RAGTUM at SemEval-2026 Task 8: Contextual Query Rewriting and Dense Retrieval for Multi-Turn RAG

Finn Wigger and Maximilian Podolsky and Merle Wilmink and Zelong Peng
Technical University of Munich (TUM)
{finn.wigger, maximilian.podolsky, merle.wilmink, zelong.peng}@tum.de

Abstract

This paper describes the system developed by a team for the TUM practical course Human-Centered Computing: applications in natural language processing, network science, machine learning, and AI for the SemEval MTRAG. Our approach addresses the challenges of multi-turn retrieval-augmented generation (RAG) by combining context-aware query rewriting with a dense retrieval strategy. We employ a pipeline that cleanses noisy corpora and utilizes dense OpenAI embeddings via Milvus for robust retrieval, and leverages Gemini 2.5 flash family of models for standalone query generation and final response synthesis. Our system demonstrates the effectiveness of integrating high-precision retrieval with fact-based generation across diverse domains.

1 Introduction

Retrieval-augmented generation (RAG) has become a standard to improve LLM reliability(Karpukhin et al., 2020). However, evaluating these systems in multi-turn settings introduces complexities such as non-standalone questions and shifting conversation contexts. Recent work further shows that combining retrieval, structured knowledge, and LLM reasoning improves interpretability and robustness in knowledge-intensive tasks (Kolli et al., 2025). To address these challenges, recent benchmarks such as MTRAG(Katsis et al., 2025) and MTRAG-UN(Rosenthal et al., 2026a) have been proposed to systematically evaluate multi-turn RAG systems under realistic conversational conditions. SemEval-2026 task 8: MTRAGEval(Rosenthal et al., 2026b) benchmark addresses this by providing 110 human-generated conversations across four domains: CLAPNQ, FiQA, Govt, and Cloud. We designed solutions for all three subtasks: Retrieval (Subtask A), Generation (Subtask B), and the end-to-end Pipeline (Subtask C).

Our main strategy relies on a **rewrite-retrieve-generate** architecture. We first transform multi-turn user turns into standalone queries using Gemini 2.5 flash. We then perform retrieval using OpenAI's text-embedding-3-large dense embeddings. Finally, we synthesize answers using a relevancy-filtered generation process that prioritizes faithfulness to the retrieved context.

By participating in this shared task, we demonstrated that our dense retrieval and Gemini-based synthesis pipeline is highly effective for multi-turn conversational RAG, consistently placing in the top third of all submissions. In Subtask A (Retrieval), our system achieved an nDCG@5 of 0.4883, ranking 11th out of 38 teams and surpassing the primary ELSER-based baseline (0.4795). In Subtask B (Generation), we secured 11th place among 26 teams with a harmonic mean score of 0.7453, significantly outperforming the GPT-OSS-120b baseline (0.639). Our end-to-end performance in Subtask C (Full RAG) was particularly competitive, ranking 10th out of 29 participants with a score of 0.5421, exceeding the top-performing Qwen-30b-a3b-thinking baseline (0.5366). These quantitative results highlight the robustness of our contextual rewriting module and its ability to maintain faithfulness across turns.

Our contributions are threefold:

- We demonstrate that few-shot LLM-based contextual query rewriting substantially improves dense retrieval performance in multi-turn RAG.
- We introduce a lightweight LLM-based relevance filtering stage optimized specifically for faithfulness metrics.
- We provide a systematic evaluation of rewrite–retrieve–filter–generate pipelines under the MTRAG benchmark.

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 1784–1790

Our code is publicly available at: github.com/mtrag-eval

2 Related Work

2.1 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) has emerged as a robust paradigm for improving the factual reliability of large language models by conditioning generation on externally retrieved evidence (Lewis et al., 2020). By integrating non-parametric memory with generative models, RAG systems reduce reliance on implicit parametric knowledge and mitigate hallucinations. Early neural retrieval approaches such as Dense Passage Retrieval (DPR) (Karpukhin et al., 2020) demonstrated that dense retrievers outperform traditional sparse methods like BM25 in open-domain question answering by embedding queries and documents into a shared semantic space. Subsequent work has explored late-interaction architectures such as ColBERT (Khattab and Zaharia, 2020) and hybrid dense-sparse retrieval methods (Formal et al., 2021), which combine lexical matching with semantic similarity to improve robustness across domains. These advances have established dense retrieval as a strong backbone for knowledge-intensive generation tasks, particularly when paired with large language models. Unlike prior work focusing primarily on single-turn question answering, our approach explicitly integrates contextual query rewriting with dense retrieval under a shared-task multi-turn evaluation setting.

2.2 Conversational and Multi-Turn Retrieval

Multi-turn retrieval introduces additional complexity compared to single-turn question answering. Conversational queries frequently contain ellipsis, pronouns, and implicit references to prior turns, making them unsuitable for direct retrieval. Elgohary et al. (Elgohary et al., 2019) introduced the CANARD dataset to study question rewriting in conversational QA, demonstrating that reformulating context-dependent utterances into standalone queries significantly improves downstream retrieval performance. The MTRAG benchmark (Katsis et al., 2025) highlights the importance of resolving conversational dependencies before retrieval in multi-domain settings. Our work builds upon these findings by integrating few-shot prompted rewriting into a dense retrieval pipeline.

2.3 Faithfulness and Hallucination Mitigation

Hallucination remains a central challenge in large language model deployment. Studies on factual consistency in generation tasks (**??**Üyük et al., 2024) have shown that generative models often produce plausible but unsupported statements when insufficiently grounded in evidence. Retrieval augmentation has been shown to reduce hallucinations in dialogue and question answering systems (Shuster et al., 2021). However, retrieval noise can still propagate into generation if irrelevant passages are included. Consequently several approaches introduce reranking or filtering mechanisms to improve evidence precision before generation. Our relevance filtering module follows this line of work by introducing an automated LLM-based document scoring that selectively retains passages with clear semantic overlap, explicitly optimizing for faithfulness-oriented evaluation metrics.

3 Background

The MTRAG task utilizes four document corpora (Katsis et al., 2025): Input consists of multi-turn conversation histories where questions often refer back to previous turns. Systems must output relevant documents (Retrieval) or natural language responses (Generation).

Corpus	Domain	# of Passages
ClapNQ	Wikipedia	183,408
FiQA	Finance	61,022
Govt	Government	49,607
Cloud	Technical doc.	72,442

Table 1: Dataset overview

4 System Overview

Our system consists of four primary components: query-rewriting, dense retrieval, relevance filtering, and context-aware generation.

4.1 Query rewriting

In order to retrieve the best possible passages based on the query we chose to rewrite the multi-turn conversation into a standalone and context independent query, as shown to be effective by Ma et al.. Therefore, we process the full dialogue, containing the user turns and the generated answers. This substantially improved the retrieval performance, as we saw improvements of around 10%. This lifted our

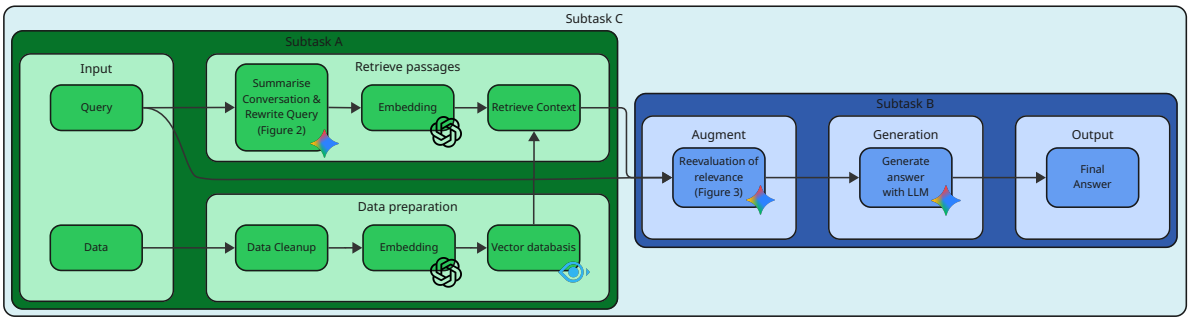

Figure 1: The full MTRAG pipeline

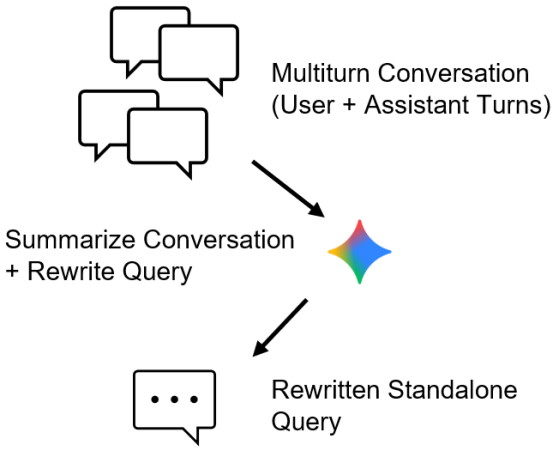

Figure 2: Query rewriting

retrieval Recall @5 results from 0.53 to 0.58 on test data and made it compatible with top retrieval methods.

We designed a query that instructs the LLM to take a look at the full conversation history, resolve the pronouns used, remove conversational fillers and polite phrases and include relevant entities and details from the whole conversation. Lastly the LLM should only return the rewritten query text which is then passed to the retrieval (Figure 2). To improve LLM performance we used few shot prompting, providing two examples. If the question is the first in the current dialogue, the LLM was asked to return the query with no changes made to it.

4.2 Dense Retrieval

To retrieve semantically relevant passages from the documents, we chose a dense similarity retrieval, as it outperformed sparse retrieval consistently. We embed the (rewritten) query with the same embedding model we used for the initial embedding of the passages, text-embedding-3-large by OpenAI. This gave us a high dimensional vector embedding of the query. Based on the collection specified by the given query, we know in which of the four corpora we need to look for the relevant passages. In the corresponding collection we search the semantically closest passages based on the inner product.

$$\mathbf{q} \cdot \mathbf{d} = \sum_{i=1}^{n} q_i d_i$$

4.3 Relevance Filtering

Building on the previously described query rewriting process, the Subtask B pipeline is specifically optimised to address the faithfulness score. Because irrelevant documents negatively influence this score and increase the risk of model hallucinations, we implemented a rigorous filtering stage using Gemini 2.5 flash. Each retrieved document is evaluated on a 5-point relevance scale; only those demonstrating clear topical overlap (score ≥ 2) are retained for the final response. This automated filtering ensures the model generates a concise answer strictly grounded in relevant evidence (Figure 3).

4.4 Answer Generation

In the context of retrieval-augmented generation (RAG), we employ an enhanced query formulation strategy that integrates the rewritten user query with retrieved document passages. This combined representation is subsequently utilized to perform the generation task using the Gemini 2.5 Flash model. The generated output must satisfy stringent feasibility criteria, which include demonstrating a comprehensive understanding of the user's original intent and ensuring strict fidelity to the content of the retrieved documents. If no or insufficient information was retrieved to generate an answer, the system was prompted to reply: "I don't know".

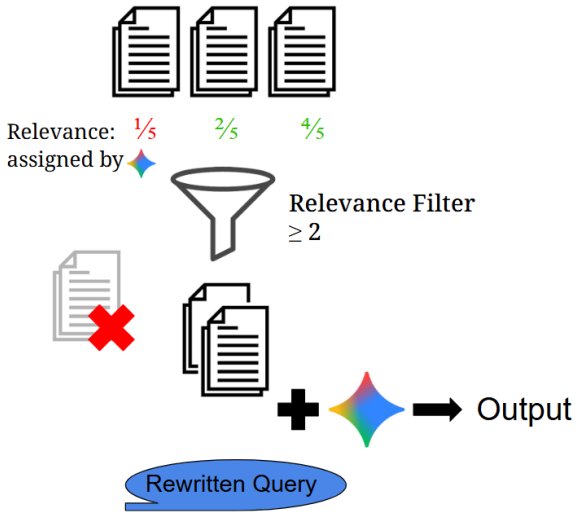

Figure 3: Relevance Filter

5 Experimental Setup

5.1 Data Splits

As we did not train a model on given data, the given conversations were used for testing retrieval performance. For generations tasks B and C, the only option was to use LLM as a judge.

5.2 Models

For tasks involving LLMs we used gemini-2.5-flash (Google, 2024). We utilised OpenAI embedding-3-large (OpenAI, 2024) for embedding corpora and queries.

5.3 Hyperparameters

For dense retrieval, we retrieve the top-k passages with $k = 5$. Similarity is computed using the inner product over embedding vectors. Query rewriting and answer generation are performed using the gemini-2.5-flash model with default decoding parameters provided by the API.No explicit temperature, top-p, or maximum token limits were manually configured. In the relevance filtering stage, the passages that receive a relevance score of ≥ 2 are retained. The threshold was empirically selected to balance recall and faithfulness, ensuring that clearly irrelevant documents are excluded while preserving potentially useful contextual evidence.

5.4 Infrastructure and Cost

Vector storage and similarity are handled using Milvus as the backend database. Embedding and language model inference are conducted via API access to proprietary models. All experiments were performed without dedicated GPU infrastructure, relying instead on cloud-based inference endpoints. We acknowledge that reliance on proprietary APIs introduces constraints regarding reproducibility and cost scalability. Future work could explore open-weight alternatives to reduce financial and infrastructural barriers.

6 Results

Table 2 summarizes the performance of our proposed system across three core evaluation tasks: Retrieval (Task A), Generation (Task B), and Retrieval-Augmented Generation (Task C). Our model consistently outperformed the established top-performing baselines in every category. Specifically, in Task A, we achieved an $nDCG@5$ of 0.4883, placing 11th out of 38 participants and surpassing the GPT-OSS-20b baseline. In Task B, the system reached a harmonic mean of 0.7453, showing a significant improvement over the 0.6390 baseline and ranking 11th out of 26. Finally, in the integrated RAG task (Task C), our approach attained a score of 0.5421, ranking 10th out of 29 submissions. These results demonstrate the robustness of our framework, as it maintains competitive performance and beats the Qwen-30b-a3b-thinking baseline in complex, end-to-end RAG scenarios.

Table 2: Model Performance Comparison

Task	Metric	Score	Rank
A:	nDCG@5	0.4883	11/38
B:	Harm. Mean*	0.7453	11/26
C:	Harm. Mean*	0.5421	10/29

*Harmonic mean of RB_{agg}, RL_F, and RB_{llm}.

7 Discussion

Our findings reveal the intrinsic end-to-end sensitivity of Retrieval-Augmented Generation (RAG) systems, wherein overall performance depends critically on the synergistic interplay among all pipeline components. Variations in query rewriting strategies, ranking algorithms, and embedding model selection—whether sparse or dense—can significantly influence downstream generation quality, underscoring the necessity for holistic optimization rather than isolated component tuning.

Notably, we observe that the OpenAI embedding model achieves optimal retrieval performance specifically when paired with Gemini-based query

rewriting, suggesting an emergent complementarity between these components.

Future research should extend benchmarking to encompass additional sparse and dense embedding models, while systematically evaluating diverse LLMs to decouple their respective contributions to query rewriting and final response generation. Such investigations will inform more principled, synergistic RAG architectures.

8 Limitations

Our approach relies heavily on proprietary large language models for rewriting, filtering, and generation. As a result, full reproducibility depends on continued API availability and consistent model behavior over time. While we conducted comparative experiments with sparse and hybrid retrieval strategies, dense retrieval consistently outperformed alternative configurations within the MTRAG benchmark domains. However, our evaluation remains limited to this specific shared-task setting. The relative performance of retrieval paradigms may differ in other domains, languages, or document distributions. The relevance filtering stage introduces additional computational overhead and latency, which may limit applicability in real-time settings. Furthermore, although filtering improves faithfulness, it may occasionally discard partially relevant documents, potentially reducing recall. Finally, our system has not been evaluated under strict latency or cost constraints, and no statistical significance testing was conducted due to the evaluation framework of the shared task.

9 Conclusion

The TUM-MTRAG system demonstrates that a robust multi-turn RAG pipeline requires more than just high-capacity LLMs; it necessitates data cleaning, intelligent query rewriting to resolve conversational dependencies and an evaluation of retrieved documents to reduce noise and hallucinations. We developed a reliable pipeline, that remains heavily dependent on LLM-usage and a models performance though.

10 Ethical Considerations

Our system uses large proprietary language models, which may encode societal biases present in their training data. Although our retrieval reduces hallucinations, it does not fully eliminate potential bias amplification during generation. The reliance on external APIs raises concerns regarding transparency, long-term accessibility, and environmental cost associated with large-scale model inference. Additionally, automated relevance filtering based on LLM judgments may introduce unintended systematic biases in document selection. While our approach aims to improve factual reliability in multi-turn RAG systems, users should remain aware that generated outputs may still contain inaccuracies. Deployment in high-stakes domains such as government or finance should therefore include oversight mechanisms.

References

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.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. Splade: Sparse lexical and expansion model for first stage ranking.

Google. 2024. Gemini 2.5 flash. Google Vertex AI Model Documentation.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Yannis Katsis, Sara Rosenthal, Kshitij Fadnis, Chulaka Gunasekara, Young-Suk Lee, Lucian Popa, Vraj Shah, Huaiyu Zhu, Danish Contractor, and Marina Danilevsky. 2025. Mtrag: A multi-turn conversational benchmark for evaluating retrieval-augmented generation systems. *Preprint*, arXiv:2501.03468.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert.

Shaghayegh Kolli, Richard Rosenbaum, Timo Cavelius, Lasse Strothe, Andrii Lata, and Jana Diesner. 2025. Hybrid fact-checking that integrates knowledge graphs, large language models, and search-based retrieval agents improves interpretable claim verification. In *Proceedings of the 9th Widening NLP Workshop*, pages 106–115, Suzhou, China. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Xiao Ma, Mingchen Chen, Wenhao Wang, Yuxuan Wang, Beidi Chen, and Tianqi Chen. 2023. Query rewriting for retrieval-augmented large language models.

OpenAI. 2024. text-embedding-3-large. OpenAI Model Documentation.

Sara Rosenthal, Yannis Katsis, Vraj Shah, Lihong He, Lucian Popa, and Marina Danilevsky. 2026a. Mtrag-un: A benchmark for open challenges in multi-turn rag conversations. *Preprint*, arXiv:2602.23184.

Sara Rosenthal, Vraj Shah, Yannis Katsis, and Marina Danilevsky. 2026b. Semeval-2026 task 8: Mtrageval: Evaluating multi-turn rag conversations. In *Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval-2026)*, San Diego, California. Association for Computational Linguistics.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation.

Cem Üyük, Danica Rovó, Shaghayeghkolli, Rabia Varol, Georg Groh, and Daryna Dementieva. 2024. Crafting tomorrow's headlines: Neural news generation and detection in English, Turkish, Hungarian, and Persian. In *Proceedings of the Third Workshop on NLP for Positive Impact*, pages 271–307, Miami, Florida, USA. Association for Computational Linguistics.

A LLM Prompt Templates

A.1 Query Rewriting Prompt

```
Instructions:
    Rewrite the LAST USER turn into a
        standalone, keyword-rich search
        query.
    - Use the full conversation history
        (both user questions and
        assistant responses) to
        understand context.
    - Resolve all pronouns (it, they,
        that, this) using information
        from previous turns.
    - Remove conversational filler and
        polite phrases.
    - Include all relevant details and
        entities mentioned in the
        conversation.
    - If the last turn is already
        standalone, return it as-is.
    - Output ONLY the rewritten query
        text.

    Examples:

    Example 1:
    1. User: "Who is the CEO of Google?"
    2. Assistant: "Sundar Pichai is the
        CEO of Google."
    3. User: "When did he take over?"
    Rewritten query: When did Sundar
        Pichai become CEO of Google?


    Example 2:
    1. User: "What are dialog nodes?"
    2. Assistant: "Dialog nodes are
        various types including the
        Welcome node and Anything else
        node. You can create custom
        nodes by adding a condition..."
    3. User: "What are intents?"
    4. Assistant: "Intents are the
        purposes or goals expressed in a
         customer's input, such as
        answering a question or
        processing a bill payment..."
    5. User: "How is it created?"
    Rewritten query: How is a dialog
        node created?

    Current Conversation:
    {conversation_text}

    Rewritten query:
    """
```

A.2 Relevance Filtering Prompt

```
    Evaluate the relevance of the
        following Document to the User
        Query.

    Query: "{query}"
    Document: "{text}"

Relevance Scale:
1: Totally Unrelated - No topical
    overlap.
2: Weakly Related - Mentions
    keywords or broad context, but
    no direct answer.
3: Partially Relevant - Provides
    context that helps understand
    the answer.
4: Relevant - Contains significant
    pieces of the answer.
5: Highly Relevant - Directly and
    fully addresses the query.

Instructions:
1. Provide the integer score.
2. Provide a brief one-sentence
    reasoning.


Response Format:
Score: <integer>
Reasoning: <brief explanation>"""
```