NLP-CEIA-UFG at SemEval-2026 Task 8: Iterative Retrieval with Notes-Guided Query Refinement for Multi-Turn RAG

Guilherme C. Dutra[1], **André F. S. Caraíba**[1], **Nádia F. F. da Silva**[1]
Paulo Victor dos Santos[1], Deborah S. A. Fernandes[1], Sávio S. T. de Oliveira[1]
[1] Instituto de Informática — Universidade Federal de Goiás (UFG)
{guilherme.dutra, andre.caraiba}@discente.ufg.br

Abstract

We describe NLP-CEIA-UFG, our system for SemEval-2026 Task 8, which evaluates multi-turn retrieval-augmented generation (RAG) over heterogeneous document corpora. Our pipeline centers on a three-iteration dynamic retrieval loop in which two `gpt-oss-120b`-powered modules—an Iterative Query Generator and a Notes Builder—interact at each step to diversify queries and accumulate structured notes on information gaps. After the loop, an Answerability Classifier routes the query to one of three generation paths (Complete Answer, Partial Answer, or Clarification Request). Hybrid BM25 and dense retrieval is fused via Reciprocal Rank Fusion and refined by the Jina listwise reranker. The retrieval pipeline is compiled under DSPy and optimized with GEPA. We achieve nDCG@5 of 0.4502 (rank 17/38, Subtask A) and HM = 0.3774 (rank 24/29, Subtask C). Post-hoc analysis identifies an over-conservative Answerability Classifier as the primary bottleneck: 75.5% of all responses were flagged as IDK by the evaluator, including 69.8% of ANSWERABLE questions, while the retrieval and generation components perform well when the classifier routes correctly. Our code is available at https://github.com/GuiiCorreia/SemEval-2026.

1 Introduction

Multi-turn conversational RAG (Lewis et al., 2020) demands that systems retrieve relevant passages from a document corpus and generate grounded responses to questions that evolve across dialogue turns. Follow-up questions frequently feature coreferences and ellipsis that render them non-retrievable in isolation; furthermore, systems must assess whether the corpus supports a response before proceeding with generation.

SemEval-2026 Task 8, MTRAGEval (Rosenthal et al., 2026b), establishes a standardized benchmark via the MTRAG-UN corpus (Rosenthal et al., 2026a), covering four domains (Wikipedia, financial, government, and cloud documentation) with answerability annotations. All corpora are in English. The task evaluates retrieval (nDCG@5, Subtask A) and end-to-end generation (harmonic mean, Subtask C).[1]

Our system tackles the retrieval challenge by replacing single-pass search with a *three-iteration dynamic retrieval loop*: an Iterative Query Generator produces diversified queries conditioned on conversation history and prior Notes, and a Notes Builder records information gaps to refine the next hop. Hybrid BM25 and dense retrieval is fused via RRF (Cormack et al., 2009) and refined by the Jina listwise reranker (Wang et al., 2025).

For generation, rather than relying on a monolithic prompt or binary abstention (IDK), we implement a 3-way routing system. An Answerability Classifier evaluates the retrieved context and routes the query to either a Complete Answer, Partial Answer, or Clarification Request generator using Gemini 2.5 Flash (Google DeepMind, 2025).

We rank 17/38 in Subtask A and 24/29 in Subtask C. Post-hoc analysis identifies the Answerability Classifier as the primary bottleneck: it routes 69.8% of ANSWERABLE questions away from the Complete Answer Generator, producing responses that the benchmark's IDK evaluator penalizes. When the classifier routes correctly, generation quality is high (RB_agg = 0.304, RL_F = 0.822).

2 Background and Related Work

Task and data. SemEval-2026 Task 8 (Rosenthal et al., 2026b) defines three subtasks. **Subtask A** requires a ranked list of up to ten passages per turn (nDCG@5). **Subtask B** evaluates generation

[1]Code repository: https://github.com/GuiiCorreia/SemEval-2026. The repository name reflects the original development branch; the code corresponds to the SemEval-2026 Task 8 submission.

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 2771–2778

given gold passages; we did not attempt Subtask B due to submission timeline constraints, although it would provide a clean upper bound on generation quality absent retrieval noise. **Subtask C** demands end-to-end retrieval and generation; the metrics are: harmonic mean (HM) of RB_agg (BertRec + BertKPrec + RougeL), RL_F (RAGAS faithfulness), and RB_llm (LLM-as-judge), with IDK-conditioning (IDK scores 1.0 for UNANSWERABLE, 0.0 for ANSWERABLE). The evaluation set is drawn from MTRAG-UN (Rosenthal et al., 2026a), extending MTRAG (Katsis et al., 2025): 507 tasks across CLAPNQ (Wikipedia; $n = 142$), Govt ($n = 157$), Cloud ($n = 131$), and FiQA ($n = 77$), with 91.7% follow-up turns. Ground-truth labels: 56.2% ANSWERABLE, 28.2% PARTIAL, 15.4% UNDERSPECIFIED, 0.2% UNANSWERABLE. Notably, conversational turns were excluded from the MTRAG evaluation (Katsis et al., 2025) and are absent from the MTRAG-UN test set by design (Rosenthal et al., 2026a).

Related work. Multi-turn retrieval necessitates resolving context-dependent queries before search (Aliannejadi et al., 2024; Kuo et al., 2025). Iterative approaches—such as IRCoT (Trivedi et al., 2023), FLARE (Jiang et al., 2023), Iter-RetGen (Shao et al., 2023), and Self-RAG (Asai et al., 2024)—demonstrate that feedback-driven retrieval consistently outperforms single-pass pipelines; our Notes Builder externalizes this signal into an optimizer-tunable module, extending Baleen (Khattab et al., 2021). Hybrid retrieval fused by RRF (Bruch et al., 2023) with listwise reranking (Wang et al., 2025; Sun et al., 2023) forms our evidence-gathering backbone. DSPy (Khattab et al., 2024), MIPROv2 (Opsahl-Ong et al., 2024), and GEPA (Agrawal et al., 2026) provide the prompt-optimization infrastructure. Answerability detection (Chen and Mueller, 2024) is critical for faithful RAG. While Katsis et al. (2025) note models struggle with unanswerable questions, our findings highlight a distinct challenge: an over-conservative answerability classifier that suppresses generation even when sufficient evidence has been retrieved.

3 System Overview

Our pipeline runs a *three-iteration retrieval loop* then classifies answerability and conditionally routes generation. Figure 1 provides an overview.

Iterative Query Generator. The DSPy `ChainOfThought` module, powered by `gpt-oss-120b` (OpenAI, 2025), receives the full conversation history $\mathcal{H}_t$, current question q_t, and Notes from the previous iteration $\mathcal{N}^{(k-1)}$ (empty at $k = 1$), producing three semantically diverse queries. Notes expose information gaps and suggest reformulations, enabling qualitatively different queries at each iteration.

Hybrid Retrieval, RRF, and Reranking. Each query independently retrieves top-10 passages via **BM25** (Robertson and Zaragoza, 2009) and **dense retrieval** with google-embedding-001 (Google DeepMind, 2025) from a Qdrant database. The lists merge via Reciprocal Rank Fusion ($k_0 = 60$; Cormack et al. 2009), yielding up to 30 candidates reranked by `jina-reranker-v3` (Wang et al., 2025)—a 0.6B listwise cross-encoder—to produce the top-10 evidence set $\mathcal{D}^{(k)}$.

Notes Builder. After each reranking step, a `gpt-oss-120b` DSPy module produces structured Notes $\mathcal{N}^{(k)}$ with three fields: (i) *key findings*, (ii) *missing information*, and (iii) *search suggestions* for the next query generation call.

Iteration count. We selected $K = 3$ iterations as a balance between retrieval diversity and API cost constraints: $K = 1$ offers no self-correction opportunity, while $K \geq 4$ risks context accumulation that may cause the LLM to drift from the original query. A systematic ablation over K was not feasible within the submission window due to API rate limits and is left for future work.

Answerability Classifier. After global reranking, a module assigns one of three labels: ANSWERABLE (A), PARTIAL (P), or UNANSWERABLE (U). To optimize compute, if the retrieval step returns exactly zero documents, the system bypasses the LLM classifier and automatically assigns UNANSWERABLE. For Subtask A, the globally reranked $\mathcal{D}_{\text{final}}$ is returned directly.

Conversational Response Routing. Based on the classification, the system routes the context to Gemini 2.5 Flash (Google DeepMind, 2025) using one of three specialized generation templates:

- **Complete Answer Generator:** Synthesizes information across documents while strictly citing details (for A).

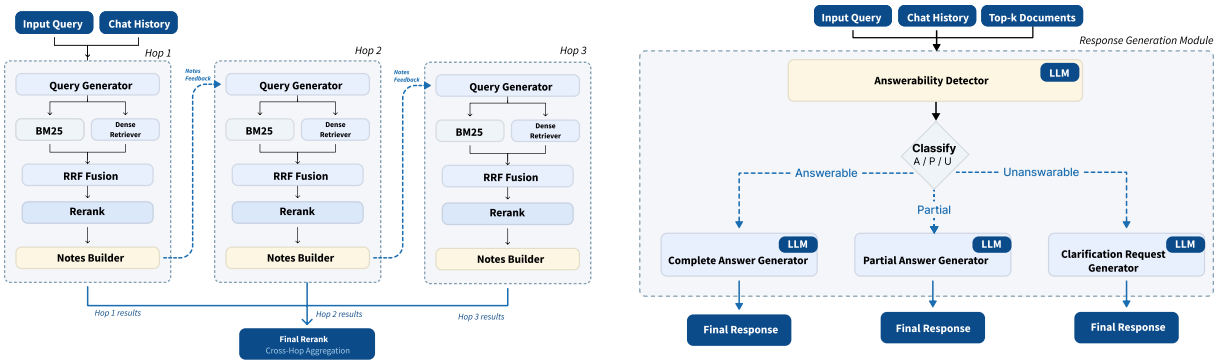

Figure 1: The NLP-CEIA-UFG pipeline. Three iterations of the Query Generator → Hybrid Search → RRF → Jina Reranker → Notes Builder loop each contribute 10 passages, accumulating 30 total, which are globally reranked. An Answerability Classifier then routes generation to one of three specialized templates (Complete, Partial, Clarification) powered by Gemini 2.5 Flash.

- **Partial Answer Generator:** Starts with what can be answered, explicitly states what is missing, and suggests what additional information is needed (for P).
- **Clarification Request Generator:** Acknowledges the lack of information and asks the user for specific clarification (for U).

DSPy/GEPA compilation. All `gpt-oss-120b` modules (Query Generator and Notes Builder) are compiled as a single DSPy program (Khattab et al., 2024) and optimized end-to-end with GEPA (Agrawal et al., 2026) on a 50-sample development set (retrieval recall objective). GEPA produced an improved prompt for the Query Generator (detailed in Appendix A), while the Notes Builder retained its initial prompt. Further optimization details are reported in Appendix B. The Response Generation Module was not optimized.

4 Experimental Setup

Data and tools. System development used the MTRAG dev split; official evaluation used the Task 8 test set (507 tasks). Query Generator, Notes Builder, and Answerability Classifier: `gpt-oss-120b` (OpenAI, 2025). Response generator: Gemini 2.5 Flash (Google DeepMind, 2025). Dense embeddings: google-embedding-001 in Qdrant v1.12. Reranker: `jina-reranker-v3` (Wang et al., 2025). Optimizer: GEPA (Agrawal et al., 2026).

Hyperparameters. Iterations $K = 3$; queries per iteration: 3; BM25 and dense top-k: 10; RRF constant $k_0 = 60$; reranker output: 10 passages.

System	nDCG@5
Top system	**0.5776**
ELSER+QR (baseline)	0.4795
NLP-CEIA-UFG (ours)	0.4502

Table 1: Subtask A results (rank 17/38).

System	RB_agg	RL_F	RB_llm	HM
Top system	—	—	—	**0.5861**
qwen-30b-a3b (baseline)	—	—	—	0.5366
NLP-CEIA-UFG (ours)	0.2707	0.4722	0.4675	0.3774

Table 2: Subtask C results (rank 24/29). HM is the IDK-conditioned harmonic mean of three components: RB_agg (aggregate of BertRec, BertKPrec, and RougeL), RL_F (RAGAS faithfulness), and RB_llm (LLM-as-judge). Under IDK-conditioning, responses classified as IDK score 1.0 on UNANSWERABLE queries and 0.0 on ANSWERABLE queries. Component scores for the top system and baseline were not publicly released.

5 Results and Analysis

5.1 Official Results

Tables 1 and 2 report our scores. Our retrieval performance (nDCG@5 = 0.4502) is competitive, falling only 0.03 points below the organizer baseline. However, there is a substantial 0.16-point gap in Subtask C (HM = 0.3774).

5.2 IDK Over-Triggering

The Subtask C deficit is driven by the Answerability Classifier routing the majority of queries away from the Complete Answer Generator. Table 3 breaks down the evaluations: the benchmark's evaluator flagged 75.5% of all responses (383/507) as IDK, including 69.8% of ANSWERABLE questions.

GT Label	N	IDK $\geq$ 0.5	RB_agg
ANSWERABLE	285	199 (69.8%)	0.2789
PARTIAL	143	121 (84.6%)	0.2009
UNDERSPECIFIED	78	62 (79.5%)	0.1672
UNANSWERABLE	1	1 (—)	0.0000
Overall	**507**	**383 (75.5%)**	**0.2392**

Table 3: Rate of responses classified as IDK (≥ 0.5) by the benchmark's evaluator, by ground-truth label. UNANSWERABLE $N = 1$; statistically uninformative.

IDK Level	N	RB_agg	RL_F
0.0 (not IDK)	124	0.3040	0.8220
0.5 (partial IDK)	204	0.2119	0.5494
1.0 (full IDK)	179	0.2253	0.0418

Table 4: Generation quality by IDK level. The 204 partial-IDK responses (53% of all penalized cases) retain moderate faithfulness (RL_F = 0.549), suggesting they contain useful grounded content despite being penalized.

Crucially, the benchmark's IDK evaluator assigns graded scores: 0.0 (not IDK), 0.5 (partial IDK), and 1.0 (full IDK). Under IDK-conditioning, any score ≥ 0.5 is treated as an abstention. Table 4 decomposes the 383 penalized responses by IDK level.

The 204 partial-IDK responses (IDK = 0.5) are informative. These retain moderate RAGAS faithfulness (RL_F = 0.549) and LLM-judge scores (RB_llm = 0.597), yet are penalized identically to full abstentions under IDK-conditioning. Manual inspection confirms these are predominantly outputs of the Partial Answer Generator: they follow the template "Based on the provided documents, here is the partial answer…", provide substantive content, and then explicitly flag missing information. This indicates two compounding failures: the classifier over-routes ANSWERABLE queries to the Partial path, and the Partial template produces hedged language that triggers IDK detection even when the content is grounded. We attribute the deficit primarily to the classifier, though the Partial Answer template's hedged language may independently trigger IDK detection even on correctly routed queries (see Limitations).

The GEPA optimizer, trained on retrieval recall, provided no signal on answerability or generation quality, leaving the classifier and templates untuned for the Subtask C metric.

Collection	N	IDK %	RB_agg	RL_F
CLAPNQ (Wikipedia)	142	76.8	0.2410	0.4654
Govt	157	77.7	0.2399	0.3839
Cloud	131	65.6	0.2583	0.5170
FiQA	77	85.7	0.2018	0.3557

Table 5: IDK rate and generation metrics by collection.

Dom.	Question	Gold answer	System output
FiQA	*Seller pay 3%?* (T6)	Seller contribution limit is 3% of purchase price.	"I don't have enough information to answer…"
Govt	*Meteoroid?* (T6)	A meteoroid is a small rocky body.	"I'm experiencing technical difficulties…"
CLAPNQ	*Who won Bull Run?* (T1)	Confederate forces won.	"…could you please provide more context…?"

Table 6: Representative outputs penalized as IDK on ANSWERABLE questions. The FiQA and CLAPNQ cases reflect the Clarification Request template; the Govt case is a hard-coded API exception fallback.

5.3 Domain and Turn Analysis

Table 5 breaks down results by collection. FiQA shows the highest IDK rate (85.7%). Financial questions frequently demand specific numeric corroboration; when exact figures were absent, the classifier routed them to the Partial or Clarification generators, triggering the penalty. Cloud shows the lowest IDK rate (65.6%) and highest RL_F (0.517), suggesting structured technical content more easily satisfies the threshold for the Complete Answer Generator.

IDK rates are stable across turns (Turn 1: 73.8%; Turn >1: 75.8%), indicating the penalty is structural rather than caused by context accumulation in later dialogue turns.

5.4 Error Analysis

Table 6 presents three ANSWERABLE questions for which our conversational responses were penalized. These cases illustrate the routing mechanics.

In FiQA and CLAPNQ, the Answerability Classifier routed answerable questions to the Clarification Request Generator. In Govt, the output ("I'm experiencing technical difficulties") is the hard-coded exception fallback from our implementation when the Gemini API encounters a block. When the system correctly routed queries to the Complete Answer Generator (the 124 non-IDK responses, 24.5% of all tasks), generation quality was substantially higher: RB_agg = 0.304, RL_F = 0.822, RB_llm = 0.754—confirming that the retrieval and

synthesis pipeline is sound when allowed to operate within the benchmark's expectations.

6 Conclusion

NLP-CEIA-UFG implements a multi-turn RAG architecture featuring an iterative retrieval loop (Query Generator ↔ Notes Builder) and a 3-way generation router (Complete, Partial, Clarification). Our retrieval performance is competitive (nDCG@5 = 0.4502), but the generation score (HM = 0.3774) is limited by an over-conservative Answerability Classifier that routes 69.8% of ANSWERABLE questions away from the Complete Answer Generator. When the classifier routes correctly, generation quality is high (RL_F = 0.822).

The analysis further reveals that partial-IDK responses (IDK = 0.5) retain moderate faithfulness (RL_F = 0.549) yet are penalized identically to full abstentions under IDK-conditioning—a finding that may inform future benchmark design toward finer-grained IDK penalty tiers distinguishing hedged-but-grounded responses from uninformative ones. Future work should (i) optimize the Answerability Classifier jointly with the generation metric rather than retrieval recall alone, (ii) calibrate generation templates to reduce hedging patterns that trigger IDK detection, and (iii) explore probabilistic routing strategies in which the classifier outputs a confidence score over labels, allowing the abstention threshold to be tuned directly against the IDK-conditioned HM objective.

Ethics Statement

This work uses an existing public benchmark. The system's design emphasizes transparency: by explicitly stating what information is missing (Partial Answer) and requesting context (Clarification Request), the architecture actively prevents hallucination while keeping the user informed. The proprietary models used (`gpt-oss-120b`, Gemini 2.5 Flash) may embed pretraining biases.

Limitations

No component-level ablations were conducted; in particular, the contribution of the iteration count ($K = 3$) was not empirically validated against $K = 1$ or $K = 2$ baselines. A systematic comparison would require full pipeline evaluations across multiple values of K, which was infeasible within the submission window due to API rate limits and latency constraints; we identify this as the primary empirical gap and plan to address it in future work. All inference ran through third-party APIs, which increased latency and limited the number of full pipeline evaluations feasible within the submission window. The GEPA optimization objective (retrieval recall) does not align with the Subtask C evaluation metric; while it improved the Query Generator prompt, the Notes Builder retained its initial prompt. Our post-hoc analysis identifies the IDK penalty at the aggregate level but cannot fully disentangle classifier behavior from template-level effects.

Acknowledgments

The authors thank the SemEval-2026 Task 8 organizers. This work was supported by the Centro de Excelência em Inteligência Artificial (CEIA) at the Universidade Federal de Goiás (UFG).

References

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. 2026. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR 2026)*. Oral presentation. arXiv:2507.19457.

Mohammad Aliannejadi, Julia Kiseleva, Jeffrey Dalton, Avishek Anand, Leif Azzopardi, and Nick Craswell. 2024. iKAT 2023: The interactive knowledge assistants track. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2802–2812. ACM.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2024)*, Vienna, Austria. Oral presentation.

Sebastian Bruch, Siyu Gai, and Amir Ingber. 2023. An analysis of fusion functions for hybrid retrieval. *ACM Transactions on Information Systems*, 42(1):1–35.

Shiyu Chen and Jonas Mueller. 2024. When do LLMs need retrieval augmentation? mitigating LLMs knowledge insufficiency with retrieval augmentation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, pages 9541–9566, Bangkok, Thailand. Association for Computational Linguistics.

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*, pages 758–759, Boston, Massachusetts. ACM.

Google DeepMind. 2025. Gemini 2.5 flash. https://deepmind.google/technologies/gemini/flash/.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023)*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Yannis Katsis, Sara Rosenthal, Lihong He, Vraj Shah, Hubert Larson, Marina Danilevsky, and Lucian Popa. 2025. MTRAG: A multi-turn conversational benchmark for evaluating retrieval-augmented generation systems. *Transactions of the Association for Computational Linguistics*, 13.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Baleen: Robust multi-hop reasoning at scale via condensed retrieval. In *Advances in Neural Information Processing Systems (NeurIPS 2021)*, volume 34, pages 27670–27682, Virtual.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. DSPy: Compiling declarative language model calls into self-improving pipelines. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2024)*, Vienna, Austria.

Tzu-Lin Kuo, Fengyuan Xiong, and Qian Xu. 2025. RAD-Bench: Evaluating large language models capabilities in retrieval augmented dialogues. In *Proceedings of the 2025 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2025)*. 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. In *Advances in Neural Information Processing Systems (NeurIPS 2020)*, volume 33, pages 9459–9474.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925. Apache 2.0. https://huggingface.co/openai/gpt-oss-120b.

Krista Opsahl-Ong, Michael J Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. 2024. Optimizing instructions and demonstrations for multi-stage language model programs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024)*, pages 9340–9366, Miami, Florida. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 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. SemEval-2026 task data paper.

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.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9248–9274, Singapore. Association for Computational Linguistics.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023)*, pages 14918–14937, Singapore. Association for Computational Linguistics.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.

Feng Wang, Yuqing Li, and Han Xiao. 2025. jina-reranker-v3: Last but not late interaction for listwise document reranking. *Preprint*, arXiv:2509.25085. https://huggingface.co/jinaai/jina-reranker-v3.

A Module Prompt Templates

All three gpt-oss-120b modules are implemented as DSPy modules. Full source is available at the repository.

A.1 Iterative Query Generator

The Query Generator is powered by our **optimized prompt**, engineered specifically to handle the complexities of multi-hop retrieval. This comprehensive instruction set receives the `question`, `conversation_history`, and `notes` (containing `search_suggestions`, `missing_info`, and `key_findings`).

System Instructions (Abridged for formatting):
You are a multi-hop retrieval planner. Your job is to produce three new search queries for a hybrid retrieval system (BM25 + dense) and supply two rerank-phrases (`rerank_query` and `final_rerank_query`).
1. Determine Question Type & Answerability: Classify the question (Factoid, How-To, Comparative, Composite, Other). Assess answerability. If Unanswerable, still generate queries targeting missing pieces.
2. Process the Notes (Order matters):

- *search_suggestions:* Use verbatim if it covers a missing element.
- *missing_info:* Ensure at least one query explicitly targets each distinct missing element.
- *key_findings:* Never create a query whose primary intent is to retrieve a fact already known.

3. Craft the Three Queries:

- *Length & Structure:* 8–15 words. Format: specific-keyword(s) → contextual terms → broader semantic cue.
- *BM25/Dense friendly:* Include exact entities for sparse, and capture overall intent for dense.
- *Diversity:* Each query must target a different facet.

4. Edge-Case Handling:

- *All info retrieved:* Return a generic specific query to reinforce knowledge.
- *Redundant queries:* Rewrite using synonyms or different facets.
- *Ambiguous phrase:* Treat as Keyword type and enrich with context.

A.2 Notes Builder

Analyze retrieved documents and build structured notes to guide the next retrieval hop. Be specific and actionable.
Your analysis should:

1. Identify what relevant information was found in the retrieved documents (`key_findings`)
2. Determine what information is still missing to fully answer the question (`missing_info`)
3. Suggest specific search strategies for the next hop (`search_suggestions`)

A.3 Response Generation Routing Prompts

Depending on the Answerability Classifier (A, P, U), one of the following templates is sent to Gemini 2.5 Flash:

1. Complete Answer Generator (ANSWERABLE)

You are a helpful assistant. Answer the question based ONLY on the provided documents and conversation context.
Instructions:

- Use only information from the provided documents.
- If information spans multiple documents, synthesize appropriately.
- Maintain conversational tone appropriate to the context.
- Be specific and cite relevant details.
- If the documents don't fully address the question, acknowledge limitations.

2. Partial Answer Generator (PARTIAL)

Based on the provided documents, you can only partially answer the user's question.
Task: Provide a partial answer and clearly explain what information is missing. Format your response as follows:

1. Start with what you can answer based on the available documents.
2. Clearly state what information is missing or incomplete.
3. Suggest what additional information would be needed for a complete answer.

3. Clarification Request Generator (UNANSWERABLE)

The user has asked a question that cannot be adequately answered with the available documents.
Task: Generate a helpful clarification request that:

1. Acknowledges that you don't have sufficient information.
2. Asks for specific clarification or additional context.
3. Suggests what type of information would be helpful.
4. Maintains a helpful and professional tone.

B GEPA Optimization Details

The 50-sample optimization dataset was stratified across: (i) collection (CLAPNQ, FiQA, Govt, Cloud); (ii) answerability; and (iii) question type (Factoid, How-To, Comparative, Composite, Explanation, Keyword, Opinion, Summarization, Troubleshooting). GEPA received as per-trajectory feedback the complete retrieved document list and the gold documents the system failed to retrieve.

The optimizer ran end-to-end over both retriever modules. For the Query Generator, GEPA accepted prompt mutations that improved retrieval recall; the resulting optimized prompt is shown in Appendix A.1. For the Notes Builder, no demonstrations were added and no mutations were accepted into the Pareto frontier across the full optimization budget—consistent across generations, not an artifact of early stopping—and it retained its initial prompt. Runs tracked via MLflow. DSPy v3.1.2; objective: retrieval recall@10 on the development set.