IIMAS-RAG at SemEval-2026 Task 8: Hybrid Sparse-Dense Retrieval and Answerability-Conditioned Generation for Multi-Turn RAG

Vania Raya-Rios[1], Helena Gómez-Adorno[1], Leon Hecht[2], Pedro Vázquez-Osorio[2], Erick Fabián-Sandoval[1], Jesus Vázquez-Osorio[2], Diego Hernández-Bustamante[3]

[1]Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas,
[2]Posgrado en Ciencia e Ingeniería de la Computación,
[3]Facultad de Estudios Superiores Aragón,
Universidad Nacional Autónoma de México (UNAM)

Correspondence: helena.gomez@iimas.unam.mx

Abstract

This paper presents the IIMAS-RAG system submitted to SemEval-2026 Task 8, which evaluates multi-turn retrieval-augmented generation (RAG) conversations. Our system is a modular pipeline composed of three stages: (1) LLM-based query rewriting to transform conversational history into standalone queries, (2) hybrid sparse–dense retrieval combining SPLADE and Voyage-3-large via Reciprocal Rank Fusion (RRF), and (3) answerability-conditioned generation using GPT-4.1. In Subtask A (Retrieval), our system ranked 4th out of 38 teams (nDCG@5 = 0.5445), demonstrating the robustness of the hybrid retrieval strategy in specialized domains. On Subtask C (Full RAG), we ranked 13th out of 29 teams (composite = 0.5397). Ablation experiments show that LLM-based query rewriting is the main driver of retrieval performance, yielding a +16.3% relative gain in nDCG@10 over the hybrid baseline without rewriting, while domain-specific prompt variants provide only localized gains on specialized corpora. Generative performance remains sensitive to low-context and partially answerable turns, where the user query lacks sufficient grounding information and the model struggles to either abstain or provide a properly qualified partial answer, explaining the performance gap between retrieval and final synthesis. Our code is available at `https://github.com/PLN-disca-iimas/mtrag_semeval2026`.

1 Introduction

This paper presents our participation in the SemEval-2026 Task 8: Evaluating Multi-Turn RAG Conversations (MTRAGEval), a shared task designed to advance the state of conversational AI systems that combine information retrieval with large language model (LLM) generation (Katsis et al., 2025). The task addresses a critical challenge in modern AI assistants: providing trustworthy, evidence-grounded responses in multi-turn conversations where users build upon previous exchanges to seek information. The task is organized into three subtasks: Subtask A (Retrieval) focuses on identifying relevant passages for a given conversation turn; Subtask B (Generation with Reference Passages) evaluates answer generation when gold-standard passages are provided; and Subtask C (Full RAG) requires systems to first retrieve passages and then generate answers based on those retrieved passages. These tasks cover English-language conversations, drawing on the MTRAGEval benchmark across four domains as described in Rosenthal et al. (2026a).

Our approach employs a modular, multi-stage pipeline designed to address the challenges of conversational RAG. For retrieval (Subtask A and the retrieval component of Subtask C), we implemented a hybrid retrieval strategy that combines sparse and dense representations with query rewriting. Recognizing that conversational queries often contain coreferences and elisions that degrade retrieval performance when used directly, we first employ an LLM to rewrite each conversational turn into a standalone, self-contained query. This rewritten query is then used to retrieve candidate passages using two complementary approaches: SPLADE (Formal et al., 2021), and Voyage-3-large (Voyage AI, 2025). The resulting ranked lists are fused using Reciprocal Rank Fusion (RRF) to produce a unified ranking, from which we select the top-k passages. Each domain was indexed independently, resulting in four separate retrieval indices.

For the generation components (Subtasks B

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 2744–2753

and C), we developed a pipeline that incorporates answerability classification as a critical preprocessing step. During development, we leveraged the reference training data, which provides gold-standard passages and target answers, to calibrate our generator prompts and establish the expected answer style, verbosity, and level of detail. At inference time, the system first evaluates whether the current question can be answered using only the retrieved evidence and conversation history. We implemented a precision-focused classifier using Command-A-03-2025 (Cohere, 2025) that analyzes the retrieved context, dialogue history, and current question to output one of four labels: `ANSWERABLE`, `UNANSWERABLE`, `PARTIAL`, or `CONVERSATIONAL`. This predicted label then conditions the GPT-4.1 (OpenAI, 2025) generator, enabling appropriate responses such as declining to answer unanswerable questions or acknowledging partial information.

Our retrieval component ranked 4th out of 38 teams on Subtask A (nDCG@5 = 0.5445), while the full RAG pipeline ranked 13th out of 29 on Subtask C. This gap suggests that retrieval was strong, but final answer synthesis remained sensitive to low-context and partially answerable turns. Our code is available at `https://github.com/PLN-disca-iimas/mtrag_semeval2026`.

2 Background

Task and Data. The MTRAGEval shared task (Rosenthal et al., 2026b,a) provides 110 English conversations (7.7 turns on average) grounded in four domain corpora: ClapNQ (general knowledge from Wikipedia; 183 408 passages), Cloud (technical documentation; 72 439 passages), FiQA (financial advice from StackExchange; 49 607 passages), and Govt (regulatory content from .gov/.mil domains; 72 422 passages). The system receives the full conversation history and the current user question. In Subtask A, the goal is to retrieve relevant passages (binary relevance judgments). In Subtask B, gold passages are provided and only generation is evaluated. In Subtask C, the system must both retrieve and generate a grounded response ($\leq$150 words). We participated in all three subtasks.

Related Work. Retrieval has evolved from lexical matching with BM25 (Robertson and Zaragoza, 2009) to dense bi-encoder models such as DPR (Karpukhin et al., 2020), which encode queries and passages into a shared embedding space. However, dense retrievers degrade on out-of-domain data with specialized vocabulary (Thakur et al., 2021), motivating learned sparse models like SPLADE (Formal et al., 2021) that combine the efficiency of inverted indices with neural term expansion. This is particularly relevant for the technical documentation and regulatory content in MTRAGEval, where domain-specific terminology often leads to the *vocabulary mismatch* problem in standard dense retrievers (Thakur et al., 2021).

Hybrid retrieval fuses sparse and dense signals to compensate for their complementary weaknesses. Reciprocal Rank Fusion (Cormack et al., 2009b) effectively integrates scores from heterogeneous sources (lexical and neural) without requiring supervised calibration, making it a robust choice for multi-domain retrieval.

In multi-turn settings, conversational queries contain coreferences and elisions that degrade retrieval when used verbatim. Elgohary et al. (2019) showed that rewriting such queries into standalone forms significantly improves downstream retrieval, an approach also validated in the TREC CAsT track (Dalton et al., 2020). More recently, HyDE (Gao et al., 2023) proposed generating a hypothetical passage as a query proxy, bypassing explicit rewriting. However, in specialized technical domains, explicit rewriting remains a more reliable strategy to mitigate the risk of generative hallucinations associated with hypothetical passage proxies (Gao et al., 2023).

RAG pipelines (Lewis et al., 2020) combine retrieval with generation, but most published systems are evaluated on single-turn queries. MT-RAG extends this to multi-turn conversations, making query handling a central design challenge.

3 System overview

Our system is designed as a modular multi-turn RAG pipeline that addresses both conversational context handling and domain-specific grounding.

Figure 1 presents the end-to-end architecture. The following subsections describe each component in detail, including the evaluated configurations and the final system used for submission.

3.1 Retrieval

Our retrieval system follows a multi-stage pipeline: (1) the conversational query is rewritten into a standalone form using an LLM, (2) the rewritten query

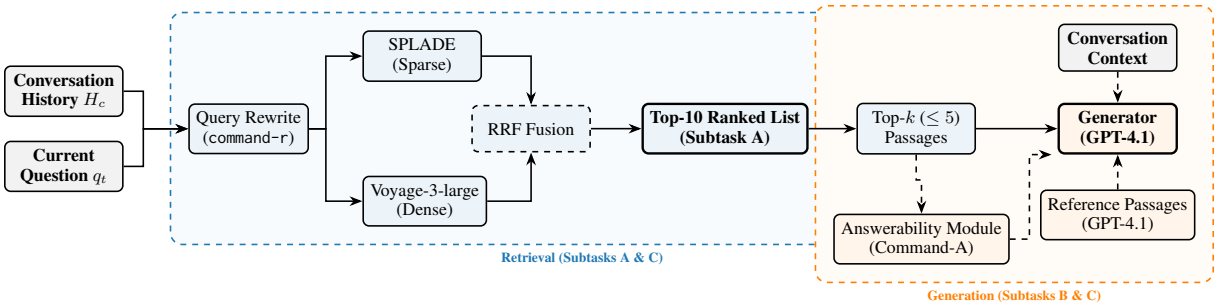

Figure 1: End-to-end IIMAS-RAG architecture. Solid arrows trace the main pipeline: inputs → query rewrite → SPLADE / Voyage-3 → RRF fusion → top-10 ranking (**Subtask A** output) → top-k passage selection → generator. For **Subtask C**, the generator also receives the conversation context (H_c, q_t); an answerability module classifies the query as answerable, unanswerable, partial, or conversational, and its decision signal conditions the generator (dashed path). For **Subtask B**, gold reference passages are supplied directly to the generator.

is used to retrieve candidates from both a sparse and a dense model, (3) the resulting ranked lists are fused via Reciprocal Rank Fusion (RRF), and (4) the top-k passages are returned. Each domain was indexed independently. The following paragraphs describe each component and the experiments conducted to select the final configuration.

Embedding Models. We evaluated five retrieval models spanning three paradigms: (i) **BM25** (k_1=1.2, b=0.75) using the `rank_bm25` library (BM25Okapi)[1] (Robertson and Zaragoza, 2009); (ii) **SPLADE** (`naver/splade-cocondenser-ensembledistil`) (Formal et al., 2021), a learned sparse model with neural term expansion; (iii) **BGE-v1.5** (`BAAI/bge-base-en-v1.5`, 768 dim.); (iv) **BGE-m3** (`BAAI/bge-m3`, 1024 dim.); (Chen et al., 2024); and (v) **Voyage-3-large** (1024 dim., API) (Voyage AI, 2025). Dense embeddings were indexed with FAISS (Johnson et al., 2021) using exact inner-product search (`IndexFlatIP`) over L2-normalized vectors. BM25 and SPLADE use inverted indices. Each model independently retrieved the top 300 candidates per query.

Retrieval Strategies. We evaluated the following strategies for combining and enhancing retrieval:

- **Hybrid Retrieval:** Ranked lists from SPLADE and Voyage-3-large were fused using Reciprocal Rank Fusion (RRF) (Cormack et al., 2009a):

$$\mathrm{RRF}(d) = \sum_{s=1}^{N} \frac{1}{k + \mathrm{rank}_s(d)}, \qquad (1)$$

where $N = 2$ for SPLADE and Voyage-3-large, and $k = 60$.

- **HyDE and Multi-query baselines:** We evaluated HyDE (Gao et al., 2023) and multi-query expansion as alternative query enhancement strategies. Both underperformed explicit standalone rewriting: HyDE reached 0.383 nDCG@10 and multi-query reached 0.401, compared with 0.528 for the best rewrite-based hybrid configuration.

Query Rewriting. We utilized `command-r-08-2024` to transform the conversation history H and the current question Q into an optimized standalone query Q', resolving coreferences and aligning vocabulary with the target domain (Elgohary et al., 2019):

$$Q' = \mathrm{LLM}(p, H, Q) \qquad (2)$$

Through iterative optimization of the prompt p, we developed three rewriting variants: V1, a generic standalone rewriting prompt; V2, which adds domain-specific length and vocabulary-preservation rules; and V3, which further adds few-shot examples from the training split. Full prompt details are reported in Appendix C. All three variants substantially outperform HyDE and multi-query, yielding a +35.1% relative improvement over HyDE and a +85.3% gain over raw-history baselines (0.406 vs. 0.219 nDCG@5). Differences between prompt versions are small and not statistically significant (H4: $p = 0.642$); V2 achieves the highest macro-average (0.520) while V3 provides marginal per-domain gains on Cloud and FiQA. The final submission used V3 rewrites, although the

[1] https://github.com/dorianbrown/rank_bm25

small and non-significant difference with V2 suggests that most gains come from standalone rewriting rather than from additional domain-specific prompt engineering. Based on these results, our final system employs a **Hybrid SPLADE + Voyage-3-large** architecture with LLM-based query rewriting.

Reranking. We evaluated `bge-reranker-v2-m3` over the top-100 RRF candidates, but it degraded nDCG@5 by -0.029 and increased inference cost. We therefore excluded reranking from the final submission.

3.2 Generation

The generation component adopts a RAG architecture where the retrieval module provides the evidence set and a GPT-4.1 (OpenAI, 2025) generator produces the final response. The generator is guided by a structured prompting scheme that enforces factual, neutral, and concise outputs strictly grounded in the retrieved contexts and consistent with the dataset's style. An answerability evaluation stage precedes generation to determine whether the question can be addressed with the available evidence, thereby reducing hallucinations and ensuring that outputs remain coherent, faithful to the retrieved information, and suitable for benchmarking in shared RAG tasks.

Answerability Module. Before generation, the system evaluates whether the current question can be answered using only the retrieved evidence and conversation history. We implemented a precision-focused classifier prompt that clearly defines each label and specifies a two-step decision process where the model should generate candidate labels with brief justifications and then select the final label. This prompt is used with Command-A-03-2025. The model receives the retrieved context, dialogue history, and new question, and outputs one of four labels: `ANSWERABLE`, `UNANSWERABLE`, `PARTIAL`, or `CONVERSATIONAL`. This modular design reduces hallucination risk when evidence is insufficient.

Generation Module. At evaluation time, the system uses the passages returned by the retrieval component (instead of the oracle passages from training). The predicted answerability label conditions the generator behavior. If the query is marked unanswerable, the system returns a fixed fallback response. Otherwise, the generator produces a concise factual answer following strict style constraints learned from the training distribution. The prompt emphasizes neutral tone, minimal verbosity, controlled use of context, and strict avoidance of unsupported claims. Full prompt details are available at Appendix C.4.

4 Experimental Setup

4.1 Retrieval

Evaluation protocol. All retrieval experiments were conducted on the development split of the four datasets. For each configuration, scores were averaged across the four datasets. We report nDCG@10 as the primary retrieval metric and Recall@10 as a complementary measure of coverage.

Evaluation Metrics. The official evaluation for Subtask A uses nDCG@5. For Subtasks B and C, the official metric is the harmonic mean of three generation quality scores: RB_{alg} (algorithmic reference-based similarity), RL_{F} (ROUGE-L F-measure), and RB_{llm} (LLM-based reference comparison).

4.2 Generation

Setup. Our generation experiments used the `reference.jsonl` training set (containing ideal reference passages and target answers) to calibrate the generator prompt. At inference time, we evaluated each generator model under two passage conditions: **Ref** (gold-standard passages from `reference.jsonl`) and **RAG** (passages retrieved by our system), and with and without the answerability classifier.

Evaluated Models. We benchmarked four LLMs for the generation stage:

- **GPT-4.1:** Selected for final submission due to superior faithfulness and instruction-following in technical domains.
- **Command-A-03-2025:** Utilized for the answerability logic.
- **Qwen-2.5-7B:** Evaluated as a local deployment baseline for low-latency scenarios.
- **DeepSeek-R1:** Evaluated via Ollama to test reasoning-heavy generation in specialized corpora.

Model	Strategy	nDCG@10	R@10
Elser†	Rewrite	0.540	0.640
BM25	Std	0.250	0.240
SPLADE	Std	0.480	0.520
BGE-m3	Std	0.460	0.460
BGE-v1.5	Std	0.440	0.450
Voyage-3-large	Std	0.440	0.450
SPLADE+BGE-v1.5	RRF	0.435	0.530
SPLADE+Voyage-3	RRF	0.454	0.540
SPLADE+Voyage-3	RRF+HyDE	0.383	0.480
SPLADE+Voyage-3	RRF+MultiQ	0.401	0.500
SPLADE+BGE-v1.5	RRF+Rewrite	0.481	0.590
SPLADE+Voyage-3	RRF+Rewrite V1	0.510	0.610
SPLADE+Voyage-3	RRF+Rewrite*	**0.528**	**0.640**

Table 1: Retrieval results on the development set (nDCG@10, Recall@10), averaged over four domains. Standard: last user turn as query. *Final V3 rewriting prompt. †Shared task organizer baseline (Elastic Learned Sparse Encoder with query rewriting).

Hyperparameters. All generative models were queried via API or local inference with a temperature of $T = 0.3$ and $top\_p = 0.9$ to reduce stochastic variance and limit hallucinations in fact-heavy responses.

4.3 Reproducibility.

Full hyperparameters and hardware details are provided in Appendix A.

5 Results

This section presents the experimental results across the three subtasks. We first report retrieval effectiveness, followed by generation performance, official test rankings, and error analysis.

5.1 Retrieval

Table 1 reports retrieval performance for different model and strategy combinations on the development set.

Impact of Query Rewriting. Table 2 breaks down nDCG@10 by domain for the SPLADE+Voyage-3 hybrid system under different query-handling strategies. Full rewriting prompts are reported in Appendix C.

Statistical Validation. Key comparisons were validated with paired Wilcoxon signed-rank tests and Holm–Bonferroni correction. The main conclusion is that hybrid retrieval with rewriting significantly outperforms the individual components, whereas differences across rewriting prompt variants are small and generally not significant after correction. Full results, assumptions, effect sizes, and clarifications are reported in the Appendix B.

Domain	HyDE	MQ	V1	V2	V3
ClapNQ	0.485	0.502	0.632	**0.639**	0.633
Cloud	0.331	0.381	0.449	0.474	**0.489**
FiQA	0.333	0.284	0.385	0.410	**0.415**
Govt	0.383	0.437	**0.571**	0.558	0.535
Mean	0.383	0.401	0.510	**0.520**	0.518

Table 2: nDCG@10 by domain across query strategies in the hybrid SPLADE+Voyage-3 system. V1–V3: rewriting prompt versions (generic to domain-specific). MQ: multi-query. Best per row in **bold**.

Retrieval Errors. Preliminary experiments on a subset of 777 MT-RAG queries showed that retrieval errors were driven primarily by *query ambiguity* in extremely short inputs (≤ 3 words; e.g., "fees?", "branch?", "refund status?") and by limitations in the corpus's geographic coverage, rather than by topic drift in long conversations. For example, in a query such as "I am in Asia, not in the UK" when asking about credit card fraud protection, the rewriter correctly preserved the user's geographic intent, but retrieval still returned documents about U.S. and U.K. regulations, suggesting that the benchmark corpus—particularly in FiQA and Govt—contains limited coverage of Asian regulatory contexts.

5.2 Generation

Table 3 presents the results of the generated answers under different experimental configurations on the development set. Full generation prompts and examples of conversations are reported in the Appendix C.4.

	RLF		RB_{llm}		RB_{alg}	
	Ref	RAG	Ref	RAG	Ref	RAG
Without answerability classifier						
GPT-4.1	**0.81**	0.75	**0.77**	0.72	**0.45**	**0.41**
Command-A-03-2025	0.79	**0.77**	0.75	**0.74**	0.41	0.39
Qwen 2.5 7B	0.68	0.67	0.66	0.68	0.34	0.33
DeepSeek	0.55	0.56	0.59	0.59	0.41	0.37
With answerability classifier						
GPT-4.1	**0.69**	**0.70**	**0.64**	**0.61**	0.43	0.40
Command-A-03-2025	0.67	0.68	0.62	0.59	**0.45**	**0.42**
Qwen 2.5 7B	0.60	0.61	0.58	0.56	0.32	0.30
DeepSeek	0.52	0.53	0.55	0.54	0.38	0.35

Table 3: Generation results under different experimental configurations. **Ref** uses the reference.jsonl file (ideal passages), while **RAG** uses passages from RAG.json retrieved by our system. Results are reported with and without the answerability classifier. Per column, the best result is in **bold** and the second best is underlined.

Error Analysis. Manual inspection revealed that the answerability module was overly conservative, frequently classifying partially answerable questions as `UNANSWERABLE`. This triggered the fixed fallback response even when relevant information was present. While this design choice successfully prevented hallucinations, it also led to systematic abstention on turns where a partial answer would have been appropriate, penalizing the system on automatic metrics.

5.3 Official Test Results

Table 4 reports the official test set results and our ranking on each subtask.

Subtask	Metric	Ours	1st Place	Rank
A (Retrieval)	nDCG@5	0.5445	0.5776	4/38
B (Generation)	Composite	0.5638	0.7827	22/26
	RB_{agg}	0.4300		
	RL_F	0.6936		
	RB_{llm}	0.6434		
C (Full RAG)	Composite	0.5397	0.5861	13/29
	RB_{agg}	0.3987		
	RL_F	0.7052		
	RB_{llm}	0.6124		

Table 4: Official test set results.

Full RAG analysis. The Subtask C results show a different picture from the automatic generation-only evaluation. Although our system ranked 13th out of 29 by the official harmonic mean, the human evaluation on the IDK-conditioned subset ranked IIMAS-RAG second overall, with an overall score of 3.33 ± 0.50. In the human evaluation, our system achieved the highest FANC score (3.80 ± 0.16) among all participants, surpassing even the task's Reference Gold baseline (3.65 ± 0.23). FANC is defined as the harmonic mean of faithfulness, appropriateness, naturalness, and completeness, providing a unified measure of response quality. Our results indicate that while automatic metrics (RB_{alg} and RB_{llm}) penalized our system, human evaluators rated our responses as more useful and faithful than the gold-standard references. Consequently, we interpret the Subtask C performance as a fundamental trade-off: while robust retrieval and conservative answerability conditioning enhanced factual reliability, the system's performance on automatic metrics was constrained by stylistic mismatches and rigid fallback behaviors.

6 Conclusions

We presented a multi-stage system for MTRAGEval that combines hybrid sparse–dense retrieval, LLM-based query rewriting, and answerability-conditioned generation. Our best results were in Subtask A, where we ranked 4th out of 38 teams (**nDCG@5** = 0.5445). These results show that hybrid retrieval works well when conversational context is first resolved through query rewriting.

Our ablation study gives two main findings. First, query rewriting is the most important component for retrieval performance. Second, domain-specific prompt tuning (V2 and V3) provides marginal per-domain gains on specialized corpora such as Cloud and FiQA, though the differences between prompt versions are not statistically significant overall ($p = 0.642$).

The gap between retrieval and generation results (22$^{\text{nd}}$ in Subtask B and 13$^{\text{th}}$ in Subtask C) shows that good retrieval alone is not enough for strong end-to-end RAG performance. Although our system produced grounded and generally faithful answers, the lower generation rankings suggest that it did not match the reference answers closely enough in style and content. Future work will focus on improving the connection between retrieval and generation through better prompting strategies, lightweight query rewriting models, multilingual retrievers, and reranking methods better calibrated to sparse–dense fusion outputs.

Limitations and Ethical Considerations

Our system relies on commercial APIs (Voyage AI, Cohere, and OpenAI GPT-4.1) for embedding, rewriting, answerability classification, and generation. This introduces cost and availability dependencies that may limit reproducibility in resource-constrained settings; future work should explore local alternatives. Since the system was developed on English-language, its performance in multilingual or other language contexts remains unverified. The system should not be treated as a substitute for expert judgment in high-stakes domains such as finance, law, health, or public policy. Sensitive user inputs should not be sent to third-party services without appropriate consent and anonymization.

Acknowledgments

We thank the MTRAGEval shared task organizers—Yannis Katsis, Sara Rosenthal, and the entire team—

for designing and coordinating this challenging benchmark. This work was supported by compute credits from a Cohere Labs Research Grant. These grants are designed to support academic partners conducting research with the goal of releasing scientific artifacts and data-for-good resources. The authors also thank Adrian Durán Chavesti, Ricardo Villareal, and Rita Rodriguez of the Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS) for their support with the computational resources (access, configuration, and administration) used to run the experiments.

Author contributions. Vania Raya-Rios, Leon Hecht and Pedro Vázquez designed and implemented the retrieval pipeline (Subtask A). Diego Hernández, Jesus Vázquez-Osorio, and Erick Fabián-Sandoval designed and implemented the generation and answerability modules (Subtasks B and C). Helena Gómez-Adorno supervised the project.

References

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Cohere. 2025. Command a: An enterprise-ready large language model. *arXiv preprint arXiv:2504.00698*.

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009a. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 758–759, New York, NY, USA. Association for Computing Machinery.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009b. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*.

Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. TREC CAsT 2019: The Conversational Assistance Track Overview. *arXiv e-prints*, arXiv:2003.13624.

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. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2288–2292.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

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. *Transactions of the Association for Computational Linguistics*, 13:784–808.

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. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

OpenAI. 2025. Introducing gpt-4.1 in the api. Accessed: 2026-03-02.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. 3(4):333–389.

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.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the Neural Information Retrieval Series at NeurIPS 2021*.

Voyage AI. 2025. voyage-3-large: The new state-of-the-art general-purpose embedding model. Accessed: 2026-02-24.

A Detailed Experimental Configuration

This appendix details the model hyperparameters, software environment configuration, and hardware specifications used to ensure the reproducibility of the reported experiments.

A.1 Hardware Environment

All local experiments (indexing, sparse/dense retrieval with open models, and reranking) were executed on a high-performance workstation with the following characteristics:

- **GPU:** 2× NVIDIA GeForce RTX 4090 (24 GB VRAM each, Ada Lovelace architecture).
- **RAM:** 128 GB DDR5.
- **Storage:** NVMe SSD Gen4 (for fast retrieval index reading).
- **Execution Environment:** Isolated Docker container to ensure library consistency.

A.2 Software Environment

The main library versions used were pinned in the `requirements.txt` file to ensure numerical consistency:

- **Language:** Python 3.11.
- **Deep Learning Framework:** PyTorch 2.9.1 with CUDA 12.1 support.
- **Hugging Face Transformers:** 4.47.1.
- **Dense Retrieval:** FAISS-cpu 1.13.2 (exact vector search).
- **Evaluation:** pytrec_eval 0.5.

To ensure deterministic results, the global seed was set (`SEED=42`) across all libraries (Python, NumPy, PyTorch), and deterministic cuBLAS algorithms were enabled via the environment variable `CUBLAS_WORKSPACE_CONFIG=:4096:8`.

A.3 Model Hyperparameters

The following specifications detail the hyperparameters employed for each component of the pipeline:

Table 5: Hyperparameters for retrieval and rewriting models.

Component	Parameter	Value / Configuration
BM25	k_1	1.2
	b	0.75
	Tokenizer	Lucene Standard Analyzer
SPLADE	Model	`splade-cocondenser-ensembledistil`
	Aggregation	Max-pooling
	Top-k	Full sparse activation
Voyage AI	Model	`voyage-3-large`
	Dimension	1024
	Truncation	True
RRF	k	60
	Weight	Uniform
Reranker (evaluated only)	Model	`bge-reranker-v2-m3`
	Precision	FP16
Rewriting	Model	`command-r-08-2024`
	Temp.	0.3
	Top-p	0.9

B Statistical Validation Details

This appendix complements the findings in Section 5.1 by presenting detailed results of the statistical significance tests.

B.1 Test Protocol

To validate the proposed hypotheses, the Wilcoxon signed-rank test was employed for paired samples. This non-parametric test was selected because the nDCG@10 metric does not follow a normal distribution.

To control the Family-Wise Error Rate (FWER) derived from multiple comparisons, the Holm-Bonferroni sequential correction was applied. The significance level was set at $\alpha = 0.05$. A difference is considered statistically significant if the adjusted p-value is less than α.

B.2 Paired Comparison Results

Table 6 summarizes the key comparisons performed to evaluate the contribution of system components.

B.3 Analysis of Results

- **Hybrid Retrieval:** The results confirm with high statistical confidence ($p < 0.001$) that the fusion of SPLADE and Voyage (Hybrid

Table 6: Wilcoxon signed-rank test results on $nDCG@10$. * denotes $p < 0.05$ after Holm–Bonferroni correction.

ID	Comparison	Diff.	p-val	Sig.
Hybrid Component Contribution				
H1	$\text{Hybrid}_{\text{S+V}}$ vs. SPLADE	+0.124	< .001	*
H2	$\text{Hybrid}_{\text{S+V}}$ vs. Voyage	+0.045	.002	*
Rewriting Strategies				
H3	Rewriting vs. Last Turn	+0.189	< .001	*
H4	Prompt V3 vs. V2	-0.002	.642	NS
Fusion & Optimization				
H5	RRF ($k = 60$) vs. Linear	+0.043	.015	*

S+V) outperforms the best individual model (SPLADE), validating the complementarity of lexical and semantic representations.

- **Impact of Rewriting:** Query rewriting proves to be the most critical component of the pipeline, showing the largest mean difference ($+0.189$) and robust statistical significance compared to using the last conversational turn.
- **Prompt Variants:** When comparing versions V2 and V3 of the rewriting prompts, no statistical evidence of significant difference was found ($p = 0.642$). This suggests that once a baseline quality level in rewriting is reached, minor prompt variations have a marginal impact on final retrieval effectiveness.

C Retrieval Prompts

C.1 Query Rewriting Prompt

The following prompt was used to rewrite user queries into standalone retrieval queries. Prompt V1 (generic version) is shown below. **V2** augments V1 with two per-domain rules appended after rule 5: (i) a length cap (30 tokens for Cloud, 25 for FiQA, unrestricted for ClapNQ and Govt) and (ii) a vocabulary constraint (e.g., "Preserve technical terms, product names, and cloud service identifiers" for Cloud; "Preserve financial terminology: ticker symbols, fund names, ratios, and regulatory terms" for FiQA; "Preserve all named entities: agency names, program titles, legislation names, and acronyms; maintain formal register" for Govt). **V3** replaces the rule list with a reformulated frame that names the target corpus type (e.g., "financial question-answering corpus"), retains the domain vocabulary rule from V2, and prepends three few-shot input–output examples selected from the ground-truth rewrite file of each domain's training split.

```
You rewrite the FINAL user message into a
single, standalone query for information
retrieval.

Use the information of the full conversation
to make the query self-contained, unambiguous,
and faithful to the user's intent.
If the query already includes all necessary
information and is self-contained, keep it
as is. Only rewrite it if there is missing
information or ambiguity that can be resolved
from the conversation history.

Rules:
- Output ONLY the rewritten query text.
- Try to use the same number of tokens as in
  the original query (max. add up to 5 tokens).
- Do NOT answer the question.
- Resolve pronouns.
- Remove conversational filler.
- Don't change the language style of the query.

The output must be a single sentence or compact
phrase suitable for a search or embedding model.
```

C.2 HyDE Prompt

```
Given the following question, write a short
paragraph (2-3 sentences) that would be a
perfect passage answering this question.
Write as if you are writing a Wikipedia article.
Question: {query}
Passage:
```

C.3 Multi-Query Prompt

```
Given the following question, generate 3
different versions of the same question that
could be used to search for relevant information.
Each version should capture the same intent but
use different wording.
Original question: {query}
Return only the 3 questions, one per line,
without numbering.
```

C.4 Generation Prompts

The following prompt is a summarized version of the original answerability classification prompt used to determine whether a new question can be answered based solely on the provided context and conversation history. The full, uncurated prompts for the answerability module are available at https://github.com/PLN-disca-iimas/mtrag_semeval2026.

```
"You are an evaluation model. Given:
1. Context
2. Conversation History (if any)
3. New Question

Determine whether the new question can be answered
     using only the provided information.

First, briefly consider why the question could be
     ANSWERABLE, PARTIAL, or UNANSWERABLE. Then
     provide your final decision.

Output strictly in the following format:

Label: [ANSWERABLE / UNANSWERABLE / PARTIAL /
     CONVERSATIONAL]

Rules:
- Do NOT answer the question itself.
```

```
- Base your decision only on the given inputs.
- Keep the explanation short and factual."
```

The following prompt is a summarized version of the original response-generation prompt used in the shared task. The full, uncurated prompts for the answerability module and generation module are available at https://github.com/PLN-disca-iimas/mtrag_semeval2026.

```
"You are an AI assistant in a shared task. Your
     objective is to generate answers that match the
      style, structure, and level of detail of the
     datasets target answers.

Before answering, read the provided answerability
     label and strictly follow it.

Rules:

1. Style:
   - Neutral, factual, concise.
   - No conversational fillers.
   - No opinions or extra commentary.

2. Detail:
   - Match the datasets level of specificity.
   - Include small clarifications only if
       consistently present.
   - Do not add extra details.

3. Context:
   - Use only the provided context.
   - Use conversation history only for
       disambiguation.

4. Answerability:
   - If UNANSWERABLE     Respond exactly:
     "I'm sorry, but I don't have the answer to your
          question."
   - If PARTIAL     Answer only with available
       information.
   - If ANSWERABLE     Provide the complete answer
       using only supported facts.

5. No Hallucinations:
   - Do not infer or fabricate missing information.

Inputs:
Answerability: {answerability}
Question: {current_question}
Conversation History: {conversation_history}
Contexts: {contexts_text}"
```

C.5 Examples of conversations

The following example illustrates a case where the question is labeled as UNANSWERABLE, and the generated response must exactly match the predefined fallback answer.

```
QUESTION:
where do the arizona cardinals play this week

GENERATED RESPONSE:
I'm sorry, but I don't have the answer to your
     question.

TARGET RESPONSE:
I'm sorry, but I don't have the answer to your
     question.
```

The following example illustrates a case where the question is labeled as PARTIAL, meaning that the available context supports only a subset of the requested information.

```
QUESTION:
How many teams qualify for the NFL playoffs and how
     many are from each conference?

GENERATED RESPONSE:
Fourteen teams qualify for the NFL playoffs, seven
     from each conference.

TARGET RESPONSE:
Fourteen teams qualify for the NFL playoffs.
```

The following example illustrates a representative generation error where the system produced a complete answer with an additional clarification, while the target response contained only the factual core. This type of error penalized our reference-based automatic scores even though the response was factually correct, grounded in the retrieved passages, and not hallucinated.

```
QUESTION:
What fees apply to my IRA account?

GENERATED RESPONSE:
The annual maintenance fee is $25. Additionally, a
     $10 wire transfer fee applies to outgoing
     transfers.

TARGET RESPONSE:
The annual maintenance fee is $25.
```