TREC iKAT 2024: The Interactive Knowledge Assistance Track Overview

Mohammad Aliannejadi
University of Amsterdam
Amsterdam, The Netherlands
m.aliannejadi@uva.nl

Zahra Abbasiantaeb
University of Amsterdam
Amsterdam, The Netherlands
z.abbasiantaeb@uva.nl

Simon Lupart
University of Amsterdam
Amsterdam, The Netherlands
s.c.lupart@uva.nl

Shubham Chatterjee
University of Edinburgh
Edinburgh, Scotland, UK
shubham.chatterjee@ed.ac.uk

Jeffrey Dalton
University of Edinburgh
Edinburgh, Scotland, UK
jeff.dalton@ed.ac.uk

Leif Azzopardi
University of Strathclyde
Glasgow, Scotland, UK
leif.azzopardi@strath.ac.uk

ABSTRACT

Conversational information seeking has evolved rapidly in the last few years with the development of large language models (LLMs) providing the basis for interpreting and responding in a naturalistic manner to user requests. iKAT emphasizes the creation and research of conversational search agents that adapt responses based on the user's prior interactions and present context, maintaining a long-term memory of user-system interactions. This means that the same question might yield varied answers, contingent on the user's profile and preferences. The challenge lies in enabling conversational search agents (CSA) to incorporate personalized context to guide users through the relevant information effectively. iKAT's second year attracted seven teams and a total of 31 runs. Most of the runs leveraged LLMs in their pipelines with some LLMs to do a single query rewrite, while others leveraged LLMs to do multiple query rewrites.

1 INTRODUCTION

Conversational information seeking stands as a pivotal research area with significant contributions from previous works [3, 8]. The TREC Interactive Knowledge Assistance Track (iKAT) builds on the foundational work of the TREC Conversational Assistance Track (CAsT) [7]. However, iKAT distinctively emphasizes the creation and research of conversational search agents that adapt responses based on the user's prior interactions and present context. This means that the same question might yield varied answers, contingent on the user's profile and preferences. Consider a scenario where a user is inquiring about alternatives to cow's milk. Three personas in Figure 1 can illustrate this:

- (A) Alice is vegan and prefers to have a type of milk that is low in sugar.
- (B) Bob is a vegan who is deeply concerned about the environment.
- (C) Christina has been recently diagnosed with diabetes, has a nut allergy, and is lactose intolerant.
-

Given Alice, Bob, and Christina's personas, their conversation with the system would evolve and develop in very different ways. This is because what is relevant to Alice may not necessarily be relevant to Bob or Christina, and vice versa. Consequently, by the end of their conversation, what they have learned about, what they have understood, and what they have decided regarding milk alternatives would vary, reflecting their personalized contexts. A detailed concrete example on the topic of "finding a university" is shown in Figure 2.

The challenge lies in enabling conversational search agents (CSA) to incorporate this personalized context to guide users through the relevant information effectively. iKAT also emphasizes decisional search tasks [10], where users sift through data and information to weigh up options in order to reach a conclusion or perform an action. These tasks, prevalent in everyday information-seeking decisions – be it related to travel, health, or shopping – often revolve around a subset of high-level information operators where queries or questions about the information space include: finding options, comparing options, and identifying the pros and cons of options. Given the different personas and their information need (expressed through the sequence of questions), diverse conversation trajectories will arise — because the answers to these similar queries will be very different.

In iKAT's debut year [1, 2], we decided to emphasize these tailored information needs by accounting for a person's knowledge, objectives, tastes, and limitations. To represent their personas, we used a Personal Text Knowledge Base (PTKB) to encapsulate

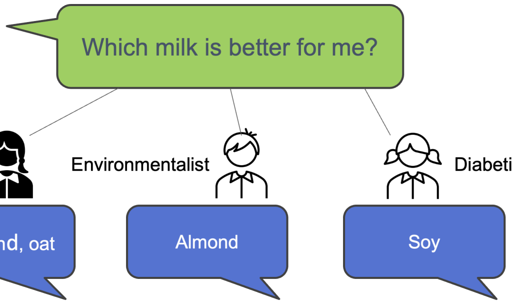

Figure 1: Example outcomes given a conversation on alternatives to cow's milk with three different personas.

TREC'23, November 2023,

both the task contexts and user specifics. The information requirements encompassed multifaceted tasks, including research, planning, and decision-making processes. Key research questions revolved around:

(1) **Personal Contexts**: How efficiently can an agent navigate various personal contexts, leading to distinct, relevant conversations?
(2) **Personalization**: Can agents adeptly modify conversational feedback based on the user's knowledge?
(3) **Elicitation**: Are agents proficient in drawing out pertinent persona information to customize discussions?
(4) **Dependent Relevance**: Can agents effectively employ context and prior responses to foster relevant conversations?

In Year 2, we continued this line by developing topics that would go beyond the complexities of Year 1 by:

- Including more complex and ambiguous PTKB statements;
- Testing models' ability not to answer the user's questions when the right answer does not exist.

The primary challenge in the track was to deliver a relevant and informative response given the user's PRKB. While these responses could be extractive passages, they might also amalgamate or summarize insights from various passages. Every response, though, should cite at least one "provenance" passage from the collection, maintaining a focus on passage/provenance ranking. As in preceding editions of TREC CAsT, systems can leverage all previous conversation turns as context, equivalent to taking the parents in the conversational topic tree.

In Year 2, we also focused on novel approaches to evaluation, focusing on *nugget-based evaluation* for generated responses and increasing the pool depth in passage assessment by introducing *dynamic pooling*. While equipped with LLM-based evaluation, we investigate novel approaches to building reusable collections for generated content.

2 TRACK, TASKS, DATA, AND RESOURCES

A detailed explanation of the task, data, and resources is provided below.

2.1 Track and Tasks

The focus of the track is on developing a personalized conversational search agent. In our track, the system is provided with some personal information about the user, and this information is considered when retrieving the relevant documents for the user's utterance and generating the response. Personal information about the user is provided in the PTKB, which is a set of narrative sentences. The sentences are assumed to be collected from previous conversations of the user with the system. Similar to CAsT [7], the main tasks are passage retrieval and answer generation but considering the user's persona in understanding the user's utterance. The track, also, includes the *statement classification* task where the relevant statements from PTKB to the current user utterance should be identified. To sum up, the track includes the following tasks:

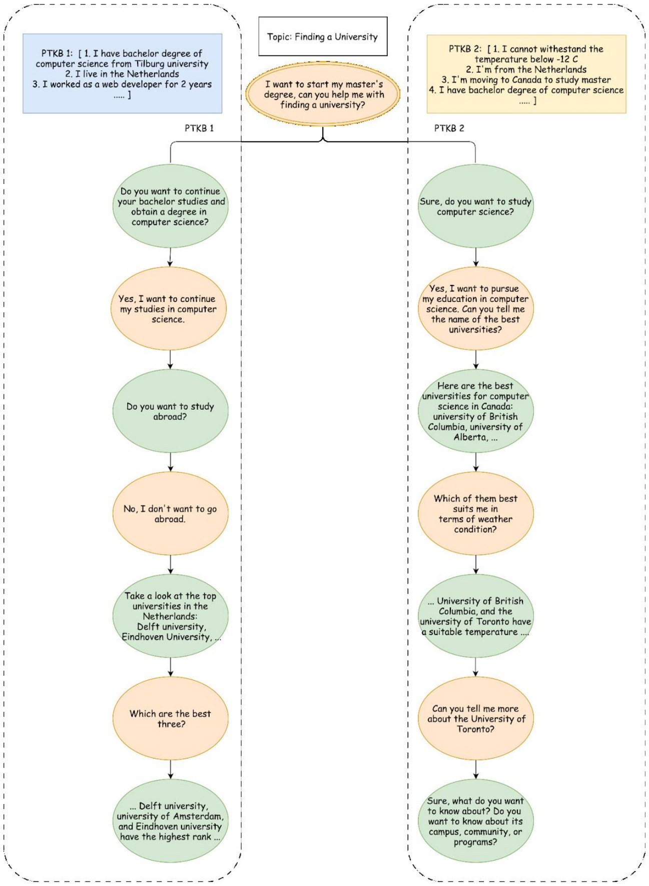
PTKB 1: [1. I have bachelor degree of computer science from Tilburg university
2. I live in the Netherlands
3. I worked as a web developer for 2 years
.....]
Topic: Finding a University
I want to start my master's degree, can you help me with finding a university?
PTKB 2: [1. I cannot withestand the temperature below -12 C
2. I'm from the Netherlands
3. I'm moving to Canada to study master
4. I have bachelor degree of computer science
.....]
PTKB 1
PTKB 2
Do you want to continue your bachelor studies and obtain a degree in computer science?
Yes, I want to continue my studies in computer science.
Do you want to study abroad?
No, I don't want to go abroad.
Take a look at the top universities in the Netherlands: Delft university, Eindhoven University, ...
Which are the best three?
... Delft university, university of Amsterdam, and Eindhoven university have the highest rank ...
Sure, do you want to study computer science?
Yes, I want to pursue my education in computer science. Can you tell me the name of the best universities?
Here are the best universities for computer science in Canada: university of British Columbia, university of Alberta, ...
Which of them best suits me in terms of weather condition?
... University of British Columbia, and the university of Toronto have a suitable temperature ....
Can you tell me more about the University of Toronto?
Sure, what do you want to know about? Do you want to know about its campus, community, or programs?

Figure 2: Two flowcharts representing different dialogues between a prospective student and an AI assistant on the topic of finding a suitable university for a master's degree in computer science. On the left, the conversation (PTKB 1) revolves around a student with a bachelor's degree from Tilburg University and work experience, who prefers to stay in the Netherlands. The dialogue suggests top Dutch universities and narrows down to the top three based on ranking. On the right, the second conversation (PTKB 2) involves a student who cannot tolerate cold temperatures below -12°C and is planning to move to Canada for a master's degree. The assistant provides options for top Canadian universities and further refines the suggestions to those with favorable weather conditions, eventually offering detailed information about the University of Toronto upon request. Each conversation flow is guided by the student's preferences, leading to tailored university recommendations.

- **Statement Classification:** The relevant statements from the PTKB for answering the current user utterance should be determined in this step. Unlike Year 1, we approach this task as a classification task. Given the context of the conversation and user utterance, the system classifies the statements from PTKB based on their relevance.
- **Passage Ranking:** Given the current user utterance, the context of the conversation, and the PTKB, the system retrieves relevant passages from the collection and ranks them based on their relevance.

- **Response Generation:** A response is the answer text that is intended to be shown to the user. It should be fluent, satisfy their information need, and not contain extraneous or redundant information. The response could be a generative or abstractive summary of the relevant passages.

2.2 Topics

The iKAT 2024 has 16 test topics. Unlike Year 1, we do not create multiple conversations per topic. A personalized turn is defined as a turn that has at least one relevant statement from PTKB. That means the system needs to consider at least one statement in the PTKB in order to answer the user's utterance accurately.

Topic creation. A complete set of guidelines was designed for topic creation. The guidelines included a detailed and step-by-step procedure for topic creation and a thorough explanation of the points that should be considered during the process. In addition, the guidelines included a checklist to ensure the quality of the topics. These criteria include quality assurance terms at the persona level, turn level, and conversation level. The topics are generated by organizers. We used a mix of LLM-assisted and personal exploration for each topic and discussed them in our meetings to improve the narrative. The topics that did not meet the quality criteria were regenerated by another annotator. Each topic developed was checked and refined by at least two other experts. The topic-creation process included the following steps:

(1) generate the user's PTKB for a given conversation/topic;
(2) form the user utterance's for each turn;
(3) identify the relevant PTKB statements;
(4) employ GPT-based relevance assessment to estimate the number of relevant passages for each turn;
(5) retrieve the relevant passages using the searcher tool provided to annotators (iKAT searcher), and;
(6) form the response of the system.

In generating the PTKB, we took great care to ensure that only high-level personal information was included (and any personally identifiable information) was not included to ensure the privacy of the contributors.

2.3 Collection

Considering the size of the ClueWeb22-B dataset, we utilized a subset of the ClueWeb22-B collection that we used in Year 1 too. To create this subset, we manually inspected the domains of the documents within the ClueWeb22-B dataset. We prioritized the diversity of domains and eliminated those that were not relevant. The final subset contained 116,838,987 passages, which was distributed by CMU.

To segment the documents into passages, we used a similar methodology to the one used by the TREC Deep Learning track for MS MARCO. We performed the following steps:

(1) each document was initially shortened to a length of 10,000 characters;
(2) a sliding window approach was then used, where we took 10 consecutive sentences as a single passage;
(3) after these 10 sentences, we moved the window by 5 sentences (i.e., a 5-sentence stride) to create the next passage.

We provided the following resources to the participants:

(1) Python scripts that were used to segment the passages;
(2) segmented passages along with MD5 hashes;
(3) Pyserini index of the collection;
(4) ir_datasets access to the collection and indexes, and;
(5) learned sparse index of the collection.

2.4 Baselines

The organizers provided six automatic baseline runs, 2 manual and 2 generation only runs, all detailed below:

(1) **baseline-auto-t5-bm25-minilm**. In this baseline, a T5 rewriter is used to rewrite the conversation, BM25 is used on the rewritten query, followed by a cross-encoder `MiniLM`. Generation is done with GPT-4o on the top 5 reranked passages, and PTKB independently with GPT-4o as well.
(2) **baseline-auto-convgqr-bm25-minilm**. This baseline is similar to **(1)**, with ConvGQR as query rewriter. The rest of the pipeline is identical.
(3) **baseline-auto-llama3.1-splade-minilm**. This baseline is similar to **(1)**, with zero-shot Llama3.1 as query rewriter.
(4) **baseline-auto-gpt4o-splade-minilm**. This run uses GPT-4o as query rewriter, with the `SPLADE++` retrieval model and the `MiniLM` cross-encoder. Generation is done with GPT-4o on the top 5 reranked passages, and PTKB independently with GPT4o as well.
(5) **baseline-auto-gpt4-bm25-minilm**. This baseline is similar to **(1)**, with zero-shot GPT-4 as query rewriter.
(6) **baseline-auto-gpt4o-bm25-minilm-genonly**. This run is the generation-only rerank runs. It is similar to **(1)**, with zero-shot GPT-4o as query rewriter.
(7) **baseline-gen-only-llama3.1-top5**. This run uses the generation-only retrieval runs, and applies Llama3.1 in zero-shot on top for response generation.
(8) **baseline-manual-bm25-minilm**. This run uses BM25 on the single human-rewritten query, with `MiniLM` reranker. Generation uses GPT-4o on the top5 reranked passages.
(9) **baseline-manual-splade-minilm**. This run uses SPLADE++ on the single human-rewritten query, with `MiniLM` reranker. Generation uses GPT-4o on the top5 reranked passages.

2.5 PTKB Statement Relevance Assessment

To assess the relevance of PTKB statements for each turn, we used two different sets of assessments which were created by the organizers and NIST assessors.

During topic generation, the organizers annotated each turn in terms of their provenance to PTKB statements and included their labels in the released topic files. To ensure the quality of these annotations, we assigned each turn to at least two of the organizers. In cases of disagreement, we assigned the turns to a third annotator and assigned the majority vote label.

Moreover, during the assessment of passage relevance, the NIST assessors were also asked to judge the relevance of PTKB statements to each turn. The assessment pool in this case was smaller than the one done by the organizers. The organizers judged all of the turns, while the NIST assessors only judged the turns that were selected for passage relevance. We only keep the turns that are

Table 1: Statistics of test data

Topics	17
Turns	218
PTKB statements	288
Assessed topics	14
Assessed turns	116
Avg. dialogue length	12.82
Avg. PTKB length	16.94
Passages assessed	20,575
Fails to meet (0)	10,680
Slightly meets (1)	4,246
Moderately meets (2)	4,325
Highly meets (3)	1,199
Fully meets (4)	125
PTKB turns assessed by NIST	114
PTKB assessments by NIST	1,917
Relevant (1)	201
PTKB turns assessed by the organizers	218
PTKB assessments by the organizers	3,660
Relevant (1)	175

annotated as *personalized* by both NIST and Organizers and report PTKB performance based on the assessments of both NIST and Organizers.

2.6 Passage Retrieval Assessment

The NIST assessors have judged the relevance of the passages based on the methodology used in CAsT (with the same scale). We selected a subset of 116 turns out of 218 to be judged by NIST assessors. Among the un-assessed turns, were responses that were clarifications (e.g., "Do you have any dietary requirements?") or were responses to utterances that were too general and returned too many relevant documents (e.g., "I'm traveling to California, do you have any suggestions?"). A pool of 20,575 passages was created and manually judged. An average number of 177 passages were judged for each turn. More detailed statistics of the collected data and judgments can be found in Table 1. We also reported the number of turns per dialogue, as well as the number of turns evaluated per dialogue in Figure 3.

2.7 Generated Response Quality Assessment

An automated approach was taken to assess the quality of generated responses, where we employed surface-based, semantic-based, and LLM-based metrics. Our comparison thus contains Rouge-1, Rouge-2, Rouge-L, BEM [4], as well as GPT-4o [5] and SOLAR-10.7B-Instructv1.0 [6], as a zero-shot answer equivalence evaluators. SOLAR has been shown to provide a good compromise between parameter size (efficiency) and effectiveness [9]. To do this, we selected a subset of the turns for assessment, discarding generic turns, while preserving personalized turns. We assessed the top-k responses generated for each turn for each submission. We also screened the responses and filtered out the low-quality responses. For example, if the response was not semantically similar to the top-ranked passages or included repeated sentences.

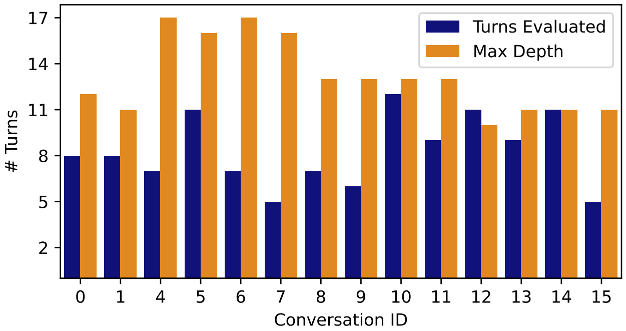

Figure 3: Number turns evaluated per dialogue in the final judgment pool vs. the maximum depth of each topic.

Given the subset of turns, we then selected the passages participants indicated that they used to generate the response. If they did not include the list of used passages, we considered the top 5 passages, as instructed in the guidelines.

We also included the groundedness metric from iKAT Y1 to measure to what extent the LLMs are faithful to the retrieval models, as defined below: **Groundedness:** Does the response appropriately reference or connect to the information provided in the provenance passages?

- *0: No* - The response does not reference the information provided in the provenance passages or is entirely disconnected from it.
- *1: Yes* - The response is directly based on the information provided in the provenance passages, accurately reflects this information, and utilizes it to enhance the response's relevance and completeness.

To ensure the quality of the assessments, we tested multiple setups and prompts and compared them to a subset of responses that the organizers manually labeled. We used the setting that had the highest agreement with the labeled data.

3 EVALUATION

Statement Ranking Task. We evaluated the PTKB statement ranking task at the turn and conversation levels. We used set-based metrics as this year we treated the PTKB ranking task as a classification task. Therefore, we report the following metrics: Precision, recall, and F1-Measure.

Passage Ranking Task. For the main task, we evaluated the runs across two dimensions given the ranking for each topic turn: (i) the ranking depth and (ii) the turn depth. For ranking depth, we focused on earlier positions 3 and 5 for the conversational scenario (where we assumed that the top k results would be used to formulate the response). Then for turn depth, we evaluated the run performance at the n-th conversational turn. Performing well on deeper rounds indicates a better ability to understand the preceding context. We used the mean nDCG@5 as the main evaluation metric, with all conversational turns averaged using uniform weights. We also measured the turn-depth measure based on nDCG@5, with the per turn nDCG@5 scores averaged at depth (*n*). In addition to the nDCG metrics (nDCG, nDCG@3, and nDCG@5), we also calculated P@10, Recall, Recall@10, and mean Average Precision, where again, we averaged over all turns.

Response Generation Task. Our evaluation for response generation this year relies on written gold responses from the NIST assessors, as well as extracted nuggets, or pieces of information from the relevant passages. First based on the gold response, we evaluated a diverse set of surface-based metrics such as Rouge. We then employed semantic-based metrics with BEM, and finally LLMEval with both GPT-4o and SOLAR-10.7B-Instructv1.0, as answer equivalence evaluators.

Using the human-written nuggets, we also assessed the coverage of the generated response with respect to these nuggets. For each response, we evaluated using GPT-4o how many of the nuggets were covered by the generated response. We used GPT-4o with Chain-of-Thoughts reasoning for this assessment: given the set of nuggets and the generated response, return the list of information pieces that are covered by the generated response. This assessment would be used to further compute recall metrics on the coverage of the nuggets.

Finally, given the high likelihood of LLMs being used in this year's submissions and the possibility of hallucination, we evaluated the generated responses in terms of groundedness. Groundedness measures whether the generated response can be attributed to the passages that it is supposed to be generated from. We use GPT-4o to evaluate the groundedness of the responses, as it demonstrated a high correlation with human labels in our preliminary experiments. For each turn, we used the GPT-4o assessments and took the mean groundedness on all turns.

4 PARTICIPANTS

The iKAT main task received 24 run submissions from seven groups shown in Table 2. The organizers provided four runs (two automatic, two manual) as baselines for comparison. Participants provided metadata and descriptions of their runs.

Most teams used a multi-step pipeline consisting of the following: (1) PTKB statement relevance prediction; (2) conversational rewriting (most incorporating the previous canonical responses as well as predicted relevance PTKB statements) and conversational query expansion; (3) retrieval using traditional or dense IR model; and (4) multi-stage passage re-ranking with neural language models fine-tuned for point-wise (mono) and pairwise (duo) ranking. Table 2 lists the submissions from the teams, as well as their pipelines.

4.1 Participant Runs

Table 2 provides an overview of the participant runs, and below we include a summary of each, starting with **Automatic** runs:

(1) **iiresearch_ikat2024_rag_top5_bge_reranker**. Uses a fine-tuned LLaMA3 for query rewriting. Statement ranking employs SBERT, GPT-4o, and a fine-tuned Gemma model. For passage ranking, BM25 retrieves 1000 documents, and BGE reranks them. A retrieval-augmented approach determines if additional retrieval optimizes the generation output.

Table 2: Participants and their runs.

Group	Run ID	Run Category
Organizers	baseline-gen-only-llama3.1-top5	gen_only
Organizers	baseline-auto-gpt4o-splade-minilm	auto
Organizers	baseline-auto-llama3.1-splade-minilm	auto
Organizers	baseline-auto-convgqr-bm25-minilm	auto
Organizers	baseline-auto-t5-bm25-minilm	auto
Organizers	baseline-auto-gpt4-bm25-minilm	auto
Organizers	baseline-manual-splade-minilm	manual
Organizers	baseline-auto-gpt4o-bm25-minilm-genonly	auto
Organizers	baseline-gen-only-gpt4o-top5	gen_only
Organizers	baseline-manual-bm25-minilm	manual
RALI	RALI_gpt4o_fusion_norerank	auto
RALI	RALI_manual_monot5	manual
RALI	RALI_gpt4o_fusion_rerank	auto
RALI	RALI_gpt4o_no_personalize_fusion_rerank	auto
RALI	RALI_gpt4o_no_personalize_fusion_norerank	auto
RALI	RALI_manual_rankllama	manual
UvA	gpt4-MQ-debertav3	auto
UvA	gpt4-mq-rr-fusion	auto
UvA	gpt-single-QR-rr-debertav3	auto
UvA	qd1	auto
UvA	manual-splade-debertav3	manual
UvA	manual-splade-fusion	manual
dcu	dcu_auto_qe_summ_TopP_3	auto
dcu	dcu_manual_qe_summ_ptkb_TopP_3	manual
dcu	dcu_manual_qe_summ_TopP_3	manual
dcu	dcu_auto_qe_key_topP-50_topK-5	auto
dcu	dcu_auto_qre_sim	auto
dcu	dcu_auto_qe_summ_ptkb_TopP_	auto
iiresearch	iiresearch_ikat2024_rag_top5_monot5_reranker	auto
iiresearch	iiresearch_ikat2024_rag_top5_bge_reranker	auto
infosense	infosense_llama_pssgqrs_wghtdrerank_1	auto
infosense	infosense_llama_short_long_qrs_3	auto
infosense	infosense_llama_short_long_qrs_2	auto
infosense	infosense_llama_pssgqrs_wghtdrerank_2	auto
ksu	ksu_created_query_reranking	auto
nii	NII_automatic_GeRe	auto
nii	nii_auto_base	auto
nii	nii_manu_ptkb_rr	manual
nii	nii_res_gen	gen_only
nii	nii_manu_base	manual
nii	nii_auto_ptkb_rr	auto

(2) **iiresearch_ikat2024_rag_top5_monot5_reranker**. Similar to **(1)** but replaces BGE with monoT5 for passage reranking.

(3) **RALI_gpt4o_fusion_rerank**. Four steps: (1) GPT-4o generates three rewritten queries: de-contextualized (non-personalized), pseudo-response concatenated, and de-contextualized personalized. (2) BM25 lists for these queries are fused. (3) The top 50 documents are reranked using monoT5 based on the personalized query. (4) GPT-4o generates a response considering the conversation context, PTKB, and top 3 reranked documents.

(4) **RALI_gpt4o_no_personalize_fusion_rerank**. Similar to **(3)**, with an additional fourth rewritten de-contextualized and non-personalized query.

(5) **RALI_gpt4o_no_personalize_fusion_norerank**. Retrieval-only. Similar to **(4)**, but skips reranking and directly uses the fused BM25 list for retrieval.

Table 3: Automatic evaluation of passage retrieval results. Evaluation at retrieval cutoff of 1000.

Group	Run ID	nDCG@3	nDCG@5	nDCG	P@20	Recall@20	Recall	mAP
UvA	gpt4-MQ-debertav3	0.5320	0.5156	0.6071	0.5849	0.1732	0.7717	0.3421
UvA	gpt4-mq-rr-fusion	0.5103	0.5119	0.6164	0.6060	0.1801	0.7789	0.3624
RALI	RALI_gpt4o_fusion_rerank	0.5288	0.5111	0.4677	0.5073	0.1525	0.5883	0.2106
UvA	gpt-single-QR-rr-debertav3	0.5178	0.5028	0.5416	0.5625	0.1685	0.6536	0.3122
RALI	RALI_gpt4o_no_personalize_fusion_rerank	0.5087	0.4979	0.4557	0.4978	0.1458	0.5806	0.2025
UvA	qd1	0.4940	0.4788	0.4330	0.5190	0.1556	0.4995	0.2295
infosense	infosense_llama_short_long_qrs_3	0.4879	0.4722	0.5338	0.5392	0.1507	0.6869	0.2591
infosense	infosense_llama_short_long_qrs_2	0.4741	0.4607	0.5080	0.4957	0.1438	0.6523	0.2433
Organizers	baseline-auto-gpt4o-bm25-minilm-genonly	0.4412	0.4150	0.3829	0.4151	0.1382	0.4503	0.1944
Organizers	baseline-auto-gpt4-bm25-minilm	0.4252	0.4086	0.3771	0.4444	0.1334	0.4391	0.1915
nii	NII_automatic_GeRe	0.4233	0.4075	0.4637	0.4362	0.1342	0.6037	0.2201
Organizers	baseline-auto-gpt4o-splade-minilm	0.4279	0.4068	0.4728	0.4302	0.1417	0.6258	0.2354
RALI	RALI_gpt4o_fusion_norerank	0.3805	0.3786	0.4337	0.4108	0.1236	0.5883	0.1809
nii	nii_auto_ptkb_rr	0.3885	0.3766	0.4096	0.3966	0.1191	0.5131	0.1991
nii	nii_auto_base	0.3867	0.3764	0.4090	0.3953	0.1187	0.5129	0.1987
RALI	RALI_gpt4o_no_personalize_fusion_norerank	0.3728	0.3645	0.4225	0.3974	0.1155	0.5806	0.1737
infosense	infosense_llama_pssgqrs_wghtdrerank_2	0.3729	0.3637	0.4197	0.4099	0.1155	0.5578	0.1921
infosense	infosense_llama_pssgqrs_wghtdrerank_1	0.3481	0.3423	0.3799	0.3655	0.0996	0.5207	0.1575
Organizers	baseline-auto-convgqr-bm25-minilm	0.2413	0.2332	0.2293	0.2539	0.0809	0.2913	0.1043
Organizers	baseline-auto-t5-bm25-minilm	0.2347	0.2331	0.2374	0.2707	0.0814	0.2955	0.1110
dcu	dcu_auto_qre_sim	0.1610	0.1632	0.1559	0.1780	0.0491	0.2074	0.0662
ksu	ksu_created_query_reranking	0.1743	0.1595	0.0478	0.0741	0.0212	0.0212	0.0159
dcu	dcu_auto_qe_key_topP-50_topK-5	0.0894	0.0878	0.0830	0.0953	0.0267	0.1170	0.0305
iiresearch	iiresearch_ikat2024_rag_top5_monot5_reranker	0.0435	0.0528	0.0114	0.0272	0.0060	0.0060	0.0032
iiresearch	iiresearch_ikat2024_rag_top5_bge_reranker	0.0493	0.0492	0.0125	0.0241	0.0060	0.0060	0.0038
dcu	dcu_auto_qe_summ_TopP_3	0.0446	0.0443	0.0376	0.0414	0.0111	0.0525	0.0107
dcu	dcu_auto_qe_summ_ptkb_TopP_	0.0311	0.0294	0.0227	0.0345	0.0083	0.0287	0.0052

Table 4: Automatic evaluation of passage retrieval results on manual runs. Evaluation at retrieval cutoff of 1000.

Group	Run ID	nDCG@3	nDCG@5	nDCG	P@20	Recall@20	Recall	mAP
UvA	manual-splade-fusion	0.5446	0.5418	0.5838	0.5948	0.1950	0.6983	0.3524
nii	nii_manu_base	0.4895	0.4776	0.4886	0.5246	0.1617	0.5837	0.2554
nii	nii_manu_ptkb_rr	0.4879	0.4756	0.4892	0.5246	0.1611	0.5837	0.2561
UvA	manual-splade-debertav3	0.4767	0.4754	0.5524	0.5470	0.1797	0.6983	0.3086
Organizers	baseline-manual-splade-minilm	0.4374	0.4284	0.5185	0.4707	0.1496	0.6983	0.2552
Organizers	baseline-manual-bm25-minilm	0.4374	0.4249	0.3932	0.4371	0.1418	0.4653	0.1973
RALI	RALI_manual_rankllama	0.4414	0.4230	0.3522	0.3664	0.1145	0.4653	0.1362
RALI	RALI_manual_monot5	0.4127	0.4042	0.3483	0.3651	0.1127	0.4653	0.1355
dcu	dcu_manual_qe_summ_ptkb_TopP_3	0.2512	0.2397	0.2066	0.2401	0.0813	0.2433	0.0867
dcu	dcu_manual_qe_summ_TopP_3	0.2244	0.2174	0.1966	0.2237	0.0732	0.2385	0.0783

(6) **RALI_gpt4o_fusion_norerank**. Retrieval-only. Similar to **(3)**, but skips reranking and uses the fused BM25 list for retrieval.

(7) **infosense_llama_pssgqrs_wghtdrerank_2**. The run uses LLaMA to summarize the previous turn's response and appends it to the conversation history. Based on this updated context, LLaMA generates a clarified version of the user's utterance and derives a dictionary of relevant PTKB queries from it. Subsequently, a 10-sentence passage is created from the clarified utterance combined with all relevant PTKBs, along with additional 10-sentence passages for up to three individual PTKBs. BM25 is employed to retrieve up to 5000 documents for each query, ensuring unique retrieval by removing duplicates across queries. The retrieved documents are then iteratively reranked using msmarco-distilbert-base-v4 and all-MiniLM-L12-v2, with scores weighted by the PTKB's

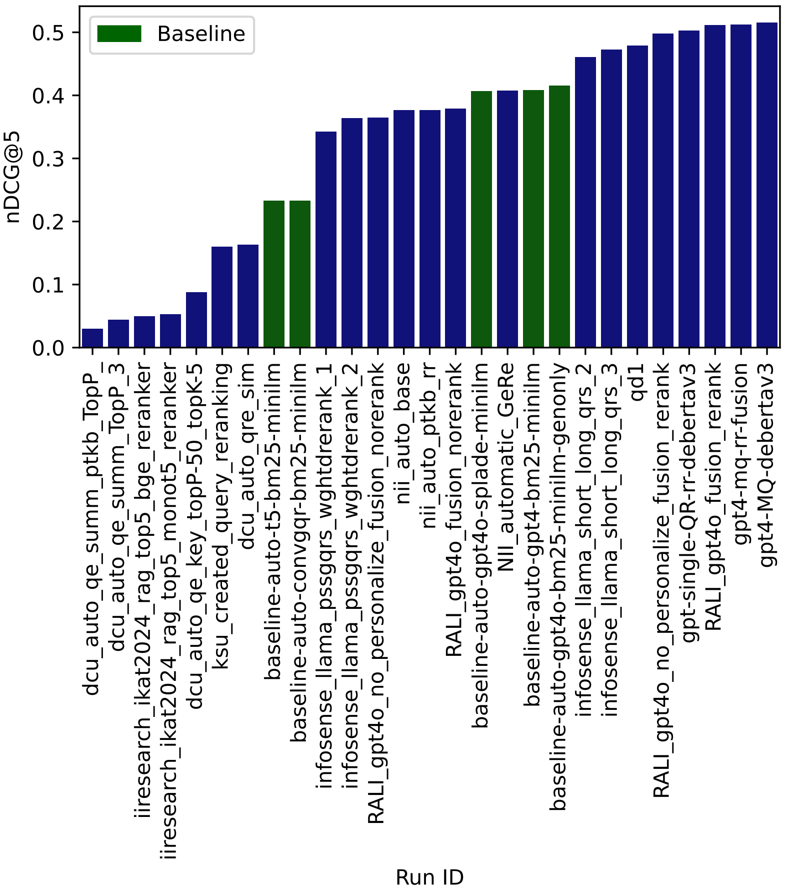

Figure 4: Performance of all automatic runs in terms of nDCG@5 on the passage ranking task.

relevance, such as assigning a weight of 1 to the query involving all relevant PTKBs. Finally, LLaMA generates the response based on the clarified user utterance and the top three reranked documents, ensuring an accurate and contextually informed answer.

(8) **infosense_llama_pssgqrs_wghtdrerank_1**. Similar to **(7)**, but uses only all-MiniLM-L12-v2 for reranking.

(9) **infosense_llama_short_long_qrs_2**. Similar to **(7)**, but generates PTKB search suggestions based on clarified user context using Llama.

(10) **infosense_llama_short_long_qrs_3**. Similar to **(7)**, but employs LLaMA-70B instead of 8B.

(11) **nii_auto_base**. Follows a GPT-4o-based pipeline: Rewrite the utterance, extract key information, rank PTKB, generate queries, and retrieve and rerank documents using BM25 and a cross-encoder.

(12) **nii_auto_ptkb_rr**. Extends **(11)** by reranking documents based on related PTKB with a cross-encoder.

(13) **NII_automatic_GeRe**. Steps: (1) Generate an initial answer using GPT-4. (2) Generate five queries based on this answer. (3) Retrieve 300 documents per query with BM25, rerank with a cross-encoder. (4) Combine, deduplicate, and rerank the top 1000 documents. (5) Use PTKB and context to refine the final answer. Steps repeated for GPT-4 and Claude3, and results are merged and reranked.

(14) **dcu_auto_qe_key_topP-50_topK-5**. Uses BM25 to retrieve 1000 passages, reranked with a cross-encoder. Extracts top 5 keywords and top 50 passages to enrich queries for subsequent turns. PTKB ranking selects the top 3 based on cosine similarity to enriched queries.

(15) **dcu_auto_qre_sim**. Historical conversational queries with cosine similarity <0.50 to user utterance are included as context. Queries are rewritten with a T5-based model, then BM25 retrieves 1000 passages followed by cross-encoder reranking. PTKB ranking selects the top 3 by cosine similarity.

(16) **dcu_auto_qe_summ_TopP_3**. Retrieves 1000 passages with BM25, reranks with a cross-encoder. Generates an abstractive summary of the top 3 passages for query enrichment. PTKB ranking selects the top 3 by cosine similarity to enriched queries.

(17) **dcu_auto_qe_summ_ptkb_TopP_**. Similar to **(16)**, but includes the top 3 PTKB in the query enrichment process.

(18) **ksu_created_query_reranking**. Steps: (1) Rewrite the utterance with LLaMA3.1 (8B). (2) Rank PTKB entries using

Sentence-T5. (3) Rewrite the query with top 3 PTKB results. (4) Decompose queries for BM25. (5) Retrieve top 10 results. (6) Summarize passages using Pegasus-XSum. (7) Generate responses using LLaMA.

(19) **gpt4-MQ-debertav3**. Generates five queries with GPT-4, retrieves 1k passages with SPLADE for each, then reranks the 5k passages using DeBERTaV3. Generates answers with GPT-4 and classifies PTKB entries.
(20) **gpt4-mq-rr-fusion**. Similar to **(19)**, but uses a fusion of five rerankers: Electra, DeBERTaV2, DeBERTaV3, RoBERTa, and ALBERT, each reranking 5000 passages. Ensembling is done with min-max normalization.
(21) **gpt-single-QR-rr-debertav3**. This run uses GPT4 as query rewrite, followed by SPLADE++ retrieval and a DebertaV3 cross-encoder. Answer generation uses GPT4, and PTKB classification aswell.
(22) **qd1**. Single-query rewrite, BM25 retrieval, and MiniLM reranking. Answer generation uses GPT4, and PTKB classification aswell.

As well as several **Manual** runs:

(1) **RALI_manual_monot5**: This run consists of two main steps: (1) Retrieval: BM25 is used to retrieve the top 1000 documents based on the manual rewrite. (2) Reranking: The top 50 documents are reranked using the monoT5 model based on the manual rewrite.
(2) **RALI_manual_rankllama**: This run also includes two steps: (1) Retrieval: BM25 retrieves the top 1000 documents based on the manual rewrite. (2) Reranking: The top 50 documents are reranked using the rankllama model based on the manual rewrite.
(3) **manual-splade-debertav3**: This run uses SPLADE for reranking 1000 passages with a single cross-encoder, debertav2. Response generation is done using GPT-4.
(4) **manual-splade-fusion**: This run also uses SPLADE for reranking, but with an ensemble of five cross-encoders: electra, debertav2, debertav3, roberta, and albert. Each cross-encoder reranks 1000 passages. Response generation is performed using GPT-4.
(5) **nii_manu_base**: This process follows a pipeline that includes: (1) Rewriting the manual utterance to extract only the necessary information using GPT-4. (2) Ranking the PTKB based on the rewritten utterance and context with GPT-4. (3) Generating queries from the rewritten utterance and relevant PTKB entries using GPT-4. (4) Retrieving and reranking documents using BM25 and a cross-encoder based on the rewritten utterance.
(6) **nii_manu_ptkb_rr**: This run follows the same pipeline as (5), with the addition of re-ranking documents based on the related PTKB using a cross-encoder.
(7) **dcu_manual_qe_summ_TopP_3**: In this manual run, resolved utterances are used as queries. BM25 retrieves the top 1000 passages, followed by reranking with a cross-encoder. An abstractive summary is generated from the top 3 passages and used to enrich the query for the next turn.
(8) **dcu_manual_qe_summ_ptkb_TopP_3**: Similar to the previous run, this manual run uses resolved utterances along with their ground-truth PTKB provenance statement for querying. BM25 retrieves the top 1000 passages, followed by reranking with a cross-encoder. An abstractive summary is generated from the top 3 passages and used to enrich the query for the next turn.

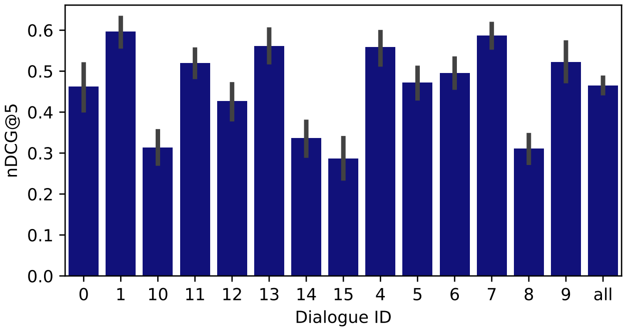

Figure 5: nDCG@5 aggregated for each topic across all runs on the passage ranking task. We report the average across runs, median or better.

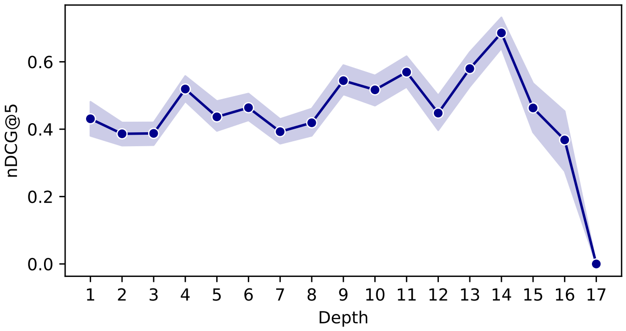

Figure 6: nDCG@5 at varying conversation turn depths on the passage ranking task. We report the average across runs, median or better.

We also received one **Generation-Only** submission:

(1) **nii_res_gen**. This run uses the provided passage ranking and generates the responses based on Gemini-1.5-flash.

5 RESULTS

5.1 Passage Ranking

5.1.1 Overall results. Table 3 lists the performance of the automatic runs in terms of all the evaluation metrics. We see the dominance of models that leverage GPT-* in their pipeline on top of the list, followed by models that leverage Llama. Figure 4 compares the performance of all the automatic runs in terms of nDCG@5, where the baseline runs are colored in green. We received 8 manual runs this year, listed in Table 4. Unlike Year 1 and CAsT, we do not see a big gap between manual and automatic runs, indicating the improved ability of LLMs to resolve user utterances automatically.

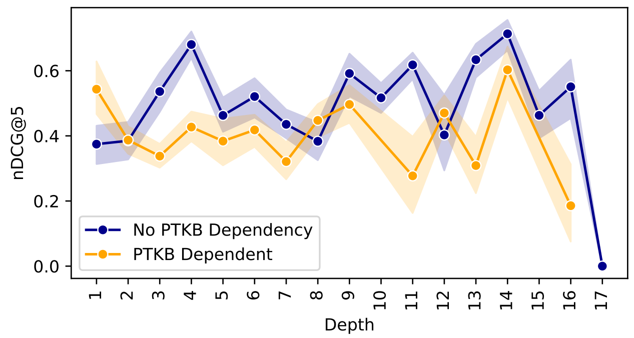

Figure 7: nDCG@5 at varying conversation turn depths on the passage ranking task, for turns that depend on PTKB statements vs. those that do not. We report the average across runs, median or better.

5.1.2 Performance per dialogue. Figure 5 reports the average performance in terms of nDCG@5 of all runs that median or better. We see that while the runs perform well for some of the topics, they fail to perform well for some. In particular, we find topics 1 and 7 to be the easiest, while 15 and 10 to be the most difficult ones.

5.1.3 Performance at different depths. Figure 6 reports the performance of all runs (median or better) at varying conversation turns in terms of nDCG@5. We also report the performance at different depths, separating the turns that depend on PTKB provenance in Figure 7. Our intuition is that the PTKB statement ranking step will introduce additional difficulty and error in the pipeline and consequently the runs exhibit lower performance. However, we see that this was not always the case, and in most cases, PTKB dependence led to lower performance. Unlike CAsT and iKAT Year 1, we see that the models do not necessarily perform best in the first turns. Interestingly, we see an upward performance trend in deeper turns, with the peak performance at depth 14. This is an interesting phenomenon that needs further investigation. When looking at the personalized turns in Figure 7, we observe a different trend, showing that PTKB-dependent turns generally become more challenging as the dialogue progresses. This is also corroborated by the turn-level PTKB performance presented in Figure 9.

5.2 PTKB Provenance

5.2.1 Overall results. As previously described, we evaluated the submissions for the PTKB statement ranking task based on two relevance judgments, namely, assessed by the NIST assessors, as well as the organizers. We report the results based on NIST assessments in Table 5, and the results based on the organizers' assessment in Table 6 in terms of all evaluation metrics. We report the results only on the intersection of turns that are deemed to be personalized by both NIST assessors and organizers. Unlike Year 1, we do not see a high agreement between the two tables in the relative order of the submissions.

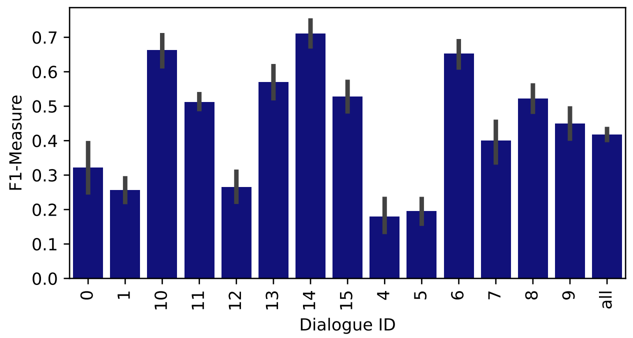

Figure 8: F1-Measure on PTKB relevance prediction, aggregated for each topic across all runs. We report the average across runs, median or better.

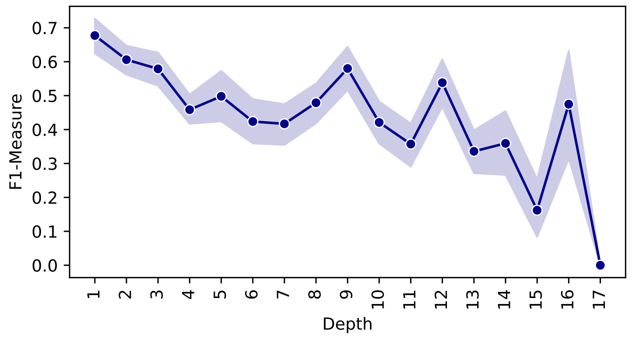

Figure 9: F1-Measure on PTKB relevance prediction at varying conversation turn depths. We report the average across runs, median or better.

5.2.2 Performance per dialogue. Using the organizers' assessments, in Figure 8 we plotted the mean performance of all the submissions (median and better) in terms of F1-Measure, aggregated on each topic. While we observed a reasonably high performance for all the topics, we find topic 4 to be the most challenging for this task, and 14 to be among the easiest ones.

5.2.3 Performance at different depths. Using the organizers' assessments, in Figure 9 we plot the mean performance of all the submissions (median and better) in terms of F1-Measure, at varying conversation depths. We noticed a high variance in the performance of different models when the higher conversation depths. Intuitively, we see the highest performance at depth 1 as the dialogues are simpler and the performance generally goes down at deeper turns.

5.3 Response Evaluation

We present in Table 7 the results of the response generation task. Overall, the runs with high effectiveness also perform well on the generation task. We can also see that manual runs (third set of rows), perform also good. Across metrics, we can see the trend that SOLAR seems to overestimate the quality of the response, while GPT-4o does not as much. Note that the LLMeval metric prompts LLMs to compare the generated response with the human written response, and not to assess the generated response based

Table 5: Performance of automatic runs on the PTKB provenance task based on NIST assessment.

Group	Run ID	Precision	Recall	F1-Measure
nii	nii_auto_base	0.5222	0.5499	0.4775
nii	nii_auto_ptkb_rr	0.5222	0.5499	0.4775
nii	NII_automatic_GeRe	0.4225	0.5806	0.4383
Organizers	baseline-auto-gpt4-bm25-minilm	0.5191	0.4169	0.4015
Organizers	baseline-auto-convgqr-bm25-minilm	0.5191	0.4169	0.4015
Organizers	baseline-auto-gpt4o-splade-minilm	0.5191	0.4169	0.4015
Organizers	baseline-auto-t5-bm25-minilm	0.5191	0.4169	0.4015
Organizers	baseline-auto-gpt4o-bm25-minilm-genonly	0.5191	0.4169	0.4015
UvA	qd1	0.4837	0.4056	0.3910
UvA	gpt4-MQ-debertav3	0.4837	0.4056	0.3910
UvA	gpt4-mq-rr-fusion	0.4837	0.4056	0.3910
UvA	gpt-single-QR-rr-debertav3	0.4837	0.4056	0.3910
infosense	infosense_llama_short_long_qrs_2	0.3847	0.4750	0.3550
iiresearch	iiresearch_ikat2024_rag_top5_bge_reranker	0.4407	0.3691	0.3424
iiresearch	iiresearch_ikat2024_rag_top5_monot5_reranker	0.4407	0.3691	0.3424
dcu	dcu_auto_qre_sim	0.3041	0.2683	0.2488
dcu	dcu_auto_qe_summ_ptkb_TopP_	0.3041	0.2418	0.2429
dcu	dcu_auto_qe_key_topP-50_topK-5	0.3099	0.2416	0.2427
dcu	dcu_auto_qe_summ_TopP_3	0.2836	0.2494	0.2309
ksu	ksu_created_query_reranking	0.2661	0.2385	0.2184

Table 6: Performance of automatic runs on the PTKB provenance task based on the organizers' assessment.

Group	Run ID	Precision	Recall	F1-Measure
UvA	gpt4-mq-rr-fusion	0.4486	0.6571	0.4935
UvA	gpt-single-QR-rr-debertav3	0.4486	0.6571	0.4935
UvA	qd1	0.4486	0.6571	0.4935
UvA	gpt4-MQ-debertav3	0.4486	0.6571	0.4935
Organizers	baseline-auto-gpt4o-bm25-minilm-genonly	0.4323	0.5785	0.4686
Organizers	baseline-auto-convgqr-bm25-minilm	0.4323	0.5785	0.4686
Organizers	baseline-auto-gpt4-bm25-minilm	0.4323	0.5785	0.4686
Organizers	baseline-auto-t5-bm25-minilm	0.4323	0.5785	0.4686
Organizers	baseline-auto-gpt4o-splade-minilm	0.4323	0.5785	0.4686
nii	nii_auto_base	0.3317	0.7099	0.4276
nii	nii_auto_ptkb_rr	0.3317	0.7099	0.4276
infosense	infosense_llama_short_long_qrs_2	0.2946	0.6208	0.3741
nii	NII_automatic_GeRe	0.2567	0.6641	0.3505
iiresearch	iiresearch_ikat2024_rag_top5_bge_reranker	0.2888	0.5147	0.3349
iiresearch	iiresearch_ikat2024_rag_top5_monot5_reranker	0.2888	0.5147	0.3349
infosense	infosense_llama_pssgqrs_wghtdrerank_2	0.2204	0.6353	0.3079
infosense	infosense_llama_pssgqrs_wghtdrerank_1	0.2204	0.6353	0.3079
dcu	dcu_auto_qre_sim	0.2179	0.3910	0.2674
dcu	dcu_auto_qe_key_topP-50_topK-5	0.2179	0.3587	0.2569
dcu	dcu_auto_qe_summ_ptkb_TopP_	0.2179	0.3019	0.2422
dcu	dcu_auto_qe_summ_TopP_3	0.1923	0.3244	0.2286
ksu	ksu_created_query_reranking	0.1859	0.3013	0.2200

on its internal knowledge. Overall, the recall on nuggets is also low compared to LLMeval. This might be due to the creation process of the nuggets, which include lots of subpieces of information from a large set of passages. Finally, when looking at the groundedness across methods, we can identify the one from NII that all have very high groundedness compared to other runs. Finally, looking at the generation-only runs, we see that Llama3.1 has better groundedness, but lower quality on the generated responses.

Table 7: Automatic, Generation-only and Manual evaluation of response generation. Best is sorted according to LLMeval GPT-4o.

Group	Run ID	BEM	Groundedness	LLMeval		R-Nuggets	Rouge-L
				SOLAR	GPT-4o		
UvA	gpt-single-QR-rr-debertav3	0.2652	0.4355	0.9355	0.7903	0.2337	0.1995
UvA	gpt4-mq-rr-fusion	0.2722	0.3387	0.9839	0.7581	0.2165	0.1967
RALI	RALI_gpt4o_no_personalize_fusion_rerank	0.2346	0.6129	0.8871	0.7097	0.1716	0.2139
UvA	gpt4-MQ-debertav3	0.2691	0.3548	0.9355	0.7097	0.2379	0.1987
Organizers	baseline-auto-gpt4o-splade-minilm	0.2879	0.5484	0.9677	0.7097	0.1962	0.1969
nii	NII_automatic_GeRe	0.2631	0.8710	0.9516	0.6774	0.1983	0.2019
RALI	RALI_gpt4o_fusion_rerank	0.2462	0.5645	0.9194	0.6452	0.1820	0.2221
Organizers	baseline-auto-gpt4-bm25-minilm	0.2530	0.4839	0.9194	0.6452	0.1656	0.1933
infosense	infosense_llama_short_long_qrs_3	0.2529	0.0968	0.8033	0.6452	0.1485	0.2373
UvA	qd1	0.2522	0.3871	0.9516	0.6290	0.1953	0.1908
Organizers	baseline-auto-convgqr-bm25-minilm	0.2673	0.5323	0.9355	0.5968	0.1559	0.1948
infosense	infosense_llama_short_long_qrs_2	0.2245	0.2903	0.7869	0.5806	0.0874	0.2277
Organizers	baseline-auto-t5-bm25-minilm	0.2667	0.7097	0.8226	0.5484	0.1578	0.1842
Organizers	baseline-auto-llama3.1-splade-minilm	0.2095	0.6774	0.6129	0.4194	0.0961	0.1981
infosense	infosense_llama_pssgqrs_wghtdrerank_2	0.2126	0.5645	0.6290	0.4032	0.0937	0.2183
infosense	infosense_llama_pssgqrs_wghtdrerank_1	0.2267	0.6452	0.6613	0.3065	0.0962	0.2173
iiresearch	iiresearch_ikat2024_rag_top5_bge_reranker	0.1903	0.6774	0.1774	0.1290	0.0147	0.1451
ksu	ksu_created_query_reranking	0.1484	0.7500	0.0645	0.0645	0.0036	0.1434
iiresearch	iiresearch_ikat2024_rag_top5_monot5_reranker	0.1223	0.8548	0.0161	0.0323	0.0006	0.0913
Organizers	baseline-auto-gpt4o-bm25-minilm-genonly	0.2830	0.4677	0.9836	0.6290	0.1860	0.2000
nii	nii_res_gen	0.2043	0.9193	0.5806	0.4355	0.0937	0.1746
Organizers	baseline-gen-only-llama3.1-top5	0.2577	0.6451	0.6557	0.4193	0.1411	0.2060
UvA	manual-splade-fusion	0.2830	0.4194	0.9839	0.7903	0.2175	0.1984
UvA	manual-splade-debertav3	0.2476	0.4355	1.0000	0.7258	0.2209	0.1953
Organizer	baseline-manual-splade-minilm	0.2668	0.4355	0.9355	0.7097	0.2453	0.1994
Organizer	baseline-manual-bm25-minilm	0.2683	0.5161	0.9672	0.6613	0.1465	0.1955

6 CONCLUSION

The second TREC iKAT edition built on the first year and developed resources for studying personalized conversational information seeking and added to the community's understanding of the topic. As a successor of TREC CAsT, it made significant advances over CAsT, by focusing on more personalized and complex conversations that require advanced reasoning and leveraging the personal knowledge graphs to provide relevant responses. The PTKB statement ranking task provided a way for participants to leverage users' personal information into the conversation. In year 2, we observed more LLM-based methods to be tested by the participants, leading to novel trends in performance. Most runs used BM25 as their first-stage retrieval method, with a few methods leveraging learned sparse indexes, but no groups leveraged dense indexes. This year we tried two novel approaches of evaluation, namely, dynamic LLM-assisted pooling, and nugget-based evaluation of generated responses with gold human-generated responses.

7 ACKNOWLEDGMENTS

We are grateful to Jamie Callan's great help and patience in enabling us to provide the participants with a sample of the ClueWeb'22 collection. We are thankful for Hoa Dang and Ian Soboroff's experience, patience, and persistence in running the assessment process. We are also thankful to Andrew Ramsay for his help throughout the project. Finally, we thank all our participants and annotators.

REFERENCES

[1] Aliannejadi, M., Abbasiantaeb, Z., Chatterjee, S., Dalton, J., Azzopardi, L.: TREC iKAT 2023: The interactive knowledge assistance track overview. CoRR **abs/2401.01330** (2024)

[2] Aliannejadi, M., Abbasiantaeb, Z., Chatterjee, S., Dalton, J., Azzopardi, L.: TREC iKAT 2023: A test collection for evaluating conversational and interactive knowledge assistants. In: SIGIR. pp. 819–829. ACM (2024)

[3] Azzopardi, L., Dubiel, M., Halvey, M., Dalton, J.: Conceptualizing agent-human interactions during the conversational search process. In: The second international workshop on conversational approaches to information retrieval (2018)

[4] Bulian, J., Buck, C., Gajewski, W., Boerschinger, B., Schuster, T.: Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation. arXiv preprint arXiv:2202.07654 (2022)

[5] Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al.: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024)

[6] Kim, D., Park, C., Kim, S., Lee, W., Song, W., Kim, Y., Kim, H., Kim, Y., Lee, H., Kim, J., et al.: Solar 10.7 b: Scaling large language models with simple yet effective depth up-scaling. arXiv preprint arXiv:2312.15166 (2023)

[7] Owoicho, P., Dalton, J., Aliannejadi, M., Azzopardi, L., Trippas, J.R., Vakulenko, S.: Trec cast 2022: Going beyond user ask and system retrieve with initiative and response generation. In: Proceedings of the NIST Text Retrieval Conference, TREC 2022. pp. 1–11 (2023)

[8] Radlinski, F., Craswell, N.: A theoretical framework for conversational search. In: Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval. pp. 117–126. ACM (2017)

[9] Rau, D., Déjean, H., Chirkova, N., Formal, T., Wang, S., Nikoulina, V., Clinchant, S.: Bergen: A benchmarking library for retrieval-augmented generation. arXiv

preprint arXiv:2407.01102 (2024)

[10] Russell-Rose, T., Chamberlain, J., Azzopardi, L.: Information retrieval in the workplace: A comparison of professional search practices. Information Processing & Management **54**(6), 1042–1057 (2018)