced our four-stage pipeline for generating system responses while considering relevant PTKB statements in a multi-turn conversational information-seeking task. Our participation in the TREC iKAT has provided us with a deeper understanding of the immense potential of utilizing large language models in multi-turn conversational information-seeking tasks. We have observed that the use of pre-trained large language models consistently yields prediction performance at a significantly higher level than the medians of all submitted runs in this track across the three provided evaluation metrics.

Furthermore, our second approach, which did not take into account the effects of relevant PTKB statements, achieved notably higher scores in NDCG_cut_5 and NDCG_cut_10 compared to the other three runs we submitted. We suspect that this difference may be attributed to the passage ranking task, where simply concatenating rewritten utterances with related PTKB statements as input to the passage retrieval model might influence the initial retrieval judgment.

For our future work, we plan to explore the potential for improved performance in multi-turn conversational information-seeking by fine-tuning prompts and experimenting with few-shot methods using pre-trained large language models.

REFERENCES

[1] Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. "LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion." arXiv preprint arXiv:2306.02561 (2023).

[2] Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. "Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agent." arXiv preprint arXiv:2304.09542 (2023). Retrieved 2023 from https://arxiv.org/abs/2304.09542

[3] 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." The Journal of Machine Learning Research 21, no. 1 (2020): 5485-5551.

[4] 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.

[5] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 708–718, Online. Association for Computational Linguistics.

[6] 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). Association for Computing Machinery, New York, NY, USA, 1933–1936.