GigitAI at SemEval-2026 Task 8: Hybrid Sparse-Dense Retrieval and Zero-Shot Generation for Multi-Turn Conversational RAG

Saran Krishnasamy
GigitAI
San Francisco, CA
saran@gigit.ai

Inez Wihardjo
GigitAI
San Francisco, CA
inez@gigit.ai

Abstract

We describe our system for SemEval-2026 Task 8 (MTRAGEval) on multi-turn conversational RAG. Our approach combines hybrid retrieval (fusing SPLADE-v3 learned sparse representations with dense embeddings via Reciprocal Rank Fusion) with a fine-tuned cross-encoder reranker and zero-shot LLM generation using Claude Opus 4.5. We systematically evaluate 56 retrieval configurations across 4 domains, and 5 generation strategies across 5 LLMs. Our findings show that: (1) SPLADE-v3 with dataset rewrites substantially outperforms BM25 across all configurations, (2) simple zero-shot prompting matches sophisticated strategies like Self-RAG and CRAG, and (3) performance varies significantly by answerability class. On the test set, we achieve **rank 5/29** on Task C (end-to-end RAG, H=0.5564), **rank 7/26** on Task B (generation, H=0.7495), and rank 13/38 on Task A (retrieval, nDCG@5=0.4782). Our analysis reveals strong performance on answerable queries (H=0.685) but degradation on underspecified queries (H=0.254).[1]

1 Introduction

Retrieval-Augmented Generation (RAG) has emerged as the dominant paradigm for grounding large language model responses in external knowledge (Gao et al., 2024). However, most RAG benchmarks focus on single-turn queries, neglecting the unique challenges of multi-turn conversational settings where queries contain unresolved coreferences (e.g., "What about *its* pricing?"), exhibit topic drift across turns, and require maintaining coherent conversational context.

SemEval-2026 Task 8 (MTRAGEval) addresses this gap by providing a comprehensive benchmark for multi-turn conversational RAG (Rosenthal et al., 2026b). The task spans four diverse domains (Katsis et al., 2025): ClapNQ (Wikipedia), IBM Cloud documentation, FiQA (financial Q&A), and government documents, each presenting unique retrieval and generation challenges.

We present a modular RAG system that achieves strong performance through careful component selection and systematic evaluation:

1. **Hybrid retrieval**: Our submitted pipeline fuses SPLADE-v3 (Lassance et al., 2024) learned sparse representations with OpenAI dense embeddings using Reciprocal Rank Fusion, followed by a fine-tuned BGE cross-encoder reranker (Xiao et al., 2024). We evaluate 56 retrieval configurations on development data using Cohere rerank-v3.5 as a controlled off-the-shelf reranker (Section 5.1).

2. **Simple generation**: We demonstrate that zero-shot prompting with Claude Opus 4.5 matches or exceeds sophisticated strategies like Self-RAG and CRAG.

3. **Comprehensive analysis**: Beyond retrieval, we evaluate 11 generation configurations spanning 5 LLMs and 5 strategies (Table 6), providing insights into what works for conversational RAG.

Our system ranked **5th out of 29 teams** on Task C (end-to-end RAG), demonstrating that carefully tuned simple components can achieve competitive results without complex architectural innovations.

2 Background

2.1 Task Description

MTRAGEval (Rosenthal et al., 2026a) evaluates conversational RAG systems through three subtasks:

Task A (Retrieval): Given a multi-turn conversation, retrieve the 10 most relevant passages

[1]Code available at https://github.com/sarankrish/mtrageval2026

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 3103–3111

from a domain-specific corpus. Evaluated using nDCG@5 (Normalized Discounted Cumulative Gain at rank 5).

Task B (Generation): Given a conversation and pre-retrieved passages (oracle contexts), generate a response grounded in the passages. Evaluated using the harmonic mean of three metrics: RougeL F1 (RL_F), LLM-judge relevance score (RB_llm), and an aggregate score combining recall, RougeL, and extractiveness (RB_agg).

Task C (End-to-End RAG): Perform both retrieval and generation using system-retrieved passages. Uses the same generation metrics as Task B.

2.2 Dataset

The MTRAG-UN dataset (Rosenthal et al., 2026a) extends the MTRAG benchmark (Katsis et al., 2025) with unanswerable and underspecified queries. It contains multi-turn conversations (2–10 turns) with evolving information needs. Each example includes conversation history, gold passages, reference answers, and pre-computed query rewrites. Table 1 summarizes the four domains.

Corpus	Domain	Passages (P)
ClapNQ	Wikipedia	183,408
Cloud	Technical documentation	72,442
FiQA	Finance	61,022
Govt	Government	49,607

Table 1: MTRAG corpus statistics by domain (Katsis et al., 2025).

2.3 Related Work

Conversational question answering has been studied extensively through benchmarks like CoQA (Reddy et al., 2019), which introduced coreference and pragmatic reasoning challenges, and QReCC (Anantha et al., 2021), which combined question rewriting with open-domain retrieval over 54M passages. The TREC Conversational Assistance Track (CAsT) (Dalton et al., 2020) established evaluation protocols for conversational search, finding that manual query rewrites improve retrieval by 35% over automatic systems. Dense retrieval methods (Karpukhin et al., 2020) and conversational adaptations (Yu et al., 2021) have shown strong performance, though hybrid approaches combining sparse and dense signals via Reciprocal Rank Fusion (Cormack et al., 2009) often outperform either alone. SPLADE (Formal et al., 2022) bridges this gap by learning sparse representations with semantic term expansion.

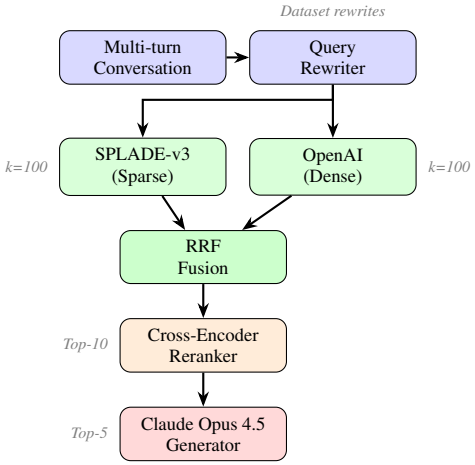

Figure 1: System architecture: queries are rewritten using dataset rewrites, then retrieved via SPLADE-v3 and dense embeddings fused with RRF. Cross-encoder reranking selects top-10, with top-5 passed to the LLM generator.

Retrieval-Augmented Generation (Lewis et al., 2020) grounds LLM responses in retrieved evidence, with recent surveys (Gao et al., 2024) documenting rapid progress. Advanced strategies like Self-RAG (Asai et al., 2024) add reflection mechanisms to critique and revise responses, while Corrective RAG (Yan et al., 2024) filters low-relevance passages before generation. Cross-encoder reranking (Reimers and Gurevych, 2019) improves precision by jointly scoring query-passage pairs. Our work systematically compares these strategies, finding that simple zero-shot prompting with capable LLMs matches more sophisticated approaches.

3 System Overview

Figure 1 illustrates our system architecture with four main components.

3.1 Query Processing

Multi-turn conversations contain pronouns and contextual references requiring resolution. Consider this example:

> *Turn 1: "What is IBM Cloud Functions?"*
> *Turn 2: "How much does it cost?"*
> *→ Rewrite: "How much does IBM Cloud Functions cost?"*

We compared three strategies: (1) **last-turn only**, (2) **LLM rewriting** using GPT-4.1 or Claude Opus 4.5 (prompt in Appendix C), and (3) **dataset**

Strategy	nDCG@5	Δ
Last turn only	0.177	–
Dataset rewrite	0.212	+19.8%
LLM (GPT-4.1)	0.221	+24.9%
LLM (Claude Opus 4.5)	0.225	+27.1%

Table 2: Query processing impact on BM25 retrieval (dev set).

rewrites pre-computed by the MTRAG organizers. Table 2 shows dataset rewrites improve BM25 nDCG@5 by 19.8% at zero cost, capturing 73% of LLM rewriting benefit.

3.2 Retrieval

We evaluated four approaches: **BM25** using Pyserini (Lin et al., 2021) (k_1=0.9, b=0.4) for lexical matching; **Dense** using OpenAI `text-embedding-3-large` (OpenAI, 2024) (3072-dim.) for semantic similarity; **SPLADE-v3** (Lassance et al., 2024), a learned sparse encoder that combines lexical precision with semantic term expansion (e.g., "cost" ↔ "pricing"); and **Hybrid Fusion** via Reciprocal Rank Fusion (RRF) (Cormack et al., 2009):

$$\text{RRF}(d) = \sum_{r \in R} \frac{w_r}{k + \text{rank}_r(d)} \quad (1)$$

where R is the set of retrievers, w_r is the weight for retriever r, and k=60. We retrieve top-100 candidates from each system before fusion in our development ablations; for official test submissions we increase this to top-300 per retriever (see Section 5.6 and Appendix A).

3.3 Reranking

We evaluate two cross-encoder rerankers. **Cohere rerank-v3.5** (Cohere, 2024) is a strong off-the-shelf API model used throughout our development-set ablations (Tables 4–9), improving nDCG@5 across all retriever configurations. For our official Task A and Task C test submissions, we use a **fine-tuned BGE-base cross-encoder** (`BAAI/bge-reranker-base`; Xiao et al. 2024), trained on MTRAG conversations using gold passages as positives and hard-negative passage pairs. Both rerankers reduce the fused candidate list to a top-10 ranking. We use Cohere in ablations to isolate retriever choice from reranker training effects.

3.4 Generation

Given retrieved passages, we generate responses using large language models. We evaluated five generation strategies:

Simple: Zero-shot prompting with retrieved passages as context. The prompt instructs the model to answer directly using only passage information (full prompt in Appendix C).

Citation-grounded: Require explicit passage citations (e.g., "[1]") in responses to improve traceability and reduce hallucination.

Self-RAG-inspired reflection (Asai et al., 2024): Following the spirit of Self-RAG, we generate an initial response then use a separate LLM call to reflect on quality (relevance, support, completeness). Unlike the original Self-RAG which requires a specially fine-tuned model, our implementation uses standard prompting.

CRAG-inspired filtering (Yan et al., 2024): Following the relevance grading concept from CRAG, we score passage relevance before generation and filter low-relevance passages. We omit CRAG's web search fallback as it is outside the task constraints.

Answerability Detection: Before generation, detect whether the question is answerable from the passages. If classified as unanswerable, return a standard abstention response rather than hallucinating.

4 Experimental Setup

We used the official MTRAG development set (777 retrieval queries, 842 generation tasks) for model selection. The test set contained 507 examples with a surprise "underspecified" answerability class. For generation, we evaluated GPT-4o-mini and GPT-5.2 (Singh et al., 2026), Claude Sonnet 4 (Anthropic, 2025b) and Claude Opus 4.5 (Anthropic, 2025a), and Llama-3.3-70B (Meta AI, 2024) (temperature 0.3, max 512 tokens, top-5 passages). Full model IDs are listed in Appendix A. SPLADE indices were built in BEIR format (Thakur et al., 2021) using `sentence-transformers`; dense embeddings used OpenAI `text-embedding-3-large` with Pinecone[2]. The fine-tuned BGE reranker was trained on MTRAG conversational training data; the Cohere reranker is used as-is. Full hyperparameters in Appendix A.

5 Results

Table 3 summarizes our official test results against organizer-provided baselines.

[2] https://www.pinecone.io

Task	Ours	Baseline	Top	Rank
A (nDCG@5)	0.478	0.480	0.578	13/38
B (H)	0.750	0.639	0.783	7/26
C (H)	0.556	0.537	0.586	5/29

Table 3: Official test results. Baselines: Task A uses ELSER + GPT-OSS-20b rewrite; Tasks B/C use gpt-oss-120b and qwen-30b-a3b-thinking respectively.

Retriever	Configuration	nDCG@5
BM25	last turn	0.177
SPLADE-v3	last turn	0.226
Dense	last turn	0.366
BM25	+rewrite+rerank	0.340
Dense	+rewrite+rerank	0.427
Hybrid BM25+Dense	+rewrite+rerank	0.417
Hybrid SPLADE+Dense	+rewrite+rerank	0.397
SPLADE-v3	+rewrite+rerank	**0.437**

Table 4: Task A retrieval results on dev set, all using Cohere rerank-v3.5 as a controlled reranker for cross-retriever comparison. Our test submission uses a fine-tuned BGE reranker (Section 3.3). Full results in Appendix E.

5.1 Task A: Retrieval

Table 4 presents retrieval results. Query rewriting improves all retrievers, with sparse methods benefiting most: SPLADE gains 88% from rewriting alone (0.226→0.425) and a further 3% from reranking (→0.437). Both hybrid configurations underperform SPLADE alone with Cohere reranking: BM25+Dense (0.417) and SPLADE+Dense (0.397) trail SPLADE-v3 (0.437). The SPLADE+Dense result is particularly notable, suggesting that SPLADE's learned sparse representations already capture the semantic signals dense embeddings would add.

Per-Domain Analysis Performance varies substantially across domains (Table 5). ClapNQ achieves the highest scores (0.513 nDCG@5), benefiting from Wikipedia's encyclopedic structure. FiQA proves most challenging (0.384), as specialized financial terminology creates vocabulary mismatch even for learned sparse methods.

Config	ClapNQ	Cloud	FiQA	Govt
BM25	0.199	0.191	0.080	0.229
+rewrite	0.227	0.221	0.120	0.269
SPLADE+rw	0.496	0.376	0.389	0.431
+rerank	**0.513**	**0.390**	0.384	**0.452**

Table 5: Per-domain nDCG@5 (dev set).

Official Test Results On the test set, our system achieved 0.4782 nDCG@5 (rank 13/38), narrowly trailing the best baseline (ELSER[3] with GPT-OSS-20b rewriting, 0.4795) and 17% below the top system (0.5776).

Submitted vs. Ablated Configuration Our official Task A submission uses hybrid SPLADE+Dense fusion with top-300 initial retrieval and the fine-tuned BGE reranker. The dev-set ablations in Table 4 use top-100 retrieval with Cohere reranking to enable controlled comparison across retrievers; absolute numbers are therefore not directly comparable to the submitted configuration. Notably, the dev ablations suggest SPLADE alone with Cohere reranking (0.437) slightly outperforms the hybrid (0.397), pointing to a possible simplification we did not exploit at submission time.

5.2 Task B: Generation

Table 6 compares generation strategies and models. Model capability strongly correlates with performance: Claude Opus 4.5 and Llama-3.3-70B achieve the highest scores, while GPT-4o-mini lags behind. Surprisingly, generation strategy has minimal impact: Self-RAG's reflection and CRAG's filtering provide no meaningful improvement over simple zero-shot prompting, suggesting that with capable models, multi-step generation pipelines yield diminishing returns.

Answerability detection significantly hurts overall scores, dropping GPT-5.2 from 0.607 to 0.399. While appropriate abstention is the correct behavior for unanswerable queries, the evaluation metrics reward lexical overlap with reference answers, penalizing short abstention responses. This creates a tension between faithful behavior and metric optimization that we discuss further in Section 5.4.

Official Test Results Our generation pipeline achieved H=0.7495, ranking 7th of 26 (RL_F=0.8906, RB_llm=0.8006, RB_agg=0.6133). This substantially outperforms the best baseline (0.6390) and approaches the top system (0.7827). The dev-to-test improvement (H=0.629→0.7495) reflects evaluation differences we cannot fully attribute from released data.

[3]Elastic Learned Sparse EncodeR, Elastic's proprietary sparse retrieval model.

Model	Strategy	H	RL_F	RB_llm	RB_a
Model comparison (Simple strategy)					
GPT-4o-mini	Simple	.579	.811	.688	.424
GPT-5.2	Simple	.607	.847	.720	.443
Claude Son. 4	Simple	.608	.830	.719	.451
Llama-3.3-70B	Simple	.628	.852	.696	.462
Claude Opus 4.5	Simple	**.629**	.844	.750	.465
Strategy comparison (GPT-5.2)					
GPT-5.2	Simple	.607	.847	.720	.443
GPT-5.2	Citation	.610	.849	.722	.444
GPT-5.2	Self-RAG	.608	.854	.715	.441
GPT-5.2	CRAG	.596	.838	.690	.438
With answerability detection					
GPT-5.2	+Answer.	.399	.548	.555	.295
Claude Opus 4.5	+Answer.	.464	.619	.588	.346

Table 6: Task B generation results (dev set). H=harmonic mean of RL_F, RB_llm, and RB_agg, all IDK-conditioned per the official evaluation protocol. Model choice matters more than strategy.

Class	n	H	RB_a	RB_llm	RL_F
Task B (Oracle passages)					
ANSWERABLE	285	**.685**	.554	.805	.940
PARTIAL	47	.575	.461	.734	.829
UNANSWERABLE	97	.000	.000	.825	.174
UNDERSPECIFIED	78	.254	.191	.653	.578
Task C (System retrieval)					
ANSWERABLE	285	**.515**	.416	.633	.776
PARTIAL	144	.165	.318	.645	.380
UNDERSPECIFIED	78	.182	.172	.426	.448

Table 7: Test set performance by answerability class. UNANSWERABLE queries exhibit catastrophic failure. The UNANSWERABLE label is defined relative to oracle passages and collapses into PARTIAL when system retrieval is used (47+97=144 in Task C).

5.3 Task C: End-to-End RAG

Our system achieved H=0.5564, ranking 5th of 29 teams, our strongest relative result, exceeding the best baseline (0.5366) and approaching the top system (0.5861). The gap between Task B (0.7495) and Task C (0.5564) quantifies the retrieval bottleneck: generation quality degrades by 25% when using system-retrieved rather than oracle passages.

5.4 Test Set Analysis by Answerability

We analyzed performance by answerability class, revealing the most important finding of our work. Table 7 shows dramatic variation across question types.

On answerable queries, our system achieves H=0.685 on Task B, competitive with top submissions. However, performance collapses on unanswerable queries (H=0.000): RB_llm remains high (0.825) because responses are fluent, but RB_agg is zero because the model hallucinates rather than abstaining. The test set also introduced an "underspecified" class not in our development split (H=0.254/0.182), where queries require clarification our system cannot provide.

5.5 Error Analysis

Retrieval Failures We manually analyzed 100 random retrieval failures (gold passage not in top-10): **lexical mismatch** (42%), where queries use different terminology than passages (e.g., "pricing" vs "cost structure"); **multi-hop reasoning** (28%), requiring synthesis of multiple passages; **context loss** (18%), where important context from earlier turns is not captured in the rewrite; and **domain jargon** (12%), with specialized terms not in SPLADE's vocabulary, especially in FiQA.

Generation Failures On UNANSWERABLE queries, the model generates plausible but unsupported answers (e.g., hallucinating "founded in 2015") instead of abstaining.

5.6 Ablation Studies

Retrieval Depth We evaluated initial retrieval depth before reranking (Table 8). Top-100 provides optimal recall-latency tradeoff; top-200 shows diminishing returns.

Initial k	nDCG@5	R@10
Top-50	0.421	0.512
Top-100	**0.437**	**0.598**
Top-200	0.435	0.601

Table 8: Retrieval depth ablation.

Passage Count Table 9 shows top-5 passages balance context with signal-to-noise ratio. More passages introduce noise.

Passages	H	RL_F	RB_a
Top-3	0.612	0.831	0.449
Top-5	**0.629**	0.844	0.465
Top-10	0.611	0.842	0.441

Table 9: Passage count ablation (Claude Opus 4.5).

RRF Fusion Weights For the hybrid SPLADE+Dense configuration, equal weighting (0.5/0.5) slightly outperforms 0.6/0.4 splits in either direction (approximately -0.5% nDCG@5),

though SPLADE alone still outperforms the hybrid overall (Table 4).

6 Conclusion

We presented a modular RAG system for MTRAGEval combining hybrid SPLADE+Dense retrieval with fine-tuned cross-encoder reranking and zero-shot generation, achieving rank 5/29 on Task C, rank 7/26 on Task B, and rank 13/38 on Task A. Our experiments reveal that SPLADE-v3 with query rewriting outperforms dense retrieval while offering interpretability, and simple zero-shot prompting performs comparably to Self-RAG and CRAG-inspired strategies. The 25% gap between Tasks B and C confirms retrieval as the primary bottleneck, while answerability detection remains a critical challenge. In future work, we plan to fine-tune sparse retrievers on conversational data, develop robust answerability classifiers, and explore clarification-question generation for underspecified queries.

References

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-domain question answering goes conversational via question rewriting. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 520–534, Online. Association for Computational Linguistics.

Anthropic. 2025a. Claude opus 4.5 system card.

Anthropic. 2025b. System card: Claude Opus 4 & Claude Sonnet 4.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Cohere. 2024. Rerank 3.5: Precise AI search.

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. 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.

Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. Trec cast 2019: The conversational assistance track overview.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From distillation to hard negative sampling: Making sparse neural ir models more effective. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2353–2359, New York, NY, USA. Association for Computing Machinery.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

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. mt RAG: A multi-turn conversational benchmark for evaluating retrieval-augmented generation systems. *Transactions of the Association for Computational Linguistics*, 13:784–808.

Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. 2024. Splade-v3: New baselines for splade. *Preprint*, arXiv:2403.06789.

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 *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2356–2362, New York, NY, USA. Association for Computing Machinery.

Meta AI. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

OpenAI. 2024. New embedding models and API updates. `https://openai.com/index/new-embedding-models-and-api-updates`. Blog post, January 25, 2024.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

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.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, and 467 others. 2026. Openai gpt-5 system card. *Preprint*, arXiv:2601.03267.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. page 641–649.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*.

Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-shot conversational dense retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 829–838, New York, NY, USA. Association for Computing Machinery.

A Hyperparameters

Component	Setting
Retrieval	
BM25 k_1	0.9
BM25 b	0.4
RRF smoothing k	60
Initial retrieval depth	top-100 (abl.), top-300 (subm.)
Reranker output	top-10
Generation	
Temperature	0.3
Max tokens	512
Passages for context	5
Max history turns	5
Models & API IDs	
Dense embeddings	text-embedding-3-large
SPLADE	naver/splade-v3
Reranker (ablation)	rerank-v3.5 (Cohere)
Reranker (submission)	BAAI/bge-reranker-base
GPT-4o-mini	gpt-4o-mini
GPT-5.2	gpt-5.2
Claude Sonnet 4	claude-sonnet-4-20250514
Claude Opus 4.5	claude-opus-4-5-20251101
Llama-3.3-70B	Llama-3.3-70B-Instruct

Table 10: Hyperparameters and model IDs. Submission reranker is fine-tuned on MTRAG data.

B Query Rewriting Cost Analysis

Table 11 shows the cost-performance tradeoff for LLM-based query rewriting.

Rewriting Method	Cost/Query	nDCG@5
None (last turn only)	$0	0.177
Dataset rewrites	$0	0.212
GPT-4.1	$0.02	0.221
Claude Opus 4.5	$0.08	0.225

Table 11: Query rewriting cost analysis. Dataset rewrites provide 73% of the benefit at zero inference cost.

C Prompt Templates

Query Rewriting Prompt

```
Given the following conversation, rewrite the last
    user question into a standalone search query
    that captures the full context needed for
    retrieval. The query should be self-contained.

Conversation: {conversation}

Rewrite the last question as a standalone search
    query. Output only the query.
```

Generation Prompt (Simple)

```
You are a helpful assistant. Answer the question
     directly and concisely using ONLY information
     from the provided passages.

Passages: {passages}
Conversation History: {history}
Question: {question}

Instructions:
- Give a DIRECT, CONCISE answer (2-4 sentences
     typical)
- Use ONLY facts from the passages
- Do NOT start with "Based on the passages"
- If passages don't contain the answer, say: "I don'
     t have specific information about [topic]."

Answer:
```

Self-RAG Reflection Prompt

```
Evaluate whether the following answer is well-
     supported by the passages.

Passages: {passages}
Question: {question}
Generated Answer: {answer}

Evaluate on a scale of 1-5:
1. Relevance: Does it address question? (1-5)
2. Support: Is every claim supported? (1-5)
3. Completeness: Does it cover all info? (1-5)

Respond with ONLY a JSON object.
Example: {"relevance": 4, "support": 5, "
     completeness": 3, "needs_revision": false}
```

CRAG Relevance Grading Prompt

```
Rate the relevance of each passage to the question.

Question: {question}
{passages}

For each passage, output a relevance score from 0.0
     to 1.0.
Output JSON: {"passage_1": 0.X, "passage_2": 0.X,
     ...}
```

Answerability Detection Prompt

```
Determine if the question can be answered using the
     provided passages.

Passages: {passages}
Question: {question}

Consider:
1. Do passages contain relevant info?
2. Is there enough for a complete answer?
3. Does it require external knowledge?

Output JSON: {"answerable": true/false, "confidence
     ": 0.X, "reason": "..."}
```

D Error Examples

Representative failure cases from our Task B test results:

Example 1: UNANSWERABLE Query

History: User asks about Watson Assistant intents, then "How is it created?" (agent explains steps)

Query: "Are those the only steps?"

Gold: "If there are additional steps, I am not aware of them as the provided documents do not contain any information about them."

System: "I don't have specific information about whether those are the only steps for creating a dialog, as no passages were provided to reference."

Issue: Both acknowledge missing info, but the system incorrectly claims no passages were provided, itself a hallucination about the task state, despite receiving oracle passages in Task B (RB_agg=0, RB_llm=0.85)

Example 2: UNDERSPECIFIED Query

History: 4 turns about inventory turnover ratios for retail and car dealers

Query: "What is the secret behind the widespread acclamation of this burger in the fast food industry?"

Gold: "There are many burgers in the fast food industry, like the Big Boy hamburger. Which one are you talking about?"

System: "The Big Boy hamburger originated in February 1937 when Bob Wian improvised and created it as a joke for regular customers who asked for 'something different, something special.' He intended it to 'look ridiculous, like a leaning tower.' Demand for 'the special' soared, and it was given the 'snappy' name Big Boy. The passages don't provide specific information about the secrets behind its widespread acclamation in the fast food industry beyond this origin story."

Issue: Abrupt topic shift; model assumes Big Boy from passages instead of asking for clarification (RB_llm=0.6)

E Extended Retrieval Results

Table 12 shows selected configurations from our 56 retrieval experiments.

ID	Retriever	Query	Rerank	nDCG@5	R@5	ClapNQ	Cloud	FiQA	Govt
1	BM25	last turn	–	0.177	0.196	0.199	0.191	0.080	0.229
2	BM25	dataset rw	–	0.212	0.238	0.227	0.221	0.120	0.269
3	BM25	GPT-4.1 rw	–	0.221	0.249	0.260	0.193	0.124	0.296
4	BM25	dataset rw	Cohere	0.340	0.356	0.416	0.323	0.222	0.383
5	Dense	last turn	–	0.366	0.387	0.450	0.337	0.290	0.374
6	Dense	dataset rw	–	0.406	0.439	0.475	0.354	0.339	0.443
7	Dense	dataset rw	Cohere	0.427	0.457	0.522	0.371	0.369	0.434
8	SPLADE	last turn	–	0.226	0.252	0.273	0.175	0.172	0.275
9	SPLADE	dataset rw	–	0.425	0.464	0.496	0.376	0.389	0.431
10	SPLADE	dataset rw	Cohere	**0.437**	**0.467**	**0.513**	**0.390**	0.384	**0.452**
11	BM25+Dense	GPT-4.1 rw	Cohere	0.417	0.460	0.538	0.346	0.321	0.444
12	SPLADE+Dense	LLM rw	Cohere	0.397	0.423	0.476	0.397	0.319	0.387

Table 12: Selected retrieval configurations (12 of 56 total). Full results available in code repository.