AILS-NTUA at SemEval-2026 Task 8: Query Diversity via Nested Reciprocal Rank Fusion and Evidence-Guided Agentic Generation for Multi-Turn RAG

Dimosthenis Athanasiou, Maria Lymperaiou, Giorgos Filandrianos, Athanasios Voulodimos, Giorgos Stamou

National Technical University of Athens

dimosthenis.athanas@gmail.com, {marialymp, geofila}@ails.ece.ntua.gr, thanosv@mail.ntua.gr, gstam@cs.ntua.gr

Abstract

We present the AILS-NTUA system for SemEval-2026 Task 8 (MTRAGEval), addressing all three subtasks of multi-turn retrieval-augmented generation: passage retrieval (A), reference-grounded response generation (B), and end-to-end RAG (C). Our unified architecture is built on two principles: (i) a *query-diversity-over-retriever-diversity* strategy, where five complementary LLM-based query reformulations are issued to a single corpus-aligned sparse retriever and fused via variance-aware nested Reciprocal Rank Fusion; and (ii) a multistage generation pipeline that decomposes grounded generation into evidence span extraction, dual-candidate drafting, and calibrated multi-judge selection. Our system ranks **1st in Task A** (nDCG@5: 0.5776, +20.5% over the strongest baseline) and **2nd in Task B** (HM: 0.7698). Empirical analysis shows that query diversity over a well-aligned retriever outperforms heterogeneous retriever ensembling, and that answerability calibration—rather than retrieval coverage—is the primary bottleneck in end-to-end performance.

1 Introduction

Large Language Models (LLMs) have revolutionized several aspects of NLP, from human-like language generation (Tian et al., 2024) to complex reasoning (Giadikiaroglou et al., 2024) and scientific tasks (Kao et al., 2024; Song et al., 2025; Han et al., 2025). However, the static nature of such knowledge does not allow for updates without the retraining burden, rendering a state-of-the-art LLM outdated within a few years. At the same time, LLMs are knowledge-rich but opaque systems that cannot attribute their knowledge of specific facts to a well-defined source, raising concerns regarding their trustworthiness.

Retrieval-Augmented Generation (RAG) addresses both limitations by dynamically consulting external sources at inference time (Lewis et al., 2020). By conditioning generation on retrieved documents, RAG enhances factual accuracy, robustness to knowledge cut-offs, and accountability of claims, becoming an established paradigm for knowledge-intensive tasks (Izacard and Grave, 2021; Menick et al., 2022; Gao et al., 2024).

In practice, RAG systems have demonstrated significant gains in dense and hybrid retrieval, multi-document fusion and grounded generation through citations and self-verification (Arivazhagan et al., 2023; Asai et al., 2024). Yet, their success relies on independent user queries, without incorporating prior conversational context. In real-world scenarios, however, conversations unfold within multi-turn interactions, suggesting the treatment of user queries in the context of the dialog and previously retrieved information. Consequently, RAG systems must become context-aware to avoid out-of-context augmentation and error accumulation across turns.

In light of such demands, *multi-turn RAG* is attracting increasing attention, empowering context-aware question reformulation and dialog-conditioned generation. Prior work has explored multi-turn conversational reasoning (Zheng et al., 2023), active retrieval (Jiang et al., 2023), unanswerability (Dziri et al., 2022) and long-form conversational RAG (Aliannejadi et al., 2024; Kuo et al., 2025), motivating the development of the MTRAG benchmark (Katsis et al., 2025). Unlike other RAG benchmarks, MTRAG features diverse question types, together with answerable, unanswerable, partial, and conversational questions and relevant and irrelevant passages, spanning four domains in a multi-turn setup.

In this paper, we approach multi-turn RAG as a retrieval stability problem: across turns, small reformulation errors accumulate, progressively degrading evidence grounding and leading to confident hallucinations. We design an architecture that explicitly separates context preservation, evidence discovery, and answer commitment. Rather than

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 1340–1365
July 3-4, 2026

relying on retriever ensembles or larger models, we stabilize retrieval through diversity-controlled rewriting over a single corpus-aligned index. We stabilize generation by deferring answer commitment until agreement is reached across multiple evidence judges. In summary, our contributions are: (1) A stability-oriented multi-strategy rewriting method that improves recall without degrading top-k precision by controlling reformulation variance through nested Reciprocal Rank Fusion (RRF) aggregation. (2) An agentic evidence-commitment generation pipeline that reduces conversational hallucinations by separating extraction, drafting, and selection into agreement-driven stages. (3) A unified analysis of multi-turn retrieval failure modes on MTRAG, showing that: (a) retrieval diversity helps depth recall but harms early precision, (b) cross-encoders complement but cannot replace aligned sparse retrieval, (c) LLM-based reranking saturates once grounding quality is sufficient. Our system ranks 1st on Task A and 2nd on Task B, validating the proposed design principles. Code is available at our GitHub[1].

2 Background

Task description MTRAG comprises 110 conversations, each with user–agent history $H_t = \{(u_i, a_i)\}_{i=1}^{t-1}$ (where u_i, a_i are the user utterance and agent response at turn i, and $q_t \equiv u_t$ the current query) and an associated document corpus. Evaluation is decomposed into three subtasks. (1) **Task A - Retrieval Only** provides the user/agent conversation, along with the corresponding document corpus. The output should contain the 10 most relevant passages from the corpus, ordered according to similarity to the last query. (2) **Task B - Generation with Reference Passages (Reference)** again provides the user/agent conversation, but contrary to Task A the relevant query passages are provided. The goal is to generate a passage-grounded agent response to the last user query. (3) **Task C: Generation with Retrieved Passages (RAG)** receives the same inputs as Task A. Then, participants have to retrieve up to 10 relevant passages and use them to generate an appropriate response to the last user query. A thorough data exploration is provided in App. A.

Related work RAG has served as a widely adopted framework for conditioning generation on external knowledge, enabling parameter-free knowledge updates (Lewis et al., 2020). Subsequent work explored multi-passage architectures for integrating multiple relevant contexts (Izacard and Grave, 2021), alongside retriever—generator training for enhanced retrieval awareness (Izacard et al., 2023), retrieval augmentation for black-box models (Shi et al., 2024), and self-reflective mechanisms to improve grounding (Asai et al., 2024). In a parallel avenue, conversational research has formalized multi-turn retrieval under context-dependent queries (Dalton et al., 2020), often via question rewriting into standalone queries (Vakulenko et al., 2021; Anantha et al., 2021), with later work addressing topic switching and context drift (Adlakha et al., 2022). The integration of retrieval and generation for multi-turn interactions has led to the introduction of self-checking strategies tailored to conversational QA (Ye et al., 2024) and active retrieval methods for adapting evidence selection during generation or across turns (Jiang et al., 2023). At the same time, unanswerability emerges as a critical challenge for faithful information-seeking dialogue (Dziri et al., 2022), while evaluation targets long-form, retrieval-augmented dialogues (Aliannejadi et al., 2024; Kuo et al., 2025). Finally, automated evaluation frameworks for RAG have been proposed to diagnose retrieval quality and grounding-related failure modes (Es et al., 2024).

3 System Overview

We address all three subtasks with a unified architecture. For retrieval (Task A), we treat conversational search primarily as a query formulation problem and aggregate diverse reformulations over a single corpus-aligned index. For generation (Tasks B and C), we adopt an agentic approach that decomposes grounded response generation into answerability detection, evidence identification, candidate drafting, and evidence-aware selection.

Task A: Multi-Strategy Retrieval Our retrieval system addresses multi-turn conversational search through a four-stage pipeline (Figure 1a).

(1) **Query Rewriting.** Given conversation history $H = \{(u_1, a_1), \ldots, (u_{t-1}, a_{t-1})\}$ and the current user query q_t, we generate five complementary reformulations, each targeting distinct failure modes: *Minimal*—resolves coreferences and conversational omissions; *Corpus-Specific*—adapts the query to domain terminology; *Hypothetical Document Embedding*—generates a hypothetical an-

[1] https://github.com/dimosathan/multiturn-RAG

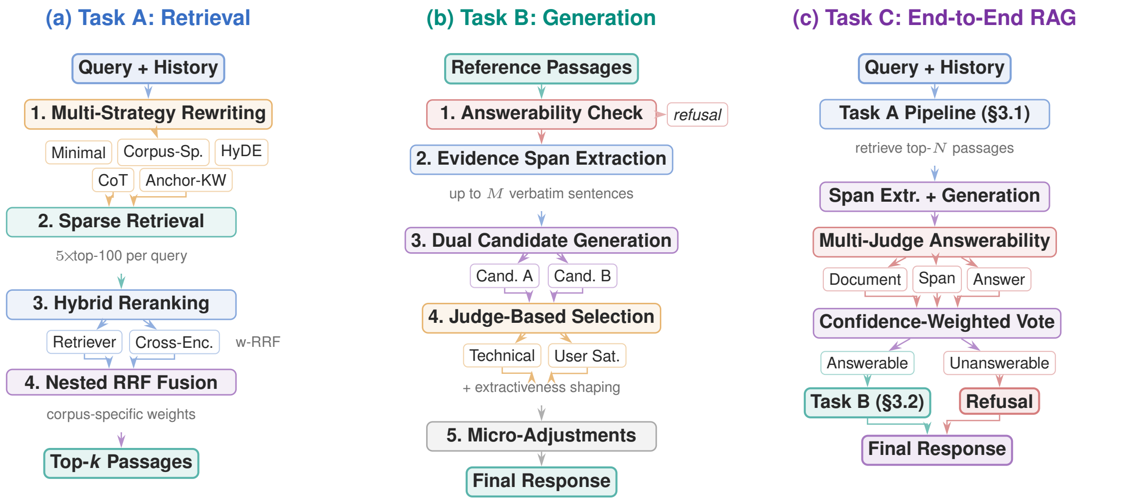

Figure 1: System architecture: (a) Task A retrieval pipeline, (b) Task B generation pipeline, (c) Task C end-to-end RAG with answerability gate.

swer passage, bridging the query–document vocabulary gap (Gao et al., 2023); *Chain-of-Thought (CoT)*—expands the information need through stepwise reasoning; *Anchor-Keyword*—extracts salient entities and keywords optimized for sparse lexical matching. All strategies use XML-formatted prompts with instructions and few-shot examples (App. B.13). 2 **Retrieval.** Each reformulation is issued to the retriever, producing five ranked lists $\mathcal{R}_1, \ldots, \mathcal{R}_5$. 3 **Hybrid Reranking.** We fuse retriever and reranker rankings via weighted RRF:

$$\text{Score}(d) = \frac{1}{k + r_{\text{E}}(d)} + \alpha \cdot \frac{1}{k + r_{\text{R}}(d)} \quad (1)$$

where r_{E} and r_{R} denote retriever and reranker ranks, k controls rank decay, and α balances the two signals. 4 **Nested Fusion.** *Minimal* and *Corpus-Specific* produce stable, high-precision rankings, while the remaining three exhibit higher variance but complementary coverage. We propose a two-level fusion: **Level 1** pre-aggregates the three high-variance strategies into a *Weak Consensus* ranking; **Level 2** combines this with the two stable strategies via corpus-specific weighted RRF. The final score for each passage d is:

$$\text{Score}_{\textit{final}}(d) = \sum_{s} w_s^{(c)} \cdot \frac{1}{k^{(c)} + \text{rank}_s(d)} \quad (2)$$

where s ranges over the three Level 2 inputs and $w_s^{(c)}$, $k^{(c)}$ are corpus-dependent parameters (Table 2). The top-k passages by $\text{Score}_{\textit{final}}$ form the retrieval output.

Task B: Agentic Generation Pipeline We treat grounded response generation as a multi-stage decision process (Figure 1b). 1 **Answerability Classification.** Each turn is classified based on context as *unanswerable* (empty context→ triggering a calibrated refusal) or *answerable/partial* (proceeding to generation). 2 **Evidence Span Extraction.** An extraction module identifies verbatim supporting sentences from provided passages. These spans replace full passages as generator input. 3 **Dual Candidate Generation.** Two response candidates are generated, conditioned on extracted spans, conversation history, and question-type-specific length guidance, with strict grounding constraints. 4 **Judge-Based Selection.** A *Technical Judge* evaluates faithfulness and completeness against extracted spans, while a *User Satisfaction Judge*, invoked on a stochastic subset of turns, evaluates naturalness. Selection combines judge preferences with an *Extractiveness Shaping* term that penalizes unsupported abstraction and verbatim copying, and discourages refusal when evidence exists. 5 **Micro-Adjustments.** A light post-editing pass addresses residual length and extractiveness bound violations, and removes formulaic hedging phrases (details in App. C.3).

Task C: End-to-End RAG Task C integrates retrieval and generation while additionally deciding whether retrieved evidence justifies answering. Unlike Tasks A and B, retrieved passages may be irrelevant and answerability is often ambiguous.

Our system retrieves the top-5 passages via the Task A pipeline, extracts spans, and generates two candidate answers. Three specialized LLM judges evaluate answerability from complementary views: a **document judge** checks passage relevance, a **span judge** verifies evidence coverage, and an **answer judge** evaluates response adequacy.

An arbiter aggregates the binary verdicts using confidence-weighted voting with dissenter override. Answerable turns continue through the Task B pipeline; otherwise a calibrated refusal is produced.

4 Experimental setup

Dataset We evaluate on the MTRAG benchmark (Katsis et al., 2025): 110 multi-turn conversations (842 turns) across four domains (ClapNQ, FiQA, Govt, Cloud). Documents are in English and segmented into 512-token passages (stride 100) and indexed using Elasticsearch Learned Sparse Encoder (ELSER v1). Task A uses the 777 answerable and partially answerable tasks; Tasks B and C use all 842 of the development set, on which tuning is conducted; results are reported on held-out test set(Rosenthal et al., 2026b,a).

Characteristic	ClapNQ	FiQA	Govt	Cloud
Documents	4,293	57,638†	7,661	8,578
Passages	183,408	61,022	49,607	72,442
Avg. passages/doc	42.7	1.1	6.5	8.4
Domain	Wikipedia	Finance	Government	Tech docs

Table 1: MTRAG corpus statistics. †FiQA contains individual forum posts.

Task A Configuration. Rewriting strategies use DeepSeek-V3.2 (τ=0.0) with 6 user and 3 assistant turns as context. Each reformulation retrieves top-100 passages from ELSER v1, selected after benchmarking 9 alternatives (App. B.1). Hybrid reranking uses Cohere Rerank v4 with k=60 and α=0.5 (Eq. 1). The weak-consensus group aggregates HyDE, CoT, and Anchor-Keyword with equal weights (k_{internal}=40); final fusion uses corpus-specific weights tuned on dev set (Table 2); uniform weights yield less than 1% R@5 degradation, confirming robustness to weight selection (App. B.9).

Parameter	ClapNQ	FiQA	Govt	Cloud
k_{final}	20	60	40	20
w_{Minimal}	0.55	0.45	0.65	0.65
$w_{\text{Corpus-Spec}}$	0.40	0.40	0.25	0.30
w_{WeakCons}	0.05	0.15	0.10	0.05

Table 2: *Nested RRF* parameters per domain.

Task B Configuration. We route pipeline stages to three models: DeepSeek-V3.2 for span extraction (τ=0.0, up to 8 spans) and technical judging; GPT-4o for dual candidate generation (τ_1=0.0, τ_2=0.1); and GPT-4o-mini for the *User Satisfaction Judge* (60% invocation rate) and micro-adjustments (τ=0.1). Target length is question-type dependent (App. C.5). Selection combines technical scores, user preference, and extractiveness calibration (App. C.3).

Task C Configuration. All three judges and the arbiter use GPT-4o (τ=0.0). Refusal responses are capped at 25 words. Answerable turns reuse the full Task B pipeline configuration.

Evaluation Metrics For Task A, we report Recall@k and nDCG@k ($k \in \{5, 10\}$). For Tasks B and C, we use three metrics from the MTRAG evaluation framework (Katsis et al., 2025): RB_{alg} (reference-based algorithmic), RB_{llm} (reference-based LLM judge), and RL_{F} (reference-less faithfulness). The official ranking metric is the harmonic mean (HM) of these three scores.

5 Results and Analysis

Official Results. Table 3 reports leaderboard performance. Our system ranks first on Task A (nDCG@5: 0.5776, +20.5% over the strongest baseline), validating the single-retriever, multi-query hypothesis. On Task B we rank second (HM: 0.7698 vs. top: 0.7827), with strong faithfulness (RL_{F}=0.8971) and LLM quality (RB_{llm}=0.8321), indicating effective grounding. The Task B→C gap (0.7698→0.5409) reflects compounding retrieval errors, most visibly in RB_{alg} (0.6327→0.3998). Relative improvements between ablation variants remain consistent across development and test splits, suggesting that the ablation findings reported below generalize beyond the development set.

Task	Metric	Ours	Rank
A	nDCG@5	**0.5776**	1/38
B	HM	**0.7698**	2/26
C	HM	0.5409	11/29

Table 3: Official test set results.

Task A Ablations. ELSER v1 outperforms all alternative retrievers by +8.6 R@5 (App. B.1), though rewriting improves all nine consistently, confirming gains stem from query diversity. Despite this, alternative retrievers surface unique gold

Configuration	R@5	Δ
No rewriting	0.483	–
+ Minimal	0.527	+9.1%
+ Corpus-Specific	0.558	+15.5%
+ CoT	0.573	+18.6%
+ HyDE	0.584	+20.9%
+ Anchor-Keyword	0.591	+22.4%
+ Nested RRF fusion	**0.607**	+25.7%

Table 4: Task A: ablations on incremental rewriting strategy contribution (dev set).

documents, motivating an ensemble. However, as rewriting improves ELSER (R@5: 0.483→0.607), ensemble gains reverse: unique documents from other retrievers appear at ranks 37–54 on average and introduce fusion noise. Thus R@100 improves but R@5/10 degrades, favoring single-retriever, multi-query architecture. Among rewriting strategies, *Corpus-Specific* achieves the best single-strategy performance (0.541), showing that performance varies across domains (App. B.10), but the full ensemble under nested RRF (0.607) outperforms it by 6.6 points; this confirms that complementarity matters more than individual strategy quality. Cohere Rerank v4 fused with ELSER yields +12.7%, while LLM-based reranking and Passage-Informed Rewriting (PIR)—a second-stage retrieval using initial results as context— provide no gains: once R@100 exceeds ∼0.95, the retrieval bottleneck shifts from coverage to top-k ranking precision, where additional signals add more noise than information (App. B.6).

Task B Ablations. The full five-stage pipeline yields +8.9 HM over single greedy generation, with judge-based selection and extractiveness calibration as the largest contributors;thus, quality control at generation time outweighs prompt engineering. Model routing (DeepSeek-V3.2 for extraction and judging, GPT-4o for generation, GPT-4o-mini for micro-adjustments) achieves comparable results to uniform GPT-4o usage at one-third the cost (App. C.2). Extracting verbatim spans before generation reduces hallucination, and the 4-gram extractiveness band (dev mean: 36.2%) balances faithfulness and naturalness, with responses outside the band showing degraded scores on both axes.

Task C: Answerability as a Structural Bottleneck. The TaskB→C performance drop is driven by answerability classification rather than retrieval or generation quality. Across architectures, all configurations exhibit a precision–recall trade-off:

Configuration	HM
Single generation (τ=0.0)	0.654
+Span extraction	0.688
+ Dual generation (random select)	0.706
+ Technical Judge	0.725
+ User Satisfaction Judge (60%)	0.738
+ Micro-adjustments	**0.743**

Table 5: Task B: generation pipeline ablation (dev set).

improving UNANSWERABLE recall consistently reduces ANSWERABLE recall, and none surpass 25% UNANSWERABLE F1 (Table 6). Our system achieves 83.9% accuracy but only 21.8% UNANSWERABLE recall, revealing a strong acceptance bias. The pattern remains stable across prompt variants and voting rules, suggesting a structural bias toward semantic plausibility over abstention. Error analysis shows document- and span-grounded judges often correctly reject insufficient evidence, while the answer-grounded judge accepts plausible responses. LLM judges prioritize semantic plausibility over evidence availability.

Why multiple judges? Removing the answer-based judge increases UNANSWERABLE recall but sharply reduces ANSWERABLE recall by rejecting partially supported responses. The judges therefore operate on distinct signal spaces: document/span judges detect evidence presence, while the answer judge evaluates semantic adequacy.

The multi-judge framework functions as an *uncertainty decomposition* mechanism rather than a simple voting ensemble. Excluding the answer judge yields strict verification; including it enables partially supported answers at the cost of conservative rejection. This trade-off persists across all tested configurations.

Method	ANS R	UNANS R	UNANS F1
Always Answer	100.0	0.0	0.0
Single Judge	94.6	18.2	21.5
Pipeline + Verif.	92.3	27.3	23.1
Multi-Judge (ours)	95.8	21.8	24.0
Multi-Judge + Calib.	92.3	27.3	23.1

Table 6: Table 6: Answerability detection performance on the development set.ANS R: recall on answerable turns;UNANS R: recall on unanswerable turns;UNANS F1: F1 score for unanswerable detection..

6 Conclusion

We present a multi-turn conversational RAG system that stabilizes retrieval through corpus-aligned query diversity, and generation via agreement-driven agentic stages. Nested-RRF multi-strategy rewriting improves retrieval over unaugmented baselines, while staged generation enforces grounding. Error analysis identifies answerability classification as the dominant end-to-end bottleneck, highlighting a key direction for future work.

Acknowledgements

This work was carried out within the framework of the Pharos AI Factory project, funded by the European High-Performance Computing Joint Undertaking (EuroHPC JU) under Grant Agreement No. 101234269 as part of the Horizon Europe and by the Greek Public Investments Program programme.

Limitations

Our system relies exclusively on proprietary API-based models (GPT-4o, DeepSeek-V3.2) for generation, judging, and rewriting, which limits reproducibility and introduces cost and latency constraints at scale. No component of the pipeline was fine-tuned on the MTRAG data; all gains stem from prompt engineering and architectural design choices, meaning that task-specific training could potentially yield further improvements that our system does not explore. The entire pipeline was tuned on the development set, which differs substantially from the test set in answerability distribution, turn structure (all test turns are non-first turns), and domain balance (FiQA underrepresented, Govt overrepresented at test time). Hyperparameters calibrated on dev — including the answerability confidence threshold (0.7), the extractiveness target band ([0.28, 0.38]), and the nested RRF corpus weights — were not adapted to the test distribution. This mismatch is the primary driver of the Task C performance gap and suggests that distribution-adaptive calibration, particularly for answerability classification, is an important direction for future work.Additionally, evaluation on MTRAG carries inherent biases: retrieval metrics favor ELSER-based systems since annotators primarily reviewed ELSER-retrieved passages during benchmark construction, while generation metrics depend on reference answers and LLM judges that may reward particular phrasing styles over genuinely correct but differently-worded responses.

References

Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2022. TopiOCQA: Open-domain conversational question answering with topic switching. *Transactions of the Association for Computational Linguistics*, 10:468–483.

Mohammad Aliannejadi, Zahra Abbasiantaeb, Shubham Chatterjee, Jeffrey Dalton, and Leif Azzopardi. 2024. Trec ikat 2023: A test collection for evaluating conversational and interactive knowledge assistants. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 819–829, New York, NY, USA. Association for Computing Machinery.

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-domain question answering goes conversational via question rewriting. In *NAACL*.

Manoj Ghuhan Arivazhagan, Lan Liu, Peng Qi, Xinchi Chen, William Yang Wang, and Zhiheng Huang. 2023. Hybrid hierarchical retrieval for open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10680–10689, Toronto, Canada. Association for Computational Linguistics.

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*.

Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. Trec cast 2019: The conversational assistance track overview. *Preprint*, arXiv:2003.13624.

Nouha Dziri, Ehsan Kamalloo, Sivan Milton, Osmar Zaiane, Mo Yu, Edoardo M. Ponti, and Siva Reddy. 2022. Faithdial: A faithful benchmark for information-seeking dialogue. *Transactions of the Association for Computational Linguistics*, 10:1473–1490.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.

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.

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.

Panagiotis Giadikiaroglou, Maria Lymperaiou, Giorgos Filandrianos, and Giorgos Stamou. 2024. Puzzle solving using reasoning of large language models: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11574–11591, Miami, Florida, USA. Association for Computational Linguistics.

Yang Han, Ziping Wan, Lu Chen, Kai Yu, and Xin Chen. 2025. From generalist to specialist: A survey of large language models for chemistry. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1106–1123, Abu Dhabi, UAE. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24(1).

Zhengbao Jiang, Frank 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*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Kuei-Chun Kao, Ruochen Wang, and Cho-Jui Hsieh. 2024. Solving for X and beyond: Can large language models solve complex math problems with more-than-two unknowns? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16821–16843, Miami, Florida, USA. 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.

Tzu-Lin Kuo, FengTing Liao, Mu-Wei Hsieh, Fu-Chieh Chang, Po-Chun Hsu, and Da-shan Shiu. 2025. RAD-bench: Evaluating large language models' capabilities in retrieval augmented dialogues. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 868–902, Albuquerque, New Mexico. 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 *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. 2022. Teaching language models to support answers with verified quotes. *Preprint*, arXiv:2203.11147.

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.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384, Mexico City, Mexico. Association for Computational Linguistics.

Zhangde Song, Jieyu Lu, Yuanqi Du, Botao Yu, Thomas M. Pruyn, Yue Huang, Kehan Guo, Xiuzhe Luo, Yuanhao Qu, Yi Qu, Yinkai Wang, Haorui Wang, Jeff Guo, Jingru Gan, Parshin Shojaee, Di Luo, Andres M Bran, Gen Li, Qiyuan Zhao, and 37 others. 2025. Evaluating large language models in scientific discovery. *Preprint*, arXiv:2512.15567.

Yufei Tian, Tenghao Huang, Miri Liu, Derek Jiang, Alexander Spangher, Muhao Chen, Jonathan May, and Nanyun Peng. 2024. Are large language models capable of generating human-level narratives? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17659–17681, Miami, Florida, USA. Association for Computational Linguistics.

Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2021. Question rewriting for conversational question answering. In *Proceedings of the 14th ACM International Conference on Web*

Search and Data Mining, WSDM '21, page 355–363, New York, NY, USA. Association for Computing Machinery.

Linhao Ye, Zhikai Lei, Jianghao Yin, Qin Chen, Jie Zhou, and Liang He. 2024. Boosting conversational question answering with fine-grained retrieval-augmentation and self-check. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2301–2305, New York, NY, USA. Association for Computing Machinery.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

A Dataset Details

A.1 Development Set

MTRAG (Katsis et al., 2025) consists of 110 human-authored multi-turn conversations comprising 842 annotated turns. For Task A, unanswerable turns are excluded, yielding 777 evaluation queries. Conversations average 7.7 turns and exhibit high passage diversity (16.9 unique relevant passages per conversation), making retrieval and generation performance strongly turn-dependent. Approximately 87% of turns are non-standalone, and conversations contain on average 1.3 co-references. Reference answers are high-quality: 92% required human repair during creation (mean edit similarity 60.7 ROUGE-L), indicating that even strong LLMs frequently fail to meet the benchmark's quality constraints without post-editing.

Corpora. Table 7 summarizes the four corpora. ClapNQ and FiQA are sourced from existing datasets (Wikipedia and StackExchange Finance, respectively), while Govt and Cloud were collected specifically for MTRAG. All corpora were chunked into 512-token passages with 100-token overlap; benchmark indexing uses ELSER v1 (ElasticSearch 8.10). Passage counts per document are heavily skewed: ClapNQ ranges from 1 to 194 passages (median: 31), while all FiQA documents are single-passage.

The corpora intentionally differ in structure and style. ClapNQ comprises long encyclopedic articles (median 31 passages/doc), FiQA consists of short, subjective forum posts (1 passage ≈ 1 post), Govt contains formal policy and instructional content, and Cloud is dense technical documentation (up to 64k words per document). This heterogeneity probes complementary retrieval and generation challenges.

Corpus	Domain	Docs	Passages	Avg P/D
ClapNQ	Wikipedia	4,293	183,408	42.7
FiQA	Finance forum	57,638	61,022	1.1
Govt	Government	7,661	49,607	6.5
Cloud	Technical docs	8,578	72,442	8.4
Total		**78,170**	**366,479**	

Table 7: Corpus statistics. FiQA "Docs" correspond to individual forum posts (atomic documents).

Task dimensions. Each turn is annotated along three axes: (i) *Question type* (10 categories; e.g., Factoid 33%, Summarization 23%, Explanation 19%), where multiple labels may apply; (ii) *Multi-turn type*, distinguishing Follow-up (74%) from Clarification (13%) for non-first turns; and (iii) *Answerability*, with Answerable (84%), Partially Answerable (8%), Unanswerable (7%), and Conversational (1%).

Answerability per domain. Table 8 shows the answerability distribution across domains. The distribution is broadly uniform: all four domains exhibit 82–86% answerable turns, with FiQA showing a slightly higher conversational rate (3%) due to its informal, opinion-driven content, and Govt exhibiting the highest partial answerability (11%) reflecting its instructional content where passages often address a question only partially.

Domain	Ans	Partial	Unans	Conv
ClapNQ	86%	7%	7%	0%
FiQA	82%	9%	7%	3%
Govt	83%	11%	6%	0%
Cloud	86%	5%	7%	1%
Overall	84%	8%	7%	1%

Table 8: Answerability distribution per domain (dev set, 842 turns). Percentages are row-normalized.

Question type per domain. Table 9 reports the three most frequent question types per domain. ClapNQ and Govt share a similar profile dominated by Factoid and Summarization questions, while FiQA is uniquely characterized by Opinion questions (reflecting its user-generated forum content) and Cloud distributes more evenly across types with lower top-3 concentration. Figure 2 visualizes the full question-type distribution across all domains.

Domain	Top-3 Question Types	Coverage
ClapNQ	Factoid, Summarization, Explanation	71%
FiQA	Opinion, Factoid, Explanation	51%
Govt	Factoid, Summarization, Explanation	63%
Cloud	Factoid, Explanation, Summarization	50%

Table 9: Top-3 question types per domain (dev set). Coverage indicates the percentage of turns with one of the three most frequent types. FiQA's lower coverage reflects its more diverse question distribution.

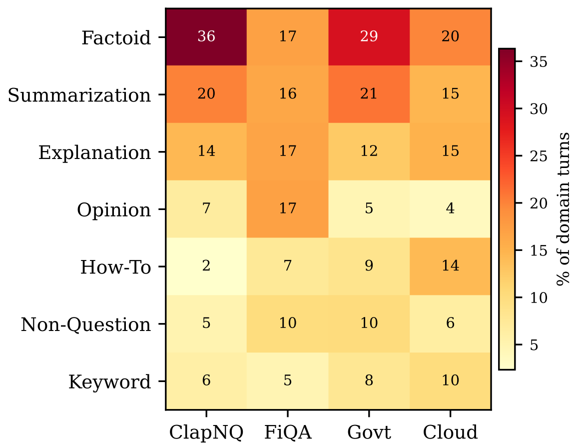

Figure 2: Question-type distribution per domain (dev set, % of domain turns). FiQA's Opinion-heavy profile and Cloud's even distribution across types stand in contrast to the Factoid/Summarization-dominated ClapNQ and Govt corpora.

Reference answer statistics. Reference answer lengths range from 5 to 319 words (mean: 90.9, median: 84, std: 57.7), with Summarization questions producing the longest references (mean: 121 words) and Factoid questions the shortest (mean: 77 words). This variance motivated the question-type-specific length targets in our generation pipeline (App. C.5). Figure 3 shows the full distribution by question type.

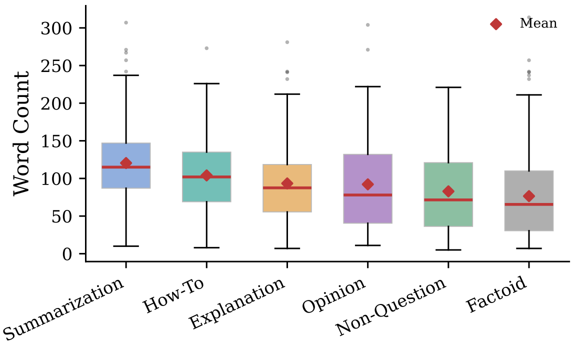

Figure 3: Reference answer length distribution by question type (dev set). Red diamonds indicate means; Summarization answers are significantly longer, motivating type-specific generation targets.

Non-standalone phenomena. Figure 4 breaks down the types of context-dependency present in non-first turns. Pronoun coreference and implicit topic carryover are the dominant phenomena, directly motivating the multi-strategy rewriting approach described in System Overview: Minimal rewriting resolves coreferences, Corpus-Specific rewriting addresses terminology gaps, and Chain-of-Thought handles multi-faceted queries requiring inference across turns.

Representative examples. Table 10 illustrates three turn types that pose distinct challenges for retrieval and generation.

Type	User Query	Challenge
Follow-up (coreference)	"How much does it cost?" refers to IBM Cloud Object Storage, 3 turns prior	Non-standalone query; retriever sees no entity without rewriting
Unanswerable (temporal)	"Where do the Cardinals play this week?" corpus contains only 2017 schedule	Topically relevant but temporally mismatched; high hallucination risk
Partial (incomplete)	"Is it worth having a web chat widget?" docs describe features, not subjective value	Evidence supports description but not the evaluative judgment

Table 10: Representative MTRAG challenge examples.

Key challenges. MTRAG stresses multi-turn retrieval and robust generation under realistic information gaps:

- **Later-turn degradation.** ELSER R@5 drops from 0.89 (first turn) to 0.47 (later turns), a 51% relative decrease (Figure 8).
- **Passage diversity.** 16.9 unique relevant passages per conversation require turn-adaptive retrieval.

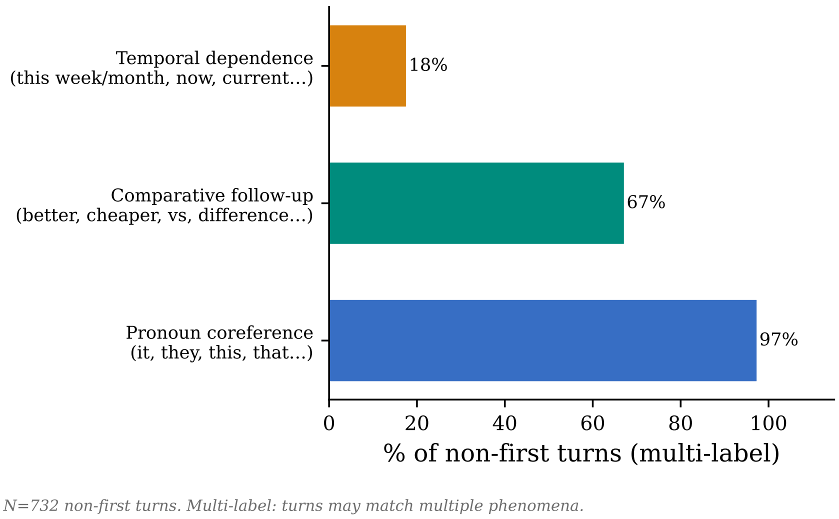

Figure 4: Non-standalone phenomena in dev set non-first turns ($N = 732$, multi-label). Pronoun coreference and implicit topic carryover are the most frequent sources of query underspecification, motivating history-aware query rewriting.

- **Unanswerable turns.** GPT-4o $\mathrm{RB}_{\mathrm{alg}}$ drops from 0.48 (answerable) to 0.20 (unanswerable), a 58% decrease.

- **Domain variance.** FiQA is consistently the hardest corpus due to informal language and high lexical variability.

A.2 Test Set

The test set was released without ground-truth labels during the evaluation phase; all official results were obtained through the evaluation server with frozen configurations. Ground-truth annotations were subsequently released for post-hoc analysis. Table 11 compares the two splits, revealing substantial distributional differences.

The test set differs from the development set in three critical respects. First, each test conversation consists of a single evaluated turn (avg 1.0 turns/conv) embedded within a multi-turn history, compared to 7.7 evaluated turns per conversation in dev. This means all 507 test turns are non-first turns requiring contextual understanding, without the “easy” first-turn queries that boost dev-set averages. Figure 5 illustrates this structural difference in the distribution of evaluated turn positions. Second, the unanswerable rate nearly triples (**19.1%** vs. 6.5%), substantially amplifying the answerability bottleneck —systems tuned on a 7% unanswerable rate face a far harder classification task at test time. Third, FiQA is underrepresented (15.2% vs. 23.6%) while Govt is overrepresented (31.0% vs. 25.4%), shifting the domain balance toward formal, instructional content. Figure 6 visualizes the answerability shift.

Characteristic	**Dev**	**Test**
Conversations	110	507
Total turns	842	507
Avg turns/conv	7.7	**1.0**
Answerable (%)	84.2	**56.2**
Partial (%)	8.1	9.3
Unanswerable (%)	6.5	**19.1**
Conversational (%)	1.2	0.0
Domain distribution (% of turns)		
ClapNQ	26.6	28.0
FiQA	23.6	**15.2**
Govt	25.4	**31.0**
Cloud	24.3	25.8
Question type (top-3, % of turns)		
Factoid	33.3	44.8
Summarization	23.2	25.0
Explanation	18.8	30.0

Table 11: Development vs. test set comparison. Notable distributional shifts are shown in **bold**. The test set is substantially harder due to a 3× higher unanswerable rate and single-turn evaluation structure.

The answerability shift is not uniform across domains. Table 12 shows that Cloud exhibits the highest test-set unanswerable rate (27%), followed by Govt (17%), FiQA (16%), and ClapNQ (15%). This contrasts with the dev set where all domains had similar unanswerable rates (6–7%), suggesting that the test set deliberately stresses answerability classification more heavily than the development set. Figure 7 visualizes these per-domain shifts jointly across both splits.

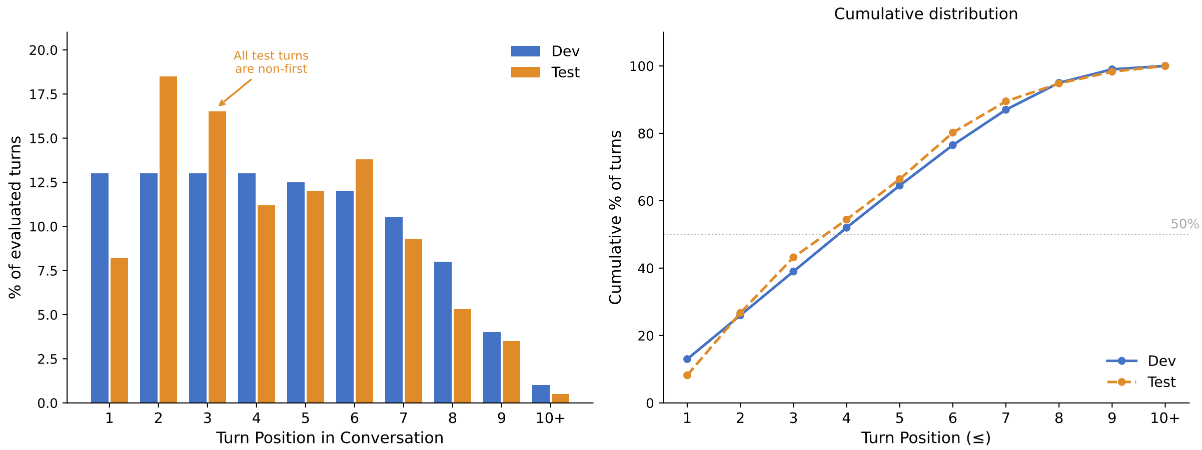

Figure 5: Turn position distribution for evaluated turns in the development (blue) and test (orange) sets. The dev set spans positions 1 10+, while all 507 test turns are non-first turns eliminating the easy first-turn queries that inflate dev-set averages and ensuring every test turn requires contextual understanding.

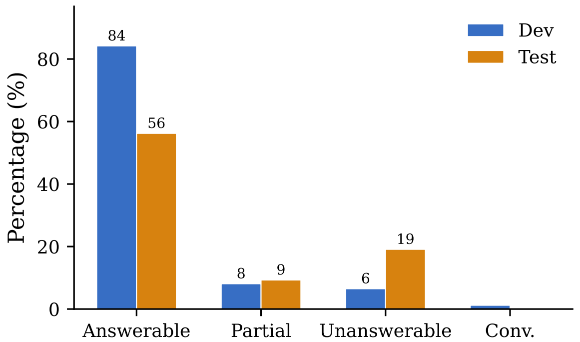

Figure 6: Answerability distribution: development vs. test set. The test set contains nearly 3× more unanswerable turns and 28 percentage points fewer answerable turns, creating a substantially harder classification setting.

Domain	Ans	Partial	Unans	Conv
ClapNQ	46%	13%	15%	0%
FiQA	66%	9%	16%	0%
Govt	56%	11%	17%	0%
Cloud	62%	4%	27%	0%
Overall	56%	9%	19%	0%

Table 12: Answerability distribution per domain (test set, 507 turns). All domains show substantially higher unanswerable rates than in development, with Cloud exhibiting the most severe shift.

A.3 Evaluation Metrics

MTRAG evaluates generation quality using a composite score designed to penalize systems that perform well on only one axis. The primary leaderboard score for the SemEval-2026 shared task is the *Harmonic Mean* (HM) over three components:

$$\mathrm{HM} = \mathrm{HarmonicMean}\Big(\mathrm{RB}_{\mathrm{alg}},\ \mathrm{RL}_{\mathrm{F}},\ \mathrm{RB}_{\mathrm{llm}}\Big) \tag{3}$$

$\mathbf{RB_{alg}}$ is a reference-based algorithmic score computed as the harmonic mean of Bert-Recall, Bert-K-Precision, and ROUGE-L against the human-edited reference answer (Katsis et al., 2025). Bert-Recall approximates *completeness* by measuring semantic overlap between the model response and the reference answer. Bert-K-Precision compares the model response to the retrieved passages, approximating *faithfulness*. ROUGE-L captures phrase-level overlap with the reference, serving as a proxy for *appropriateness*. In our analysis, this metric is the most sensitive to paraphrase and stylistic variance: a faithful answer may still be penalized if it uses different phrasing than the reference.

$\mathbf{RL_F}$ is a reference-less faithfulness metric from the RAGAS framework (Es et al., 2024), assessing whether generated claims are grounded in the provided passages without requiring the reference answer. Unlike the other two metrics, $\mathrm{RL_F}$ measures faithfulness to the *passages* rather than similarity to the *reference*.

$\mathbf{RB_{llm}}$ is a reference-based LLM-judged score adapted from RAD-Bench (Kuo et al., 2025) that compares the model response to the reference answer using multiple LLM evaluators (median aggregated) (Katsis et al., 2025). The evaluation prompt assesses three criteria:

- *Faithfulness*: whether the response is grounded in the provided passages and prior conversation turns, without hallucinations.

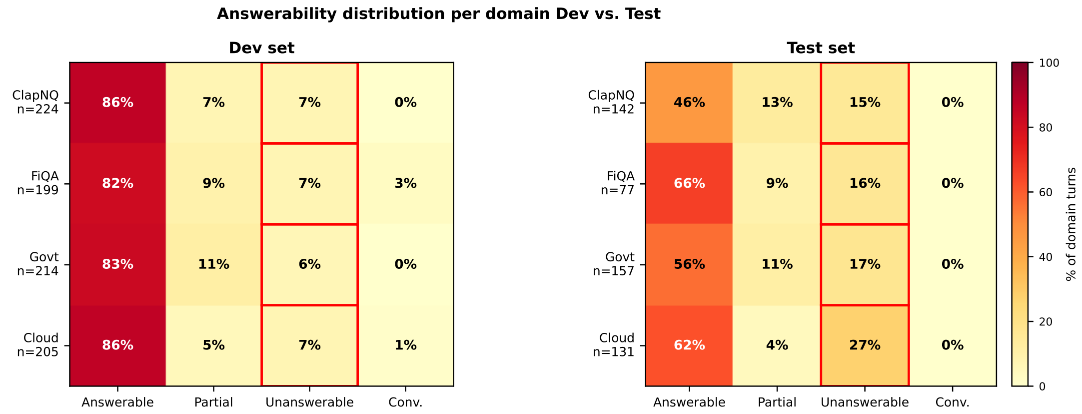

Figure 7: Answerability distribution per domain (% of domain turns), development set (left) vs. test set (right). Red borders highlight the Unanswerable column. Cloud exhibits the most severe shift: unanswerable turns rise from 7% (dev) to 27% (test), the highest unanswerable rate across all domains.

- *Appropriateness*: whether the response directly addresses the current-turn question and handles answerability correctly.
- *Completeness*: whether the response includes all information from the passages relevant to the question.

Note that *Naturalness*, the fourth FANC criterion used during MTRAG's human evaluation (Katsis et al., 2025), is *not* assessed by the RB_{llm} judge.

Answerability conditioning. Prior to computing metrics, an IDK ("I Don't Know") judge determines whether the response contains a full, partial, or no answer. Metric scores are conditioned on the interaction between ground-truth answerability and IDK detection: correct refusals on unanswerable turns receive a score of 1.0, while incorrect refusals on answerable turns receive 0.0. This conditioning makes answerability classification a high-stakes binary decision that directly impacts the composite HM score—and explains why the $3\times$ increase in unanswerable turns from dev to test (Table 11) disproportionately affects Task C performance.

A.4 Official Test Set Results

Table 13 reports our system's performance on the held-out test set alongside the top-performing baseline per task. We rank **1st on Task A** (nDCG@5: 0.5776), surpassing the strongest retrieval baseline by +20.5%. On Task B we rank **2nd out of 26** (HM: 0.7698 vs. top: 0.7827), with strong faithfulness (RL_F=0.8971) and LLM-judged quality (RB_{llm}=0.8321). Task C ranks 11th, with the Task B→C gap (0.7698→0.5409) driven primarily by RB_{alg} degradation (0.6327→0.3998), consistent with our answerability analysis in results.

B Task A: Retrieval Details

B.1 Retriever Comparison

We evaluated 9 retrievers spanning lexical, sparse, and dense families on the development set (777 queries) without query rewriting. ELSER v1 substantially outperforms all alternatives (Table 14), consistent with MTRAG's construction process in which annotators primarily reviewed ELSER-retrieved passages, creating a favorable bias toward ELSER in the relevance annotations (Katsis et al., 2025). The gap is especially pronounced relative to BM25 (+33 R@5 points) and remains meaningful even against the strongest dense competitor, Cohere Embed v4 (+8.6 points). All improvements over BM25 and over dense retrievers are statistically significant under paired bootstrap resampling with 10,000 iterations ($p < 0.01$).

Type	Retriever	R@5	nDCG@5
Lexical	BM25	0.153	0.130
Dense	Cohere Embed v4	0.397	0.362
	AWS Titan v1	0.352	0.328
	Voyage-3.5-L	0.321	0.302
	GTE-Large	0.326	0.293
	BGE-Large-v1.5	0.282	0.251
Sparse	ELSER v1	**0.483**†	**0.444**†
	ELSER v2	0.411	0.380
	SPLADE v3	0.402	0.374

Table 14: Retriever comparison (777 queries, no rewriting). ELSER v1 exceeds Cohere Embed v4 by +8.6 R@5. †Significant vs. all others (p<0.01, paired bootstrap).

Task	Metric	Ours	Rank	Top Score	Top Baseline	Baseline Score
A — Retrieval	nDCG@5	**0.5776**	1/38	0.5776	ELSER + GPT-OSS-20b rewrite	0.4795
B — Generation	HM	**0.7698**	2/26	0.7827	GPT-OSS-120b	0.6390
	RB_{alg} / RL_F / RB_{llm}	0.633 / 0.897 / 0.832	–	–	–	–
C — RAG	HM	0.5409	11/29	0.5861	Qwen-30B-A3B-Thinking	0.5366
	RB_{alg} / RL_F / RB_{llm}	0.400 / 0.729 / 0.598	–	–	–	–

Table 13: Official test set results. HM = harmonic mean of RB_{alg}, RL_F, RB_{llm}. Top Baseline refers to the strongest non-participant system released by the organizers. Task C rank reflects the answerability bottleneck identified .

Optimized individual retrievers. To ensure that ELSER's advantage is not an artifact of pipeline co-optimization, we also optimized each retriever independently using the same downstream pipeline components (multi-strategy rewriting and reranking), and then compared the resulting recall at multiple cutoffs (Table 15). Even after optimization, ELSER maintains the best R@5/R@10, confirming that its advantage is intrinsic rather than an artifact of score-scale compatibility with downstream fusion.

Retriever	R@5	R@10	R@50	R@100
ELSER v1	**0.607**	**0.729**	**0.873**	**0.901**
SPLADE	0.539	0.657	0.836	0.874
Cohere	0.503	0.657	0.823	0.854
BM25	0.444	0.536	0.688	0.735

Table 15: Per-retriever recall after the same optimization pipeline. ELSER dominates at R@5/R@10.

B.2 Query Rewriting: Strategy Performance

Table 16 evaluates rewriting strategies in isolation. A central finding is that query rewriting is *not* universally beneficial: poorly calibrated reformulations—such as those produced by FlanT5, which over-generalizes the query—*degrade* retrieval below the unaugmented baseline. This confirms that rewriting must be both semantically faithful and corpus-aware to be effective; naive rewriting can introduce more noise than it resolves. For this reason, smaller rewriting models (FlanT5) and classical PRF were ruled out early in development: preliminary runs indicated that strong instruction-following capability is a prerequisite for reformulation quality in this setting, after which exploration focused exclusively on prompt-level variants over a fixed DeepSeek-V3.2 backbone.

Among LLM-based strategies, Corpus-Specific yields the best single-strategy performance (R@5: 0.541), but the final system design is driven by *strategy complementarity*: each reformulation targets distinct failure modes, and their combination under nested RRF (R@5: 0.607) outperforms any individual strategy by at least 6.6 points. Decomposition and Question Type were excluded from the final pipeline: Decomposition exhibited high reformulation overlap with CoT rewrites, while Question Type showed inconsistent gains across domains. All LLM-based strategy gains over the original query are statistically significant ($p < 0.01$, paired bootstrap resampling).

Category	Strategy	R@5	nDCG@5
Baselines	Original query	0.483	0.444
	FlanT5	0.463	0.426
	Annotator prompt*	0.528	0.493
LLM-based (DeepSeek-V3.2)	Minimal	0.527	0.487
	Corpus-Specific	**0.541**†	**0.498**†
	Chain-of-Thought	0.521	0.482
	HyDE	0.485	0.445
	Decomposition	0.539	0.478
	Anchor-Keyword	0.501	0.460
	Question Type	0.513	0.470

Table 16: Rewriting strategy comparison on the dev set. *Annotator prompt replicates the query rewriting prompt used during MTRAG data collection (Katsis et al., 2025) (see Appendix C.1 therein). Non-LLM methods (FlanT5) fall below the unaugmented baseline, confirming that rewriting *hurts* when not properly grounded. Among LLM-based variants, Corpus-Specific achieves the best single-strategy R@5 (†); the final pipeline uses five complementary strategies (Minimal, Corpus-Specific, CoT, HyDE, Anchor-Keyword). All LLM gains over the original query are significant (p<0.01, paired bootstrap).

B.3 Query Rewriting Examples

Table 17 shows representative outputs for a context-dependent Cloud query.

Strategy	Example Output
Original	"What about the pricing?"
Minimal	"What is the pricing for IBM Cloud Object Storage?"
Corpus-Spec	"IBM Cloud Object Storage pricing tiers and cost structure"
HyDE	"IBM Cloud Object Storage offers flexible pricing based on storage class, including Standard, Vault, and Cold Vault tiers. . . "
CoT	"Need: storage costs, transfer fees, API pricing → IBM Cloud Object Storage full pricing breakdown"
Anchor-KW	"IBM Cloud Object Storage pricing cost GB month tier"

Table 17: Reformulations for a non-standalone Cloud query. "The pricing" refers to an entity introduced earlier in the conversation.

B.4 Conversation History Ablation

We vary the amount of conversation context provided to the rewriter in Table 18. Including history is essential; performance saturates after 4–6 user turns, and adding assistant turns has negligible aggregate impact. We retain 3 assistant turns as a conservative default to support coreference resolution in edge cases (e.g., when the user references an entity introduced only in an assistant response), despite the marginal aggregate effect.

Configuration	User	Asst	R@5
No history	0	0	0.490
Full history	all	all	0.531
2 user turns	2	0	0.506
4 user turns	4	0	0.528
6 user turns	6	0	0.530
8 user turns	8	0	0.528
6u + 1 asst	6	1	0.528
6u + 3 asst	6	3	0.533
6u + 5 asst	6	5	0.532

Table 18: History ablation (Minimal rewriting). Performance saturates at 4–6 user turns; assistant turns add negligible benefit.

B.5 Rewriting Model Selection

Table 19 compares rewriting models across strategies. The results reveal a nuanced pattern: GPT-4o's advantage over DeepSeek-V3.2 is *not* uniform across reformulation types. For strategies that require strict instruction following and structured output—Minimal and Corpus-Specific—the two models perform nearly identically, confirming that DeepSeek-V3.2's instruction-following capability is sufficient for this task. GPT-4o's marginal gains are concentrated on *open-ended* strategies such as HyDE, where the task is less about following a precise rewriting schema and more about generating a plausible hypothetical passage from open-ended context—a regime that favors GPT-4o's stronger generative fluency. However, these gains are small in absolute terms and come at a $15\times$ cost premium per query. Since HyDE is only one of five complementary strategies and its contribution to the final nested RRF score is down-weighted (Table 24), the quality delta does not justify the additional expense. Notably, GPT-4o with Minimal rewriting matches DeepSeek-V3.2 with Corpus-Specific (both R@5 = 0.541), suggesting that model capability can partially substitute for prompt specialization. We therefore select DeepSeek-V3.2 for all rewriting strategies in the final system.

Strategy	Model	R@5	nDCG@5	Cost/Q
Minimal	DeepSeek-V3.2	0.527	0.487	$0.001
	GPT-4o-mini	0.513	0.481	$0.003
	GPT-4o	**0.541**	**0.501**	$0.015
HyDE	DeepSeek-V3.2	0.485	0.445	$0.001
	GPT-4o-mini	0.503	0.462	$0.003
	GPT-4o	**0.504**	**0.464**	$0.015
Corpus-Spec	DeepSeek-V3.2	0.541	0.498	$0.002
	GPT-4o-mini	0.517	0.479	$0.005
	GPT-4o	**0.542**	**0.503**	$0.020

Table 19: Rewriting model comparison across strategy types. GPT-4o gains are marginal on instruction-following strategies (Minimal, Corpus-Specific) and only slightly larger on the open-ended HyDE strategy, at $15\times$ higher cost. DeepSeek-V3.2 offers the best cost–quality trade-off across all strategies.

B.6 Passage-Informed Rewriting (PIR)

A recurring challenge in multi-turn retrieval is that query reformulation operates *blind*: the rewriter has no access to what the retriever actually found, and cannot adapt its output to fill evident coverage gaps. This observation motivated an approach we term *Passage-Informed Rewriting* (PIR): rather than rewriting the query in isolation, PIR feeds the initially retrieved passages back to the LLM as semantic context for a *second* reformulation step, effectively creating a closed loop between retrieval signal and query adaptation. Unlike classical Pseudo-Relevance Feedback (PRF), which relies on term co-occurrence statistics, PIR leverages LLM semantic understanding to interpret passage content. To our knowledge, this formulation has not been previously evaluated in a conversational multi-turn RAG setting.

The motivation was two-fold. First, we hypothesized that PIR could improve *ranking quality*: if the rewriter observes which passages were retrieved, it can sharpen the query toward the terminology and structure of the most promising candidates, nudging the retriever to surface those documents at higher ranks. Second, we hoped PIR would improve *coverage*: by identifying what the first-stage results clearly *lack*, the second-stage query could be expanded to surface complementary evidence—a property particularly valuable in multi-turn conversations where information needs span multiple retrieval rounds.

Table 20 reports the results. Classical PRF collapses severely (R@5: 0.182–0.410), confirming that term-extraction statistics induce severe query drift in conversational settings. PIR fares substantially better and exhibits a clear and interpretable pattern: its performance tracks the quality of the initial retrieval signal almost linearly. When seeded with a minimal rewrite and cross-encoder reranking, PIR reaches R@5 = 0.541—matching our best single-strategy baseline. Yet this is precisely where the hypothesis breaks down: despite a more informed second-stage query, PIR *never exceeds* the baseline it is seeded from.

We attribute this to a diminishing-returns effect that is structurally tied to our retrieval setup. Once first-stage R@100 approaches saturation (>0.90 in our setup; see Table 15), the gold passages are already present in the candidate pool; the bottleneck shifts entirely to top-k *ranking*, not coverage. PIR addresses coverage by broadening the query, but broadening a query that already has near-complete coverage introduces fusion noise rather than new signal. In other words, PIR solves a problem—coverage gaps—that our pipeline has already largely eliminated. We therefore exclude PIR from the final submission, while noting that it may prove effective in lower-recall settings or corpora where first-stage coverage is a genuine bottleneck.

Category	Method	R@5	nDCG@5
Baseline	Minimal only	0.527	0.487
	Minimal + Rerank	0.594	0.508
Classical PRF	BM25 TF-IDF	0.182	0.155
	ELSER TF-IDF	0.410	0.387
PIR (init. qual.)	+ original query	0.504	0.472
	+ minimal rewrite	0.520	0.485
	+ min. + rerank	0.541	0.503
PIR (variant)	PIR-HyDE	0.541	0.503
	PIR-CoT	0.520	0.483
	PIR-Dense Summary	0.516	0.478
	PIR-Corpus-Spec	0.497	0.458

Table 20: Classical PRF vs. PIR on the dev set. PRF collapses due to query drift; PIR tracks the quality of its seed signal but does not exceed it, suggesting that gains are bounded by the coverage already achieved in the first retrieval stage.

Qualitatively, PIR offers the most promise when initial passages contain strong structural cues—headings, canonical terminology, entity definitions—that the rewriter can leverage to sharpen the query. When early retrieved passages are noisy or off-topic, PIR amplifies that noise rather than correcting it, underscoring that its effectiveness is contingent on a sufficiently reliable first-stage retrieval signal.

B.7 Multi-Retriever Ensemble Paradox

Table 21 documents a consistent pattern: as ELSER improves through rewriting, adding additional retrievers increasingly harms R@5 even while improving R@100. Alternative retrievers contribute unique gold documents at the final stage, but these appear at avg. rank 37–54 after fusion—below the top-10 cutoff—while fusion displaces borderline ELSER hits. This motivates a single-retriever architecture with multi-query diversity. The final-stage degradation (ELSER alone: 0.607 vs. ELSER+All: 0.569) is statistically significant ($p < 0.01$, paired bootstrap resampling).

Configuration	R@5	R@100	Uniq. golds	Avg rank
Early Stage — No Rewriting				
ELSER v1 alone	0.497	0.779	–	–
+Cohere	0.551↑	0.906↑	293	19.8
+SPLADE	0.559↑	0.895↑	288	22.2
+BM25	0.498↑	0.859↑	222	25.6
+All	0.568↑	0.920↑	803	22.5
Mid Stage — Basic Rewriting				
ELSER v1 alone	0.531	0.877	–	–
+Cohere	0.578↑	0.910↑	89	29.8
+SPLADE	0.574↑	0.900↑	63	36.9
+BM25	0.536↑	0.895↑	89	23.0
+All	0.569↑	0.918↑	241	29.9
Final Stage — Optimized 5-Rewrite				
ELSER v1 alone	**0.607**†	0.901	–	–
+Cohere	0.582↓	0.925↑	50	37.0
+SPLADE	0.592↓	0.919↑	32	54.2
+BM25	0.536↓	0.915↑	59	27.5
+All	0.569↓	0.926↑	141	39.5

Table 21: The ensemble paradox. As ELSER improves through rewriting, multi-retriever fusion increasingly harms R@5 while improving R@100. ↑/↓: change relative to ELSER alone at the same stage. Unique gold documents from other retrievers shift to avg. rank 37–54 after fusion—below the top-10 cutoff. †ELSER alone vs. +All: $p<0.01$ (paired bootstrap).

B.8 Reranking Experiments

Cross-encoder rerankers as replacement. Table 22 shows that using a reranker as the sole ranking signal degrades performance despite strong results on standard benchmarks. Weighted RRF fusion successfully combines ELSER's calibration with reranker semantic signals.

Configuration	R@5	Δ
ELSER v1 (baseline)	0.527	–
Cohere Rerank v4 (sole)	0.481	−8.7%
Voyage-Reranker-2.5 (sole)	0.469	−11.0%
CrossEncoder MS-MARCO (sole)	0.421	−20.0%
ELSER + Cohere RRF (α=0.5)	**0.594**	+12.7%

Table 22: Reranking experiments. Sole reranking degrades performance; weighted RRF fusion yields complementary gains.

LLM-based reranking (negative results). Having established that cross-encoder reranking complements ELSER via weighted RRF, we explored whether an LLM judge could serve as a stronger reranker by leveraging deeper semantic reasoning. LLM-based reranking was our last resort: we turned to it only after confirming that no combination of retrievers or cross-encoders could push R@5 further. The core appeal was that an LLM, conditioned on the full query context and conversation history, might resolve fine-grained relevance distinctions that embedding-based models miss.

Three formulations were evaluated, all operating on the same top-20 candidate pool. The pool size was a deliberate constraint: expanding beyond 20 passages would have required either splitting the list into chunks—defeating the purpose of holistic comparison—or providing the full list in a single prompt, which at 20×512-token passages already strains the effective context window and triggers the well-documented lost-in-the-middle effect, where passages in the middle of a long prompt receive disproportionately less attention regardless of their relevance. Preliminary runs confirmed this: ranking quality with top-5 input was nearly identical to top-20, suggesting the LLM could not reliably distinguish relevance gradients even within a short list.

Table 23 reports the results. No LLM formulation improved over weighted RRF, while cost increased by 20–200× and latency by 5–20×. The generation-based variant—which asks the LLM to re-generate the ideal answer and rank passages by proximity to that answer—performs worst, likely because it conflates generation quality with retrieval relevance. We attribute the overall failure to the same saturation dynamic observed with PIR: once weighted RRF achieves high top-20 recall, the marginal relevance differences between candidates are too subtle for a general-purpose LLM judge to resolve reliably without task-specific fine-tuning.

Method	R@10	Cost/Q	Latency
Weighted RRF (baseline)	**0.725**	\$0.001	1×
Pointwise LLM (1–10)	0.701	\$0.050	10×
Listwise LLM reorder	0.688	\$0.020	5×
Generation-based	0.618	\$0.200	20×

Table 23: LLM reranking on top-20 candidates. All three formulations degrade R@10 relative to weighted RRF, at 20–200× higher cost and 5–20× higher latency. The listwise formulation requires all candidates in a single prompt to enable holistic comparison; splitting into chunks eliminates the main theoretical advantage of LLM-based reranking and was therefore not evaluated.

B.9 Nested RRF Parameters

Flat RRF treats all strategies as peers, assigning equal fusion weight to every ranked list. This is problematic in our setting because the five rewriting strategies are *not* equally reliable: Minimal and Corpus-Specific produce stable, high-precision rankings with low turn-to-turn variance, while HyDE, CoT, and Anchor-Keyword exhibit higher variance but contribute complementary coverage. Assigning them equal weight allows the high-variance group to inject noise into top ranks precisely on turns where they produce poor reformulations—offsetting the precision of the stable strategies.

This observation directly motivated the nested RRF design: rather than down-weighting individual strategies by hand, we pre-aggregate the three high-variance strategies into a single *Weak Consensus* ranking (Level 1), which smooths their individual noise before they compete with the stable strategies at Level 2. The result is that the high-variance group collectively contributes one vote—rather than three—in the final fusion, naturally reducing their influence without discarding their coverage gains. Empirically, nested RRF outperforms flat RRF by 1.8 R@5 points on average across the dev set, with the largest gain on FiQA (+3.2 points), where high-variance strategies are most prone to hallucinated reformulations.

The corpus-specific weights in Table 24 were tuned by grid search on the development set and reflect genuine corpus structure: formal domains

(Govt, Cloud) assign higher weight to Minimal (w=0.65), which preserves domain terminology, while FiQA assigns comparatively more weight to the Weak Consensus group (w=0.15), where coverage diversity matters more given the corpus's informal, high-variance language. That said, the tuned weights are *improvements*, not prerequisites: uniform weights across all corpora yield less than 1% R@5 degradation, confirming that the nested structure itself—not the specific weight values—is the operative design choice.

Parameter	ClapNQ	FiQA	Govt	Cloud
k_{final}	20	60	40	20
w_{Minimal}	0.55	0.45	0.65	0.65
$w_{\text{Corpus-Spec}}$	0.40	0.40	0.25	0.30
w_{WeakCons}	0.05	0.15	0.10	0.05

Table 24: Corpus-specific nested RRF parameters (dev set). Formal corpora (Govt, Cloud) weight stable strategies higher; FiQA assigns more weight to the Weak Consensus group. Uniform weights across corpora degrade R@5 by less than 1%, confirming robustness to weight selection.

B.10 Per-Domain Retrieval Analysis

Table 25 reports final performance per domain alongside absolute gains over the unaugmented ELSER baseline. The per-domain pattern is consistent with the corpus characteristics described in Table 1: ClapNQ and Govt, which feature formal encyclopedic and governmental language, benefit most from Minimal rewriting, which preserves domain-specific lexical anchors while resolving conversational coreferences. Cloud documentation follows the same pattern. FiQA is the hardest corpus throughout: its informal, forum-style language creates a large vocabulary mismatch between user queries and document terminology, making it the only domain where Corpus-Specific rewriting outperforms Minimal.

Table 26 breaks down R@5 per strategy per domain, directly motivating the corpus-specific weights in Table 24. Two patterns are notable. First, HyDE and CoT show the highest cross-domain *variance*: they provide large gains on FiQA (where bridging the vocabulary gap matters most) but are near-neutral or slightly negative on ClapNQ (where the precise Wikipedia terminology of the original query is often optimal). This is precisely why these strategies are grouped into the Weak Consensus tier—their per-domain reliability is too inconsistent to be trusted with high fusion weight globally. Second, Corpus-Specific is the only strategy that improves on *all four* domains, confirming its role as the most broadly applicable single rewriting strategy.

Note that the "Top Strategy" column in Table 25 refers to the strategy receiving the highest fusion weight in the final nested RRF configuration, not the best single-strategy R@5 from Table 26.

Domain	Baseline	Final	Δ	Top Strategy
ClapNQ	0.528	0.690	+16.2	Minimal
FiQA	0.419	0.527	+10.8	Corpus-Spec
Govt	0.507	0.629	+12.2	Minimal
Cloud	0.477	0.541	+6.4	Minimal
Overall	0.483	0.597	+12.4	–

Table 25: Per-domain R@5 on the dev set: unaugmented ELSER baseline vs. final system. Δ is the absolute R@5 gain. FiQA shows the smallest gain despite benefiting most from Corpus-Specific rewriting, reflecting its structurally harder retrieval problem.

Domain	Minimal	Corpus-Sp.	CoT	HyDE	Anchor-KW
ClapNQ	0.622	**0.637**	0.575	0.566	0.598
FiQA	0.466	**0.491**	0.475	0.392	0.419
Govt	0.561	0.572	**0.580**	0.550	0.564
Cloud	**0.463**	0.462	0.457	0.431	0.429

Table 26: R@5 per strategy per domain (single-strategy, no fusion, no reranking).

B.11 Per-Turn Retrieval Performance

Figure 8 reports R@5 as a function of turn index for both the unaugmented ELSER baseline and the final optimized system, broken down by domain. Without rewriting, retrieval quality degrades sharply from turn 1 (R@5 = 0.879) to later turns (R@5 = 0.428), a 51% relative decrease. Multi-strategy rewriting reduces this to 39% (Turn 1: 0.890, Turn 6+: 0.540).

B.12 Standalone vs. Non-Standalone Performance

Table 27 and Figure 9 decompose R@5 by query type. The unaugmented baseline achieves R@5 = 0.879 on standalone queries but drops to 0.440 on non-standalone (gap: 43.9 pp). Rewriting disproportionately benefits non-standalone queries (+12.4 pp overall) vs. standalone (+1.1 pp), confirming history-aware rewriting as the operative mechanism.

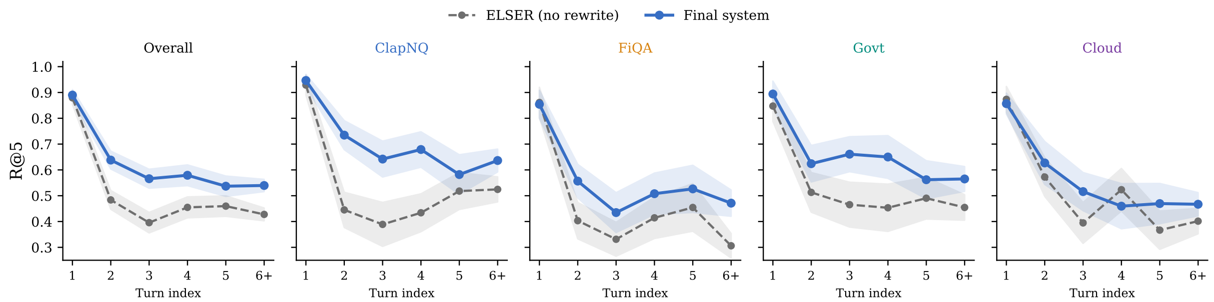

Figure 8: R@5 vs. turn index: ELSER baseline (dashed) vs. final system (solid). Rewriting reduces the later-turn drop from 51% to 39%.

	ELSER (no rw.)		Final system	
Domain	**SA**	**Non-SA**	**SA**	**Non-SA**
Overall	0.879	0.440	0.890	0.564
ClapNQ	0.929	0.477	0.946	0.651
FiQA	0.861	0.363	0.854	0.493
Govt	0.847	0.470	0.894	0.600
Cloud	0.873	0.440	0.857	0.499

Table 27: R@5 for standalone (SA) vs. non-standalone (Non-SA) queries. Rewriting benefits Non-SA queries by +12.4 pp vs. +1.1 pp for SA.

B.13 Query Rewriting Prompt Templates

All five rewriting strategies receive the current-turn query and a truncated conversation history formatted via XML structural tags. History windows are capped at 6 user turns; the number of assistant turns varies by strategy (Table 28). All strategies use DeepSeek-V3.2 with τ=0.0 and return structured JSON containing a standalone classification and the rewritten query.

Table 28: Rewriting strategy configurations.

Strategy	Asst.	Key Behavior
Minimal	3	Coreference only
Corpus-Spec.	0–3	Domain-aware (Tab. 29)
CoT	3	Reasoning trace
HyDE	3	Hypothetical passage
Anchor-KW	1	Entity + intent extraction

We show one representative prompt per structural category. The remaining Corpus-Specific variants differ only in domain rules per Table 29.

Minimal rewriting.

```
System:
Rewrite the final utterance into a single
standalone utterance without needing history.

Rules:
- Do not rephrase or introduce new terms.
- Stay close to the original.
- If standalone: return THE SAME query.
- Use assistant turns ONLY to resolve
  pronouns/references.

JSON: {"class": "standalone|non-standalone",
       "rewritten version": "..."}

{history}
user: {question}
```

Table 29: Domain-specific preservation rules for Corpus-Specific rewriting. All four variants share the same template; only the preservation targets differ.

Domain	Asst.	Preservation Targets
ClapNQ	3	Entity variants, Wikipedia terminology, temporal phrasing
FiQA	0	Amounts, currencies, tickers, time horizons
Govt	3	Program names, agency acronyms, form IDs, locations
Cloud	3	Error codes, CLI commands/flags, config keys, service names

Corpus-Specific (ClapNQ variant shown).

```
System:
You are rewriting queries for retrieval from
Wikipedia using ELSER (sparse semantic search).

CRITICAL RULES:
1. ENTITY FORMS: include formal AND common
   names (e.g., "Apple Inc. Apple")
2. PRONOUNS: resolve to exact entity
3. TEMPORAL: Wikipedia's natural phrasing
4. FORMALIZATION: preserve conversational terms
5. DISAMBIGUATION: qualifier only when ambiguous
6. KEYWORDS: preserve who/what/when/where

If standalone, return UNCHANGED.

JSON: {"class": "standalone|non-standalone",
       "rewritten version": "..."}

{history}
user: {question}
```

Chain-of-Thought (CoT) rewriting. The `reasoning` field is discarded at inference; only `rewritten version` is passed to the retriever.

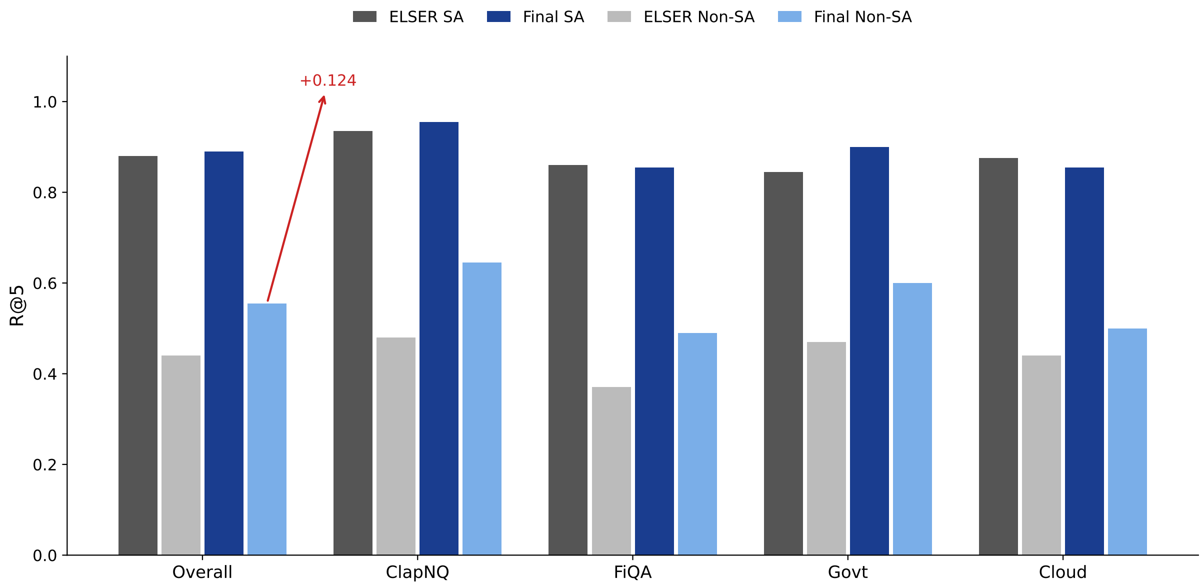

Figure 9: SA vs. Non-SA R@5, baseline vs. final system, per domain.

```
System:
Expert query rewriter for IR systems.

PROCESS (Chain-of-Thought):
1. ANALYZE: entities, pronouns, ambiguity
2. REASON: standalone? what history needed?
3. REWRITE: resolve, add context, keep wording

JSON: {"reasoning": "<step-by-step>",
       "class": "standalone|non-standalone",
       "rewritten version": "<query>"}

{history}
user: {question}
```

HyDE rewriting. Generates a hypothetical 2–4 sentence answer passage, concatenated with the standalone rewrite. The combined string is submitted to ELSER.

```
System:
Generate hypothetical document passages for
retrieval (HyDE strategy).

TASK: Generate a 2-4 sentence passage that would
answer the query. Used for retrieval only.

JSON: {"standalone_query": "<rewrite>",
       "hypothetical_passage": "<2-4 sent>"}

Rules:
- Write as a document answering the question
- Include facts, names, dates when relevant
- Use vocabulary from authoritative sources

{history}
user: {question}
```

Anchor-Keyword rewriting. The final query submitted to ELSER is `rewritten_version` + `anchors` + `keywords`, capped at 28 words.

```
System:
Rewrite into ONE standalone query for RETRIEVAL.
Extract RETRIEVAL TERMS for ELSER:
- anchors: entity names, acronyms, IDs,
  error codes, CLI flags (max 8)
- keywords: intent terms (max 12)

Rules:
- Do NOT invent new entities/facts
- Preserve numbers/codes/tickers exactly
- Query <= 28 words

JSON: {"class": "standalone|non-standalone",
       "rewritten version": "...",
       "anchors": ["..."],
       "keywords": ["..."]}

{history}
user: {question}
```

C Task B: Generation Details

C.1 Prompt Templates

We provide the prompt templates used in each pipeline stage, as deployed in our final test-set submission. All prompts receive the current-turn question, a truncated history window (up to 4 recent turns formatted as `User: ...` / `Assistant: ...` pairs), and either the raw retrieved passages (Stage 1) or extracted evidence spans (Stages 2–4). Each prompt also specifies a *system message* (shown in parentheses) that sets the model's persona.

Conversational response (Stage 0, GPT-4o-mini, τ=0.3). Triggered when the user turn matches a short conversational pattern (e.g., "thanks", "ok", "got it") detected via regex.

```
System: Friendly assistant.
Brief friendly response to conversational message.
History: {history}
User: {question}
Response (under 50 words):
```

Unanswerable / no-context response (Stage 0, GPT-4o-mini, τ=0.0). Triggered when the retrieval pipeline returns zero passages for the current turn.

```
System: Helpful assistant.
No information available for this question.
Question: {question}
Short response (under 25 words) stating information is not available.
Do NOT say "I don't know" - say "The information is not available" or similar.
Response:
```

Span extraction (Stage 1, DeepSeek-V3, τ=0.0). Extracts up to K=8 verbatim sentences from the top-N=5 retrieved passages. If the first attempt returns fewer than 1 span, a retry is issued with an urgency prefix (`"Answer MUST exist - find it!"`).

```
System: Extract relevant sentences.
Extract sentences answering the question.
{urgency_if_retry}
History: {history}
Question: {question}
PASSAGE 1:
{passage_1_text}
. . .
PASSAGE N:
{passage_N_text}
Rules:
1. Copy EXACT sentences
2. Include: names, numbers, dates, key facts
3. Max 8 sentences
4. Prioritize earlier passages
JSON format:
{"extractedSpans": [{"passageId": 1, "sentence": "exact text"}]}
```

Generation (Stage 2, GPT-4o, $\tau \in \{0.0, 0.1\}$). Called twice per turn to produce two candidate answers: a greedy candidate (τ=0.0, higher faithfulness) and a stochastic candidate (τ=0.1, often more natural phrasing). The `qtype` and `style_hint` fields are set by the rule-based question-type classifier (Table 32); `target_words` is the base target (90 words) plus a per-type offset.

```
System: Helpful, accurate assistant.
Generate a natural answer using ONLY the facts below.
FACTS:
{extracted_spans_as_bullet_list}
Conversation context: {history}
Question: {question}
Type: {qtype} - {style_hint}
CRITICAL RULES:
- Use ONLY information from FACTS above
- Copy exact phrases for: names, numbers, dates, technical terms
- Aim for ~35% verbatim overlap with facts
- Make it sound natural and complete
- NO outside knowledge
- NO hedging (seems, possibly, maybe)
- NO meta-phrases (based on, according to)
Length: ~{target_words} words
Answer:
```

The explicit instruction to "aim for $\sim$35% verbatim overlap" was added after observing that unconstrained GPT-4o tends to over-paraphrase evidence, reducing extractiveness below the target band and harming faithfulness scores.

Technical judge (Stage 3, DeepSeek-V3, τ=0.0). Receives both candidates with their word counts and 4-gram extractiveness percentages precomputed. Evidence spans are truncated to 120 characters each to fit within the token budget.

```
System: Judge assistant.
Compare two answers for quality.
Question: {question}
Facts: {truncated_spans}
A: {answer_A}
  ({wc_A}w, {extr_A}% extractiveness)
B: {answer_B}
  ({wc_B}w, {extr_B}% extractiveness)
Evaluate: Faithfulness, Completeness, Naturalness
Ideal extractiveness: 28-45%
JSON: {"winner": "A|B", "score_A": 0-10, "score_B": 0-10, "reason": "brief"}
```

User satisfaction judge (Stage 3, GPT-4o-mini, τ=0.0, 60% sampled). Invoked on a fixed 60% of turns (controlled by a random draw with `seed=42`) to provide a complementary user-preference signal. The 60% rate was chosen to reduce variance in stylistic selection while keeping latency and cost bounded.

System: User perspective.

```
As a user, which answer do you prefer?

Context: {recent_history}
Question: {question}

A: {answer_A}
B: {answer_B}

JSON: {"preferred": "A|B", "confidence":
"HIGH|MEDIUM|LOW"}
```

Force-answer fallback (GPT-4o-mini, τ=0.2). When both candidates are classified as "I don't know" responses (via pattern matching on refusal phrases) but contexts *do* exist, the pipeline forces a third generation attempt using a simplified prompt that explicitly prohibits refusal. This fallback prevents the system from incorrectly refusing to answer when evidence is available.

System: Always answers when facts exist.

```
Answer using ONLY these facts.  Do NOT
say you cannot answer.

Facts:
{extracted_spans_as_bullet_list}

Question: {question}

Direct answer using the facts:
```

Micro-adjustment (Stage 4, GPT-4o-mini, τ=0.1). Applied only when the selected answer violates one of three constraints: (i) too short ($<$50 words), (ii) too long ($>$150 words), or (iii) low extractiveness ($r_4 < 0.28$). The reason field is populated dynamically based on the specific violation detected.

System: Editor.

```
Fix this answer: {reason}

Facts:
{truncated_spans}

Current: {answer}

Fix with minimal changes.  Use exact
phrases from facts.

Fixed:
```

C.2 Model Routing and Cost

Table 30 summarizes the model assignment and estimated per-call cost for each pipeline stage. The routing principle is *capability-aligned assignment*: each stage is assigned to the model best suited to its specific demands, with cost efficiency as a secondary criterion. Concretely, GPT-4o is reserved exclusively for the generation stage, where fluency, instruction grounding, and long-form coherence are most critical. DeepSeek-V3.2 handles span extraction and technical judging—tasks that require *strict instruction following*, deterministic JSON output formatting, and precise boundary detection; empirically, DeepSeek-V3.2 outperforms GPT-4o-mini on these structured tasks despite comparable cost. GPT-4o-mini handles all remaining lightweight tasks (conversational triage, unanswerable detection, user judging, force-answer fallback, and micro-edits), where response brevity and low latency matter more than generation depth. All models are accessed via Azure OpenAI / Azure AI Foundry endpoints with exponential backoff retry logic (up to 5 attempts, jittered).

Stages 2a and 2b execute concurrently, limiting the per-turn latency overhead of dual candidate generation to approximately that of a single API call. At nominal API pricing, the routed configuration costs approximately \$0.008 per turn on average (dev set), compared to \$0.024 for uniform GPT-4o assignment—a 3$\times$ cost reduction with less than 1 HM point degradation (Table 33).

Stage	Role	Model	Est. Cost/call
0b	Unanswerable triage	GPT-4o-mini	\$0.0001
1	Span extraction (×1–2)	DeepSeek-V3.2	\$0.0008
2a	Generation (greedy)	GPT-4o	\$0.0045
2b	Generation (stochastic)	GPT-4o	\$0.0045
–	Force-answer fallback	GPT-4o-mini	\$0.0003
3a	Technical judge	DeepSeek-V3.2	\$0.0003
3b	User satisfaction judge	GPT-4o-mini	\$0.0001
4	Micro-adjustment	GPT-4o-mini	\$0.0002
	Total (avg/turn)		**\$0.0080**

Table 30: Model routing for Task B with estimated per-call API cost at nominal pricing. GPT-4o is used only for answer generation (Stages 2a–2b); DeepSeek-V3.2 handles all instruction-following and structured output tasks; GPT-4o-mini covers all lightweight conditional stages. The total average cost per turn (\$0.008) is 3× lower than uniform GPT-4o assignment (\$0.024). Costs are estimated at nominal API pricing as of submission; actual per-call cost varies with input/output token counts (std = ±\$0.002 per turn on the dev set) and is subject to provider pricing changes.

C.3 Selection Score Details

Given two candidates A (τ=0.0) and B (τ=0.1), we select the final answer via a composite score combining four signals: technical quality, user preference, extractiveness calibration, and forbidden-phrase penalization.

4-gram extractiveness. We measure how much of the candidate answer y is grounded in the extracted evidence spans S via 4-gram overlap:

$$r_4(y, S) = \frac{|4\text{grams}(y) \cap 4\text{grams}(\text{concat}(S))|}{|4\text{grams}(y)|}, \tag{4}$$

where $\text{concat}(S)$ is the concatenation of all extracted spans. The target band (28–38% ideal, up to 50% acceptable) was calibrated on dev set reference answers, which exhibit a mean 4-gram overlap of 36.2%. This metric operationalizes the faithfulness–naturalness trade-off formally studied in abstractive summarization: answers below the lower bound risk hallucination by straying too far from grounded evidence, while answers above the upper bound sacrifice naturalness through mechanical repetition.

Extractiveness shaping function. A piecewise shaping term $\phi(r_4)$ discourages both hallucination-prone outputs (too low overlap) and robotic verbatim copying (too high):

$$\phi(r_4) = \begin{cases} -1.5 & \text{if } r_4 < 0.28 \quad \text{(under-extractive)} \\ +2.5 & \text{if } 0.28 \leq r_4 \leq 0.38 \quad \text{(ideal)} \\ +1.5 & \text{if } 0.38 < r_4 \leq 0.50 \quad \text{(acceptable)} \\ +0.5 & \text{if } r_4 > 0.50 \quad \text{(over-extractive)} \end{cases} \tag{5}$$

The function is deliberately asymmetric: the penalty for under-extractive responses (-1.5) is larger in magnitude than the reward reduction for over-extractive ones ($+0.5$), reflecting the empirical finding that hallucination degrades scores more severely than mild verbatim copying. The discontinuity at r_4=0.28 creates a strong grounding incentive; empirically, this threshold corresponds to approximately one verbatim fact per two sentences. Figure 10 visualizes the full step function.

User-preference term. When the user-satisfaction judge is invoked (60% of turns after ablaton study), its confidence level maps to a weight $c \in \{1.0, 0.7, 0.4\}$ for {HIGH, MEDIUM, LOW} confidence respectively. A signed preference term $\pm 5c$ is added to the preferred candidate's score. When the judge is *not* invoked, a constant prior of $+2.0$ favors Candidate A (the greedy candidate), which is typically more faithful.

Full selection score. Let $T(y) \in [0, 10]$ be the technical judge score and $U(y) \in \{-1, 0, +1\}$ the signed user preference indicator. The composite score is:

$$\begin{aligned} \text{Score}(y) &= 0.35\, T(y) + 5c\, U(y) \\ &\quad + \phi(r_4(y, S)) - 2.0 \cdot \mathbb{I}[\text{forbidden}(y)] \end{aligned} \tag{6}$$

where $\mathbb{I}[\text{forbidden}(y)]$ flags residual hedging or refusal patterns (App. C.4). The candidate with the higher score is selected; ties favor A. The coefficient 0.35 on the technical score was chosen so that $T(y)$ can contribute at most 3.5 points—comparable in scale to the extractiveness shaping term—preventing either signal from dominating selection.

Micro-adjustment policy. After selection, we apply a GPT-4o-mini editing step *only* when the chosen answer violates one of three constraints: (i) too short (<50 words), (ii) too long (>150 words), or (iii) low extractiveness ($r_4 < 0.28$). The editor prompt specifies the violation and is constrained to "minimal changes" grounded in the extracted spans. If the edited output fails validation (is a refusal or <30 words), the original answer is retained unchanged.

C.4 Forbidden-Phrase Filtering

As a final post-processing step, we remove a fixed set of hedging and refusal phrases that consistently degraded naturalness and appropriateness scores on answerable turns during development:

```
"I don't know", "I do not know", "I'm not
sure", "I am not sure", "I'm uncertain",
"I cannot say", "It's unclear", "It is
unclear", "I cannot answer", "Unable to
answer", "Cannot find information"
```

Removal is applied via case-insensitive string matching on the final output only; the generation and judging stages operate on unfiltered text. Whitespace artifacts left by removal are cleaned (multiple spaces collapsed, orphaned punctuation corrected).

Table 31 reports the per-phrase fire rate on the dev set, confirming that the pattern is not uniformly distributed: "I'm not sure" and its variants account for 61% of all activations, primarily on partially answerable turns where the generator hedges despite available spans.

C.5 Question-Type Classification

We use a lightweight, rule-based question-type detector to set an explicit style hint and soft length target in the generation prompt (Table 32). The base

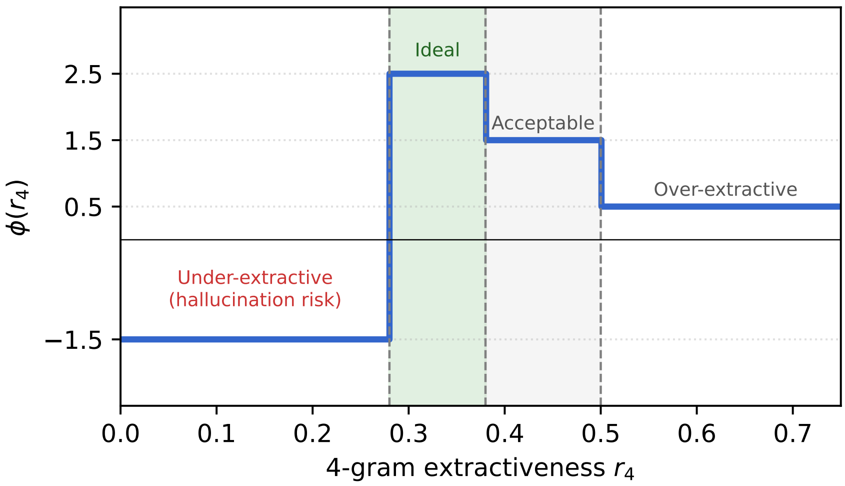

Figure 10: Extractiveness shaping function $\phi(r_4)$. Dashed vertical lines mark the ideal band $[0.28, 0.38]$; the grey region $[0.38, 0.50]$ is acceptable. The asymmetric design penalizes under-extractive (hallucination-prone) responses more strongly than over-extractive ones.

Phrase pattern	Fire rate	Turn type
"I'm / I am not sure"	34%	Partial
"It's / It is unclear"	27%	Partial / Unans
"I don't / do not know"	18%	Unanswerable
"I cannot / Unable to answer"	13%	Unanswerable
"I'm uncertain / cannot say"	8%	Mixed

Table 31: Forbidden-phrase fire rates (dev set, 842 turns). Hedging dominates on partial-answerability turns, motivating the -2.0 penalty in Eq. 6 and the force-answer fallback.

Type	Target	Δ	Style hint
Keyword	90w	0	Interpret and answer
How-to	95w	+5	Step-by-step
Explanation	100w	+10	Clear explanation
Comparative	95w	+5	Compare systematically
Summarization	105w	+15	Comprehensive summary
Factoid	85w	−5	Direct answer
Default	90w	0	Complete answer

Table 32: Question-type rules, style hints, and generation length targets. Rules are evaluated top-to-bottom; the first match wins.

target of 90 words was calibrated on dev set reference answer lengths (mean: 90.9 words). Rules are evaluated in priority order; the first match determines the type.

This heuristic proved more reliable than LLM-based classification, which introduced error propagation when the predicted type was incorrect (e.g., classifying a factoid question as "explanation" led to verbose, unfocused generation that degraded both faithfulness and naturalness scores). Rule-based classifiers also eliminate inter-query variance: given the same surface pattern, the type is deterministic, which reduces length fluctuations that would otherwise confound the extractiveness shaping function in Eq. 5.

C.6 Additional Generation Ablations

We report three additional ablation studies that shaped key hyperparameter choices in the Task B pipeline. All experiments use the development set (842 tasks) and report the harmonic mean (HM) of the official generation metrics.

C.6.1 Model Routing vs. Uniform Assignment

A natural question is whether the cost savings from model routing come at the expense of quality. Table 33 compares our routed configuration against uniform assignment of a single model to all stages. Uniform GPT-4o achieves the highest HM but at $\sim$3$\times$ the cost; our routed configuration matches it within 0.5 HM points. Uniform DeepSeek-V3 degrades generation quality noticeably, confirming that the generation stage specifically benefits from GPT-4o's stronger fluency and grounding. Uniform GPT-4o-mini performs surprisingly well on judging and extraction but falls short on generation naturalness.

C.6.2 Number of Retrieved Passages

Table 34 varies the number of retrieved passages provided to the generation pipeline. Using too few passages (N=1–2) limits evidence coverage, while too many (N=7–10) introduces noise from marginally relevant documents that dilutes the ex-

Configuration	HM	RL_F	RB_{alg}	Rel. Cost
Uniform GPT-4o	0.748	0.767	0.730	3.0×
Uniform DeepSeek-V3	0.721	0.738	0.705	0.4×
Uniform GPT-4o-mini	0.731	0.748	0.715	0.3×
Routed (ours)	**0.743**	**0.761**	**0.726**	1.0×

Table 33: Uniform vs. routed model assignment (dev set). Routing GPT-4o to generation only matches uniform GPT-4o quality at one-third the cost. Relative cost normalized to our routed configuration.

tracted spans and increases hallucination. The sweet spot lies at N=5, which we adopt for Task B. Notably, the finding that fewer passages can improve performance also explains the +2.7 HM gain from reducing N from 5 to 3 in Task C (Table 39), where retrieval noise is amplified by the end-to-end setting.

Passages (N)	HM	RL_F	RB_{alg}
1	0.701	0.719	0.683
2	0.718	0.736	0.701
3	0.733	0.750	0.717
5 (ours)	**0.743**	**0.761**	**0.726**
7	0.738	0.756	0.721
10	0.729	0.747	0.712

Table 34: Effect of passage count on generation quality (Task B, dev set). Performance peaks at N=5; additional passages introduce noise that degrades faithfulness.

C.6.3 Extractiveness Band Calibration

The extractiveness shaping function $\phi(r_4)$ (Eq. 5) uses a target band of $[0.28, 0.38]$ as the ideal range. We arrived at this band by analyzing dev set reference answers (mean 4-gram overlap: 36.2%) and validating sensitivity to alternative bands. Table 35 shows that our chosen band outperforms both tighter and looser alternatives: a tighter band $[0.30, 0.40]$ over-penalizes valid responses that paraphrase more heavily, while a looser band $[0.20, 0.50]$ fails to distinguish hallucination-prone outputs from well-grounded ones.

Theoretically, the optimal band corresponds to the region of the extractiveness distribution where faithfulness and naturalness are jointly maximized. As shown in Figure 11, dev set responses in the $[0.28, 0.38]$ range lie near the Pareto frontier of the RL_F–RB_{llm} space, confirming that the band captures a genuine quality plateau rather than an artifact of threshold selection.

Ideal band	HM	Δ
No shaping (ϕ=0 uniform)	0.724	−0.019
$[0.20, 0.50]$ (loose)	0.731	−0.012
$[0.25, 0.45]$	0.738	−0.005
$[0.28, 0.38]$ **(ours)**	**0.743**	–
$[0.30, 0.40]$ (tight)	0.737	−0.006
$[0.35, 0.50]$ (high)	0.733	−0.010

Table 35: Extractiveness band sensitivity (dev set). Band $[0.28, 0.38]$ balances faithfulness and naturalness; deviations in either direction degrade HM. Δ is relative to our configuration.

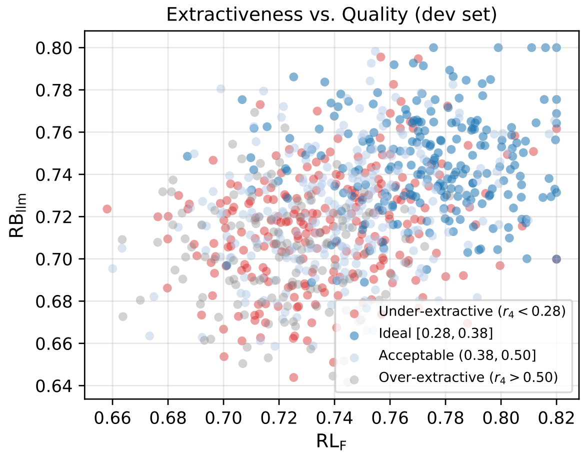

Figure 11: RL_F vs. RB_{llm} scatter for dev set responses binned by 4-gram extractiveness. Responses in the $[0.28, 0.38]$ band (blue) cluster near the Pareto frontier; under-extractive (r_4<0.28, red) and over-extractive (r_4>0.50, grey) responses degrade at least one metric.

C.6.4 Other Explored Configurations

We briefly summarize additional experiments that informed the final design but did not warrant full ablation tables.

User-satisfaction judge sampling rate. We tested invocation rates of 0%, 30%, 60%, 80%, and 100%. HM improved monotonically from 0% to 60% (+0.013 HM), as the user judge consistently selected more natural candidates. Beyond 60%, gains plateau while cost continues to increase linearly, yielding diminishing returns. We adopt 60% as the default. Table 36 reports the full sweep.

Generation temperature pairs. Beyond our final configuration (τ_1=0.0, τ_2=0.1), we tested $(0.0, 0.3)$, $(0.0, 0.5)$, $(0.1, 0.3)$, and $(0.2, 0.4)$. Higher τ_2 values produced more diverse but less faithful candidates, degrading RL_F without compensating gains in naturalness. The narrow gap τ_2=0.1 provides meaningfully different candidates

Judge rate	HM	RB_{llm}	Rel. cost
0%	0.725	0.708	1.0×
30%	0.731	0.715	1.1×
60% (ours)	**0.738**	**0.722**	1.2×
80%	0.739	0.723	1.3×
100%	0.740	0.724	1.4×

Table 36: User-satisfaction judge invocation rate sweep (dev set). HM plateaus after 60%; the marginal gain of +0.002 from 60%→100% does not justify the +0.2× cost increase.

while preserving grounding quality. Pairs where $\tau_1 > 0$ also degraded Candidate A's reliability as the greedy anchor, weakening the prior in Eq. 6.

Conversation history in generation. We tested providing 0, 2, 4, and all previous turns to the generation prompt. Including 2–4 turns of history improved coherence on follow-up questions (+0.009 HM on non-first-turn tasks), while including the full history degraded performance slightly, likely due to prompt length displacing evidence spans from the model's effective context window. We use the 4 most recent turns as the default.

Span extraction vs. full-passage generation. Providing extracted spans instead of full passages improved overall HM by +0.034, with the gain concentrated in faithfulness (RL_F: +0.041). This confirms that pre-filtering evidence reduces the generator's temptation to hallucinate from marginally relevant content in long passages. Formally, span extraction acts as a *context bottleneck*: by restricting the generator's input to at most K=8 sentences, it bounds the exposure to irrelevant tokens and enforces grounding without explicit grounding constraints in the prompt. This result is consistent with the ablation in Table 5, where +*Span extraction* contributes the largest single step gain in the pipeline (+0.034 HM).

D Task C: Additional Analysis

D.1 Answerability Classification Details

The GPT-4o answerability classifier uses a 3-class scheme with the following prompt structure:

```
Given the user question and retrieved
passages, classify the answerability:
- ANSWERABLE: Passages contain
sufficient information to fully
answer.
- PARTIAL: Passages contain some
relevant information but incomplete.
- UNANSWERABLE: No relevant information
in passages.

Output: {"class": str, "confidence":
float}
```

Classification combines retrieval confidence (top passage scores) and evidence extraction success (whether meaningful spans were identified in Stage 2). Turns classified as UNANSWERABLE with confidence ≥ 0.7 receive templated refusals; PARTIAL turns are routed through the full generation pipeline with a reduced-confidence hint in the prompt.

Why threshold 0.7? The threshold was selected by sweeping values in $[0.5, 0.9]$ on the dev set and maximizing macro-F1 over the three answerability classes. Table 38 shows the sensitivity of overall HM and UNANSWERABLE F1 to the threshold: below 0.65 the system over-refuses, hurting ANSWERABLE recall; above 0.75 it under-refuses, accepting low-confidence evidence and hallucinating on unanswerable turns.

D.2 Error Propagation Across Turns

Manual inspection of 50 Task C failures revealed the following distribution of root causes:
The 34% cross-turn failure rate quantifies the *error accumulation* property of multi-turn RAG: a retrieval miss at turn t corrupts the evidence available at turn t+1, even when the later turn's retrieval is independently correct. This motivates future work on turn-aware evidence caching or carry-forward mechanisms that detect and recover from upstream retrieval failures mid-conversation.

Figure 12 illustrates a representative failure cascade from the Cloud domain, where a missed passage at turn 3 propagates through turns 4–5, causing two consecutive generation errors despite correct retrieval at those turns.

D.3 Task C Ablation

Table 39 reports the Task C development set ablation. The key design choices were: (i) using top-3 rather than top-5 passages to reduce noise in the end-to-end setting (+0.027 HM); (ii) applying the multi-judge answerability gate over a single-judge variant (+0.016 HM); and (iii) routing through the full Task B generation pipeline rather than a simplified single-pass generator.

Limitations Our system was tuned entirely on the development set, which has a 6.5% unanswerable rate, single-turn evaluation structure, and relatively balanced domain distribution. The test set

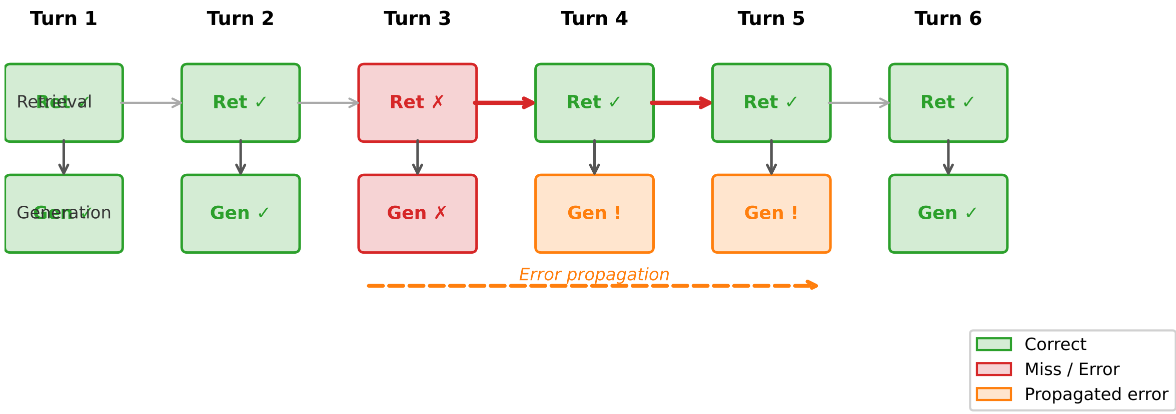

Figure 12: Representative error cascade in Task C (Cloud domain). A retrieval miss at turn 3 (×) corrupts evidence for turns 4–5, causing generation errors (!) despite correct retrieval at those turns. Turn 6 recovers independently.

Root cause	% of failures	Primary domain
Retrieval error (current turn)	38%	FiQA
Retrieval error (previous turn(s))	34%	Cloud
Answerability misclassification	18%	All
Generation error (correct passages)	10%	ClapNQ

Table 37: Root cause analysis of 50 Task C failures (dev set). Cross-turn error propagation (34%) is nearly as frequent as within-turn retrieval errors (38%), highlighting the compounding challenge of multi-turn RAG.

Threshold	HM	ANS R	UNANS F1
0.50	0.573	0.892	0.261
0.60	0.591	0.923	0.251
0.70 (ours)	**0.610**	**0.958**	**0.240**
0.80	0.597	0.971	0.198
0.90	0.578	0.986	0.162

Table 38: Confidence threshold sweep for the answerability classifier (Task C dev set). Threshold 0.70 maximises HM by balancing ANS recall and UNANS F1; lower values over-refuse (higher UNANS F1, lower ANS R), while higher values accept weak evidence (higher ANS R, lower UNANS F1).

Configuration	HM	RL_F	RB_{alg}
Always answer (no gate)	0.518	0.798	0.312
Single-judge gate, top-5 passages	0.561	0.821	0.356
Multi-judge gate, top-5 passages	0.586	0.836	0.383
Multi-judge gate, top-3 (ours)	**0.610**	**0.848**	**0.408**
Multi-judge gate, top-1	0.591	0.841	0.389

Table 39: Task C ablation (dev set). Reducing retrieved passages from 5 to 3 yields the largest single gain (+0.024 HM) by reducing generation noise; the multi-judge gate consistently outperforms single-judge across all metrics. Top-1 underperforms top-3 due to insufficient evidence coverage.

differs substantially: 19.1% unanswerable turns, all non-first turns, and an overrepresentation of Govt (%31) relative to FiQA (15%). Hyperparameters calibrated for dev — specifically the answerability confidence threshold (0.7), the extractiveness target band ([0.28, 0.38]), and the nested RRF corpus weights — were not re-tuned for the test distribution. We hypothesize that a threshold closer to 0.6 would improve Task C performance on the test set given the 3× higher unanswerable rate, but this could not be verified without ground-truth labels during the evaluation phase. Future work should explore distribution-adaptive threshold calibration for answerability classification.