GRILL Lab at TREC 2025: Agentic Iterative Retrieval and Gap-Aware Refinement for TREC IKAT and TREC RAG

Paul Owoicho and Jeff Dalton

University of Edinburgh, United Kingdom

March 9, 2026

Abstract

This paper describes the GRILL Lab's participation in the TREC 2025 Interactive Knowledge Assistance Track (IKAT) and the Retrieval-Augmented Generation (RAG) track, covering four sub-tasks: IKAT Passage Ranking/Response Generation, IKAT Simulation, RAG Retrieval Only, and RAG Full. Our approach centres on a modular, agentic pipeline that pursues high recall through iterative feedback. The system proceeds in three stages: (1) initial candidate generation via BM25; (2) document expansion using Query-by-Document techniques; and (3) an LLM-driven gap analysis phase in which the model identifies informational gaps and formulates supplementary queries. A key architectural feature is a fine-tuned GPT-4.1 nano binary relevance filter, trained on TREC CAsT 2022 and IKAT 2023 relevance judgments, which prunes irrelevant documents between each stage to contain topic drift.

1 Introduction

Interactive information retrieval demands systems that can adapt to evolving user needs and multi-turn conversational context. Static, single-pass retrieval pipelines struggle with complex topics that require iterative exploration: a single query rarely captures the full scope of a user's information need, and naïve result-set expansion risks flooding the system with off-topic material.

For TREC 2025, the GRILL Lab tested the hypothesis that a saturation-based retrieval strategy in which the system continues seeking information until no new relevant documents are returned provides superior topic coverage compared to single-pass approaches. We applied this philosophy across both the IKAT and RAG tracks, deploying Large Language Models (LLMs) not only as response generators but as active agents within the retrieval loop: rewriting queries, expanding document sets, filtering noise, and identifying residual informational gaps.

The remainder of this paper is organised as follows. Section 2 describes the shared retrieval architecture underlying all submissions. Sections 3 and 4 detail our IKAT and RAG runs respectively. Section 5 discusses limitations identified in our pre-submission analysis, and Section 6 concludes.

2 System Architecture

All submissions share a common modular backbone. A three-stage retrieval pipeline designed to progressively refine the candidate document set, supported by cross-cutting filtering and reranking components.

2.1 Three-Stage Retrieval Pipeline

Stage 1 - Candidate Generation. The first stage establishes an initial candidate pool. For IKAT, the user's raw utterance is rewritten by an LLM conditioned on the full conversation history and the user's Personal Text Knowledge Base (PTKB), before being issued as a BM25 query. For RAG, complex queries are first decomposed into sub-questions, each of which is processed independently. In both cases, the top-*K* retrieved documents form the input to Stage 2.

Stage 2 - Document Expansion. To improve recall beyond the reach of the original query, we apply Query-by-Document (QbD): each Stage 1 document is used as a BM25 query to retrieve semantically similar neighbours from the collection. This expansion is effective at surfacing documents that share vocabulary with relevant passages but may not have matched the original query formulation.

Stage 3 - Gap Analysis and Refinement. An LLM is presented with the top-*N* documents from Stage 2 and prompted to identify informational gaps: aspects of the user's need that the current result set does not yet adequately address. The model generates multiple targeted queries per gap, each executed via BM25. Results from all sub-queries, together with those from prior stages, are merged using Reciprocal Rank Fusion (RRF) to produce the final ranked list.

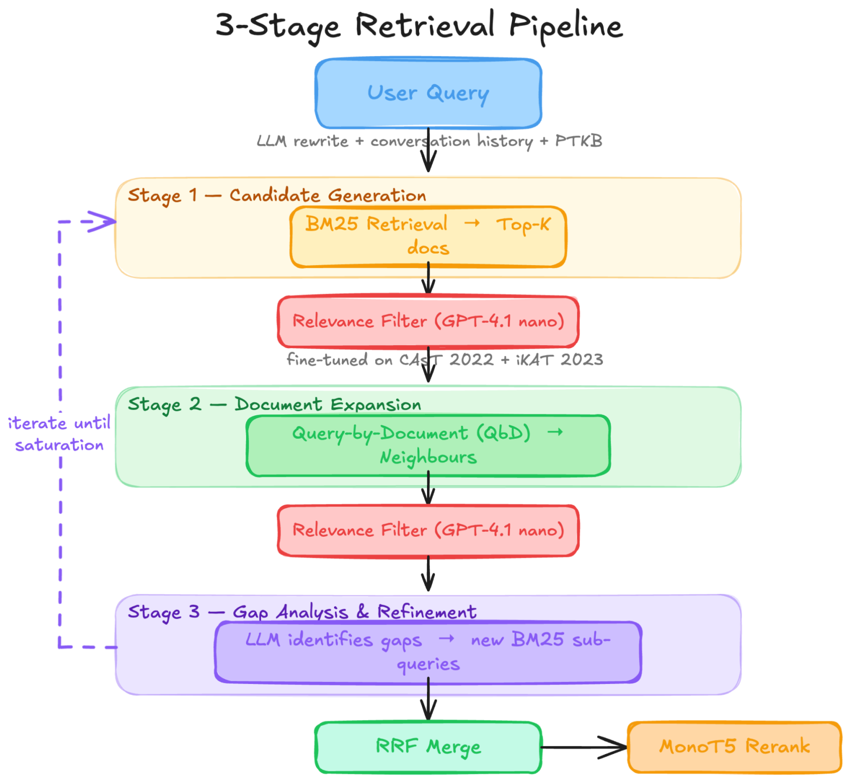
3-Stage Retrieval Pipeline
User Query
LLM rewrite + conversation history + PTKB
Stage 1 — Candidate Generation
BM25 Retrieval → Top-K docs
Relevance Filter (GPT-4.1 nano)
fine-tuned on CAsT 2022 + iKAT 2023
Stage 2 — Document Expansion
Query-by-Document (QbD) → Neighbours
Relevance Filter (GPT-4.1 nano)
Stage 3 — Gap Analysis & Refinement
LLM identifies gaps → new BM25 sub-queries
iterate until saturation
RRF Merge
MonoT5 Rerank

Figure 1: The three-stage retrieval pipeline with relevance filtering between stages.

2.2 Relevance Filtering and Reranking

A central challenge in multi-stage expansion is topic drift: as the candidate set grows through successive expansion and refinement steps, off-topic documents accumulate and can corrupt later stages. To mitigate this, we apply a filtering step between every stage of the pipeline.

The filter is a GPT-4.1 nano model fine-tuned as a binary relevance classifier on human judgments from TREC CAsT 2022 and IKAT 2023. Documents classified as irrelevant are discarded before they can influence subsequent stages. Surviving documents are then reranked using MonoT5, with a reranking budget capped at the top 500 documents.

3 IKAT Track

We participated in both the Passage Ranking/Response Generation task and the Simulation task.

3.1 Passage Ranking and Response Generation

Submissions for the main task apply the architecture from Section 2 directly to the conversational retrieval setting. The four runs vary in the depth of the saturation loop and in the final reranking mechanism.

3.1.1 Submitted Runs

grilllab-larf-finetuned: Baseline run. The three-stage pipeline (Retrieve → Expand → Refine) is executed once with the fine-tuned relevance filter applied between stages. This establishes a performance floor for the saturation experiments below.

grilllab-larf-finetuned-10-rounds: Saturation run, capped at 10 rounds. The Retrieve → Expand loop is executed iteratively, with each round seeded by the documents discovered in the previous one. The loop terminates early upon reaching saturation; i.e., when no new passages are returned.

grilllab-larf-finetuned-22-rounds: Identical to the 10-round configuration but with a higher saturation cap of 22 rounds, testing whether additional iterations continue to surface new relevant material or yield diminishing returns.

grilllab-larf-finetuned-rankllm: Standard single-pass pipeline with MonoT5 replaced by GPT-5 under the RankGPT listwise reranking algorithm, evaluating whether a stronger generative ranker improves precision at the top of the list.

3.2 Simulation Task

For the Simulation Task, we deployed a fully agentic pipeline in which GPT-4.1 acts as the central orchestrator. At each turn, the agent receives the user's query, profile, and conversation history, and decides between two primary operational modes: invoking a retrieval tool or issuing a direct JSON reply. This design simulates realistic interactive assistant behaviour, requiring the system to judge when additional evidence gathering is necessary versus when the current context is sufficient to respond.

3.2.1 Action Schema

The agent is governed by a strict grounding policy: all factual responses must be traceable to retrieved documents, and unsupported generation is prohibited. The full action space is as follows.

Tool actions (information gathering):

- **Search:** The default action for queries requiring factual or current information. The agent generates 3–5 diverse search queries and specifies a target document count informed by the user profile.
- **RefineSearch:** Invoked when prior retrieval results are insufficient. The agent issues new queries and provides an explicit justification explaining why previous retrieval failed to satisfy the need.

Direct reply actions (response generation):

- **Answer:** A concise summary followed by a detailed response synthesised from retrieved documents. This action may only be selected after at least one tool call has been made and search results are present in context. All claims must be cited (e.g., [DOC-1]), and the response must include suggested follow-up questions.
- **Clarify:** Issues a targeted clarifying question to resolve ambiguity in the user's intent before proceeding to retrieval.
- **Converse:** Provides a brief conversational reply for non-factual exchanges (e.g., greetings, acknowledgements) without invoking retrieval.

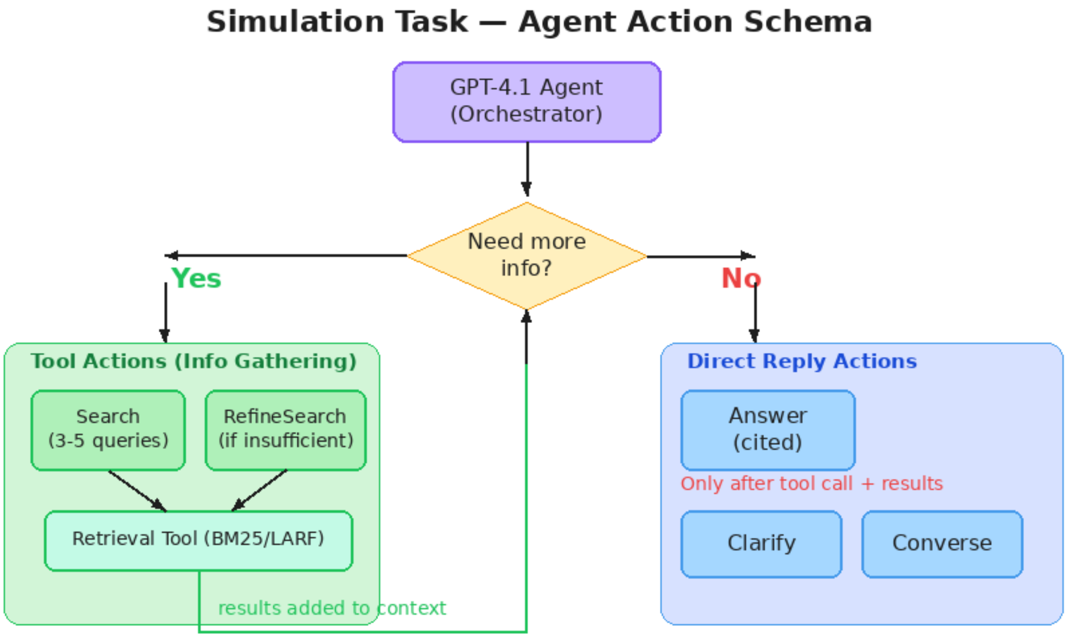
Simulation Task — Agent Action Schema
GPT-4.1 Agent (Orchestrator)
Need more info?
Yes
No
Tool Actions (Info Gathering)
Search (3-5 queries)
RefineSearch (if insufficient)
Retrieval Tool (BM25/LARF)
results added to context
Direct Reply Actions
Answer (cited)
Only after tool call + results
Clarify
Converse

Figure 2: Agent action schema for the Simulation Task.

3.2.2 Submitted Runs

All four runs use GPT-4.1 as the orchestrating agent; they differ in the underlying retrieval tool exposed to the agent.

grilllab-agentic-gpt4.1: The agent is provided with a standard BM25 retriever only, with no expansion or refinement stages. This isolates the contribution of the agentic orchestration from the multi-stage pipeline.

grilllab-agentic-gpt4.1-larf: The agent has access to the full three-stage pipeline (Retrieve → Expand → Refine) as its retrieval tool.

grilllab-agentic-gpt4.1-larf-v2: A modified pipeline with revised stage ordering: Candidate Generation (BM25) → Gap Analysis → Document Expansion. This tests the hypothesis that identifying informational gaps before QbD expansion yields more targeted expansion queries and higher precision.

grilllab-larf-fine-tuned-judge: Standard BM25 retrieval augmented with the fine-tuned binary relevance filter, without expansion or refinement. This isolates the contribution of the filter from the multi-stage pipeline.

4 RAG Track

For the RAG track, all submissions build on the three-stage pipeline and additionally employ query decomposition to handle complex, multi-faceted information needs. A complex query is first decomposed into a set of targeted sub-questions by an LLM; each sub-question is then processed independently through the full Retrieve → Expand → Refine pipeline; and the resulting ranked lists are merged via RRF.

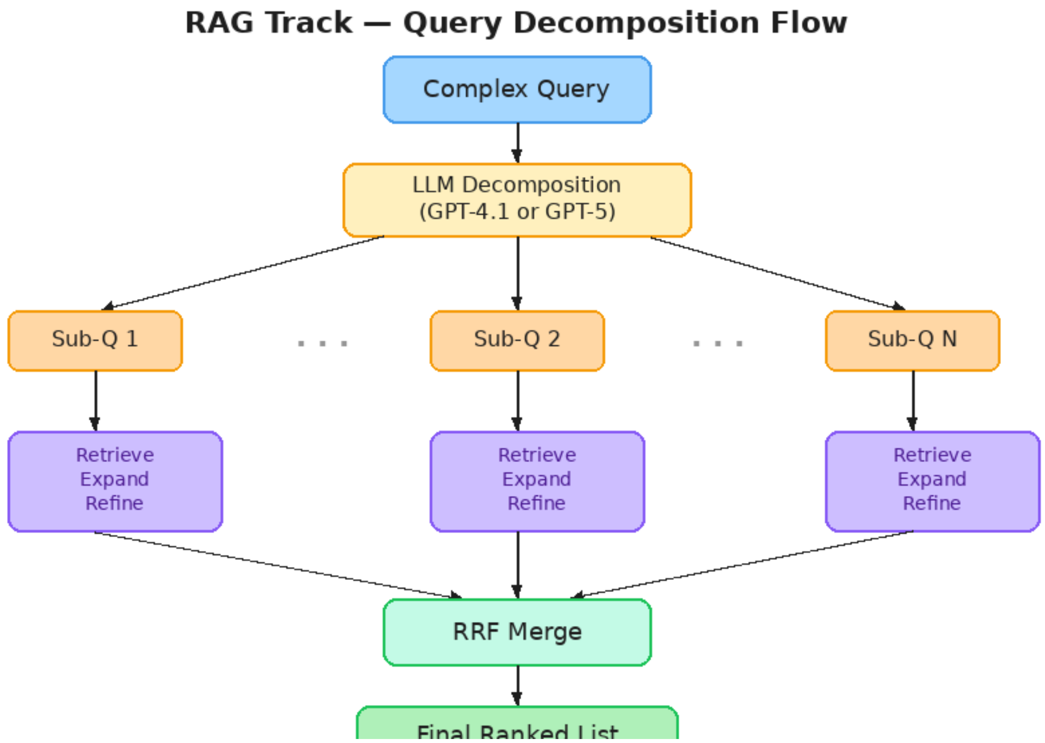

Figure 3: RAG query decomposition — each sub-question runs the full pipeline before RRF merging.

4.1 Retrieval Only Task

grilllab-agentic-gpt4: GPT-4.1 decomposes the input query into sub-questions. Each sub-question is independently processed by the three-stage pipeline, and the result lists are merged using RRF.

grilllab-agent-gpt45: Identical to the above, but decomposition is performed by a GPT-5 class model. This tests whether improved reasoning in the decomposition step yields better sub-questions and consequently better retrieval coverage.

4.2 Full RAG Task

grilllab-agentic-gpt4-generation: Retrieval follows the grilllab-agentic-gpt4 pipeline. Final answer generation is performed by GPT-4.1 conditioned on the retrieved passages.

grilllab-gpt45-gen: Retrieval follows the grilllab-agent-gpt45 pipeline. Final generation is performed by GPT-5, evaluating whether a stronger model better synthesises evidence from complex multi-document contexts.

5 Discussion and Limitations

Our pre-submission analysis identified several limitations that merit discussion.

The most significant concern is the hard reranking budget of 500 documents. In multi-round saturation configurations, particularly the 22-round run, the pipeline aggregates substantially more candidates across all stages before reranking is applied. Enforcing a 500-document cut-off may discard documents that, while not highly ranked in isolation, contain content valuable for subsequent expansion or gap analysis. This could artificially cap the recall ceiling of the saturation approach and understate its potential.

A related issue is the cost of iterative execution. Each additional saturation round incurs retrieval and LLM inference overhead. In practice, the marginal gain from successive rounds is likely to diminish well before the nominal cap is reached. An adaptive stopping criterion, terminating when the gain in new relevant documents per round falls below a threshold, may offer better efficiency without sacrificing coverage.

Finally, the reordered pipeline in grilllab-agentic-gpt4.1-larf-v2, in which gap analysis precedes document expansion, is an architectural hypothesis that has not yet been empirically validated. Performing gap analysis on a smaller initial candidate set introduces the risk that the gap queries are underdetermined if the BM25 results are insufficiently diverse to reveal the full scope of the user's need.

6 Conclusion

The GRILL Lab's TREC 2025 submissions develop and evaluate a saturation-based, agentic retrieval architecture in which LLMs serve as active reasoning components throughout the retrieval pipeline. By interleaving BM25 retrieval with fine-tuned relevance filtering, Query-by-Document expansion, and LLM-driven gap analysis, the system pursues a dynamic, iterative approach to information seeking that continues until the topic is exhaustively covered.

Our participation across four sub-tasks - IKAT Passage Ranking, IKAT Simulation, RAG Retrieval Only, and RAG Full - allows us to evaluate the architecture across complementary settings, from passage-level conversational ranking to end-to-end answer generation over complex queries. We look forward to the official evaluation results and intend to investigate the reranking budget limitation, adaptive saturation stopping, and the effect of pipeline stage ordering in future work.

References

[1] S. E. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. Foundations and Trends in Information Retrieval, 3(4):333-389, 2009.

[2] G. V. Cormack, C. L. A. Clarke, and S. Buettcher. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009), pages 758-759, Boston, MA, USA, July 2009. ACM.

[3] R. Nogueira, Z. Jiang, R. Pradeep, and J. Lin. Document ranking with a pretrained sequence-to-sequence model. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 708-718. Association for Computational Linguistics, 2020.

[4] W. Sun, L. Yan, X. Ma, P. Ren, D. Yin, and Z. Ren. Is ChatGPT good at search? Investigating large language models as re-ranking agents. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023), pages 14918-14937. Association for Computational Linguistics, 2023.

[5] M. Aliannejadi, Z. Abbasiantaeb, S. Chatterjee, J. Dalton, and L. Azzopardi. TREC iKAT 2023: The interactive knowledge assistance track overview. arXiv preprint arXiv:2401.01330, 2024.

[6] M. Aliannejadi, Z. Abbasiantaeb, S. Chatterjee, J. Dalton, and L. 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 2024), pages 2765-2769. ACM, 2024.

[7] P. Owoicho, J. Dalton, M. Aliannejadi, L. Azzopardi, J. R. Trippas, and S. Vakulenko. 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). NIST Special Publication 500-338, 2023.