

Sifei at SemEval-2026 Task 8: Hybrid Retrieval and Query Rewriting for Multi-Turn RAG

Sifei Meng

Faculty of Computer Science
HSE University
smen@edu.hse.ru

Dmitry Ilvovsky

Faculty of Computer Science
HSE University
dilvovsky@hse.ru

Abstract

Multi-turn retrieval-augmented generation (RAG) is challenging due to evolving user intent, conversational noise, and strict context limits. We propose a training-free, hybrid retrieval pipeline for SemEval-2026 Task 8 that combines dense and sparse retrieval with controlled query rewriting and cross-encoder reranking. On the official test set of Task A, our system achieves 0.5453 nDCG@5, ranking third among 38 teams and outperforming the strongest baseline (0.4795). For Task C, we reuse the documents retrieved for Task A and apply a lightweight generation pipeline guided by the official prompt, achieving 0.5312 (the harmonic mean of relevance and faithfulness) and ranking 15th out of 29 teams. All retrieval components are open-source, while rewriting and generation rely on LLM APIs. Code and scripts are available on GitHub¹.

1 Introduction

Multi-turn retrieval-augmented generation (RAG) is central to real-world chat assistants. While many benchmarks focus on single-turn queries and underrepresent multi-turn phenomena, the English MTRAG benchmark (Katsis et al., 2025) captures contextual shifts with human-generated responses, showing that even strong RAG systems degrade on later turns and context-dependent queries.

SemEval-2026 Task 8 (MTRAGEval) (Rosenthal et al., 2026b) evaluates multi-turn RAG in three subtasks. Task A focuses on retrieval, where systems rank the top-10 relevant passages. Task B evaluates response quality given gold relevant passages. Task C is a RAG setting: systems retrieve up to 10 passages from the corpus and generate a grounded response. We participate in Task A and Task C because together they form an end-to-end RAG system.

¹https://github.com/mengsifei/Multi_turn_RAG

Our Task A system is a training-free hybrid pipeline that combines dense and sparse retrievers with corpus cleaning, conversational query rewriting, and reranking. We sweep cleaning strategies, dense backbones, context-window and chunking settings, sparse variants, rerankers, and fusion methods, finding that (i) strict cleaning improves dense retrieval, with Jasper-Token-Compression-600M performing best, (ii) SPLADE-v3 is the strongest sparse component, and (iii) reciprocal rank fusion with a BGE reranker (v2-m3) and an α -rescoring yields the best nDCG@5. The final system reaches 0.5453 nDCG@5 on the official test set (Rosenthal et al., 2026a), ranking 3rd out of 38 teams in Task A. For Task C, we reuse the retriever and conversational query reconstruction within the official generation framework, achieving 0.5312 and ranking 15th out of 29 teams.

Our contributions are threefold. (1) We propose a training-free hybrid multi-turn retrieval pipeline that uses open-source retrieval and reranking components and LLM APIs for query rewriting, ranking third in Task A. (2) We provide systematic ablations and analysis for Task A to identify the design choices that yield the largest gains. (3) We find that concatenating the last turn with a controlled standalone rewrite consistently improves retrieval and generation at no additional training cost.

2 Background

2.1 MTRAGEval Task Setup

MTRAGEval is built on the MTRAG benchmark, consisting of human-generated multi-turn conversations grounded in four domain corpora: CLAPNQ, FiQA, Govt, and Cloud. The released benchmark contains 110 conversations with 7.7 turns per conversation on average, totaling 842 turn-level tasks; after filtering unanswerable queries, we obtain 777 tasks in our experiments. Given a dialogue history ending with the final user query, **Task A** requires

an ordered list of $k=10$ passages from the corresponding corpus, evaluated by nDCG@5 on the answerable subset. **Task C** uses retrieved passages to generate the final assistant response and ranks systems by the harmonic mean of multiple metrics measuring response quality and faithfulness to evidence (Katsis et al., 2025). Following the official release, the organizers provide each corpus as pre-segmented passages (512 tokens with 100-token overlap), and evaluation is performed at the same passage level using these passage IDs. We therefore reuse the provided passage segmentation and passage IDs.

2.2 Related Work

Prior work on retrieval-augmented generation (RAG) highlights several recurring components for improving retrieval robustness and ranking quality.

Preprocessing and chunking. Several studies (Merola and Singh, 2025; Zhao et al., 2025; Bansal, 2023) show that corpus refinement and contextual chunking can improve retrieval alignment.

Query reformulation in multi-turn conversations. For ambiguous or non-standalone queries, query rewriting and keyword-based reformulation are effective (Katsis et al., 2025; Chen et al., 2025b; Pattanayak et al., 2025); concatenating a hypothetical document with the user query is also beneficial (Wang et al., 2024).

Hybrid retrieval to increase recall. Hybrid pipelines combine sparse and dense retrievers with fusion methods such as reciprocal rank fusion, which consistently improves retrieval quality (Kalra et al., 2025); hybrid retrieval performance is also sensitive to fusion hyperparameters (Wang et al., 2024).

Reranking to increase precision. Reranking consistently improves nDCG in competitive RAG settings, and ablations identify it as a major contributor to final ranking quality (Lee and Roh, 2024).

We evaluate these techniques on MTRAGEval and provide ablations to quantify their impact.

3 System Overview

Our submissions follow a multi-stage hybrid retrieval pipeline for Task A, and reuse the same retriever for Task C generation. The overall retriever workflow is shown in Figure 1.

3.1 Query reconstruction

Given a multi-turn dialogue, we construct the retrieval query by concatenating the last user turn

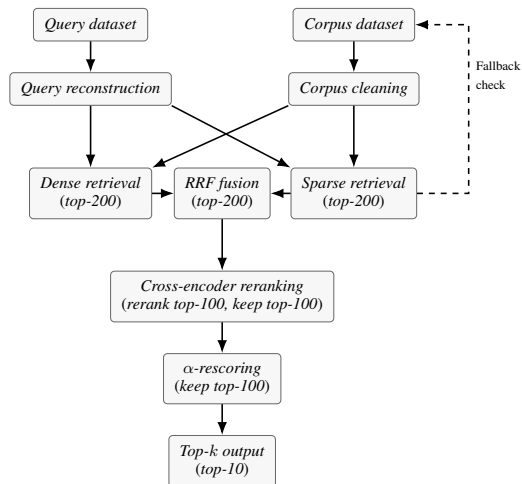


Figure 1: Hybrid retrieval and reranking workflow.

with a custom standalone rewrite generated by GPT-5.2 (prompt provided in Appendix A.1 and fixed before official evaluation). The last turn preserves surface phrasing, while the rewrite resolves implicit constraints and reduces ambiguity. The final query is:

$$q = \text{last_turn} \parallel \text{custom rewritten query}.$$

This concatenation consistently outperforms either component alone and is used for both retrieval (Task A) and generation (Task C). All rewritten queries are generated once, cached, and not modified after official test submission; the files are included in our submission artifacts.

3.1.1 Analysis of Conversational Rewriting

On the MTRAG training split, our rewritten queries achieve lower language-model perplexity than the official rewrites across all four domains (CLAPNQ: 91.60→77.69; Cloud: 124.88→105.08; FiQA: 78.42→60.64; Govt: 91.35→90.05), suggesting more regular and self-contained standalone formulations. We also observe a small reduction in ambiguous deictic markers (e.g., *this*, *that*, *these*), indicating improved contextual grounding.

3.2 Corpus preparation and indexing

To reduce noise in dense retrieval, we build a cleaned corpus by removing non-English passages, near-duplicates (detected via normalized-text hashing, e.g., whitespace and markup canonicalization), ultra-short fragments, and pathological overlong texts, while protecting any qrels-referenced passages from filtering. In hybrid retrieval, we query the cleaned corpus for both dense and sparse components; when cleaning invalidates sparse hits for

a given query (i.e., some retrieved passage IDs are absent from the cleaned index), we rerun sparse retrieval on the original corpus for that query to preserve lexical coverage.

3.3 Stage-1 retrieval: dense and sparse

We retrieve an initial candidate pool with hybrid retrieval. For dense retrieval, we use Jasper-Token-Compression-600M with 0.5 compression ratio (Zhang et al., 2025), selected for its strong English MMTEB performance (Enevoldsen et al., 2025); after sweeping settings, we use a 768-token query/passage budget to retrieve top-200 candidates. For sparse retrieval, we use SPLADE-v3 (Lassance et al., 2024) to retrieve another top-200, which helps on entity-centric and exact-match queries via stronger lexical anchoring.

3.4 Hybrid fusion

We merge dense and sparse ranked lists using Reciprocal Rank Fusion (RRF) with $k=10$, producing a single hybrid candidate list of the top-200 fused candidates for downstream reranking. RRF provides a robust combination without requiring score-scale calibration between heterogeneous retrievers.

3.5 Cross-encoder reranking and α -rescoring

From the fused list, we rerank the top-100 candidates using the cross-encoder BAAI/bge-reranker-v2-m3 (Chen et al., 2025a), truncating inputs to 1024 tokens. To further calibrate the final ordering, we perform α -weighted score fusion between the reranker score and the initial retrieval score:

$$s(d) = \alpha \cdot s_{retr}(d) + (1 - \alpha) \cdot s_{ce}(d),$$

where α is tuned on the released training split only (final $\alpha=0.15$) and fixed before official test submission, $s_{retr}(d)$ is the hybrid stage-1 retrieval score after RRF fusion, and $s_{ce}(d)$ is the cross-encoder score for candidate d . We then output the top-10 passages as the Task A result.

3.6 Task C pipeline

For Task C, we feed the model the top-5 retrieved passages and the same concatenated query reconstruction from Task A, which also helps generation. Due to limited development time, we keep the generation module lightweight and close to the official setup.

4 Experimental Setup

4.1 Data usage and evaluation protocol

We do not fine-tune any model in this work. The dataset is relatively small, which makes stable supervised tuning difficult. We therefore use the entire training dataset for development-time evaluation and follow the official passage-level segmentation and evaluation setup. All hyperparameters (including α , fusion parameters, and candidate pool sizes) are tuned on the training dataset only and fixed before official test submission. For Task A we report nDCG@5 on the answerable subset; for Task C we report the official harmonic-mean score over response quality and faithfulness metrics (Katsis et al., 2025). Unless otherwise stated, all development-split retrieval results are averaged over three repeated runs; we observed only minor variation across runs.

4.2 Model selection and hyperparameter sweeps

We conduct systematic sweeps over corpus cleaning strategies, dense and sparse retrievers, context length, fusion strategies, rerankers, and query reconstruction.

For dense retrieval, we evaluate multiple open-source embedding models, including BGE-1.5-base², Qwen3-Embedding-0.6B³, Jasper-Token-Compression-600M⁴, gte-Qwen2-1.5B-instruct⁵, and Yuan-embedding-2.0-en⁶. Sparse retrieval candidates include BM25 and SPLADE variants (v2-distil⁷ and v3⁸). For reranking, we consider cross-encoders such as BGE-reranker-v2-m3⁹, all-MiniLM-L6-v2¹⁰, and Qwen3-Reranker-0.6B¹¹.

We tune retrieval and fusion hyperparameters on the development evaluation, including dense context window size, RRF parameters, and an α sweep for score calibration.

Our final configuration uses Jasper-Token-Compression-600M for dense retrieval, SPLADE-v3 for sparse retrieval, RRF fusion with tuned

²<https://huggingface.co/BAAI/bge-base-en-v1.5>

³<https://huggingface.co/Qwen/Qwen3-Embedding-0.6B>

⁴<https://huggingface.co/infgrad/Jasper-Token-Compression-600M>

⁵<https://huggingface.co/Alibaba-NLP/gte-Qwen2-1.5B-instruct>

⁶<https://huggingface.co/IEITYuan/Yuan-embedding-2.0-en>

⁷https://huggingface.co/naver/splade_v2_distil

⁸<https://huggingface.co/naver/splade-v3>

⁹<https://huggingface.co/BAAI/bge-reranker-v2-m3>

¹⁰<https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

¹¹<https://huggingface.co/Qwen/Qwen3-Reranker-0.6B>

$k = 10$, BGE-reranker-v2-m3 for cross-encoder reranking, and α -rescoring with $\alpha = 0.15$.

4.3 Generation setup

For Task C, we reuse the Task A retriever and provide the top-5 retrieved passages together with the same concatenated query (last turn + standalone rewrite); the full prompt is given in Appendix A.2. Due to GPU constraints, we use the DeepSeek-V3.2 API for answer generation with low-temperature decoding (temperature=0.2, top_p=1.0, max_new_tokens=300). During development, we used DeepSeek-V3.2 as an LLM-as-a-judge via API (temperature=0, max_new_tokens=800) for metrics RL_F and RB_llm. Official Task C scores are computed by the organizers using the provided evaluation scripts and the harmonic mean of RL_F, RB_llm, and RB_alg.

4.4 Implementation and compute

All experiments in Task A use a single NVIDIA V100 GPU (32GB). We cache document embeddings for dense retrieval to enable efficient sweeps over fusion and reranking settings.

5 Results

5.1 Official leaderboard performance

Table 1 reports our official results. Our hybrid retriever achieves 0.5453 nDCG@5 on Task A, ranking 3rd out of 38 teams. On Task C, our system yields 0.5312, ranking 15th out of 29. The faithfulness score is substantially higher than the relevance scores on the evaluation set (RB_alg 0.3841, RL_F 0.7286, RB_llm 0.5982), suggesting that the retrieved documents provide strong grounding for answer generation.

Compared with the top systems, our method is competitive on retrieval while relying mainly on open-source components. Top Subtask A systems consistently used conversational query rewriting to produce self-contained queries before retrieval, and two of the top three teams used sparse retrieval. In particular, the winning AILS-NTUA system combined five history-aware rewrites with Elser-v1 retrieval, cohere-rerank-v4.0-pro reranking, and weighted RRF fusion. One factor that may have contributed to this result is the use of Elser-v1, given that the organizers note a possible benchmark bias toward Elser because it was used during data creation (Katsis et al., 2025). In addition, using

Subtask A (nDCG@5)			
Rank	Team	Score	Note
1	AILS-NTUA	0.578	Sparse
2	uva-irlab-conv	0.548	Sparse
3	Ours (Sifei)	0.545	Hybrid
Subtask C (HM)			
Rank	Team	Score	Note
1	GenAIus	0.586	GPT, ZS
2	HowardU-AI4PC	0.580	Llama, ZS
3	5ting	0.560	GPT, ZS
15	Ours (Sifei)	0.531	DS, ZS

Table 1: Top-3 leaderboard entries and our submission on MTRAGEval.

multiple rewrites may help expose complementary aspects of the information need before fusion.

For Subtask C, the top systems, GenAIus and Howard University-AI4PC, employed richer hybrid pipelines with multi-query rewriting, HyDE, lexical-dense retrieval, and reranking. Both also used BM25, which was less effective than SPLADE-v3 in our own sparse-retrieval experiments, suggesting that BM25 may still be useful as a complementary lexical branch inside a stronger hybrid pipeline. Generation was performed in a zero-shot setting with large models such as GPT-4o and Llama-3.3-70B-Instruct.

5.2 Task A ablation study

Table 2 shows a progression from recall-oriented improvements to precision-focused refinements.

As a baseline, we use dense-only retrieval with Jasper-Token-Compression-600M at a compression ratio of 0.5 in the last-turn setting, i.e., searching with the raw final user query without rewriting.

Corpus cleaning yields a modest but consistent gain, as dense retrievers are sensitive to templated boilerplate, duplicated blocks, and malformed passages that introduce spurious neighbors. Increasing the dense context window to 768 tokens further improves performance by preserving more entities and constraints in the embedding input. This context length is not tied to the official 512-token segmentation, since our dense encoder relies on a different tokenizer and embedding scheme; empirically, 768 tokens cover over 99% of passages after preprocessing.

Query rewriting produces a larger improvement, suggesting that many last-turn queries lack sufficient standalone context. Compared with the official rewrites, our custom rewrites have lower language-model perplexity, indicating cleaner and more regular formulations.

Setting	nDCG@5	Δ
Last-turn baseline	0.3457	–
+ Cleaned corpus (qrels-protected)	0.3497	+0.0040
+ Context Window (768 tokens)	0.3507	+0.0010
+ Official rewrite queries	0.3938	+0.0431
+ Custom rewrite queries	0.4144	+0.0206
+ Concat(last-turn rewrite) queries	0.4150	+0.0006
+ SPLADE-v3 + RRF (k=10)	0.4668	+0.0518
+ BGE reranker v2-m3 + α -rescoring ($\alpha=0.15$)	0.4896	+0.0228

Table 2: Key ablations for Task A on the released train split used for pipeline development. All nDCG@5 numbers are averaged over three runs.

Method	HM \uparrow	RB_alg \uparrow	RL_F \uparrow	RB_llm \uparrow
Query Rewrite	0.5813	0.4415	0.8540	0.6770
Concatenation	0.5922	0.4481	0.8563	0.6910
Δ	+0.0109	+0.0066	+0.0023	+0.0140

Table 3: Task C ablation on query formulation (training split). Concatenation improves HM by 1.87% relative (0.5813 \rightarrow 0.5922), with the largest gain on RB_llm.

Concatenating the last turn with the rewrite outperforms either component alone, likely because it preserves the user’s phrasing while adding missing contextual information. Finally, combining dense retrieval with SPLADE-v3 via RRF substantially increases nDCG by recovering entity-heavy and exact-match evidence that dense retrieval alone may miss. We find that fusion and reranking are sensitive to design choices: small candidate pools limit reranking gains, whereas large pools (e.g., 500–1000) introduce noise that can degrade final ranking quality.

The gap between the train-split ablation result (0.4896 nDCG@5) and the official test result (0.5453) is expected, since the two numbers were obtained on different splits.

5.3 Task C Ablation Study

Due to time constraints, we devoted less effort to Task C and largely followed the official prompt. Still, we observed a practical gain from simply concatenating the rewritten query with the last-turn query. We therefore compare two query formulations for Task C under the same pipeline and evaluation setting: (1) **Rewrite**: use the custom rewritten standalone query, as described in Section 3.1; (2) **Concatenation**: concatenate the *raw last user turn* with the rewritten query. We report RB_alg, RL_F, RB_llm, and their harmonic mean (HM).

As shown in Table 3, **Concatenation** improves all metrics, with the largest gain on **RB_llm** (+0.0140). One possible explanation is that rewrit-

ing alone can over-compress the query and bias it toward the rewriter’s interpretation. Appending the **raw last turn** preserves additional cues from the user’s phrasing and anchors the model to the current question. This simple change strengthens both generation relevance and faithfulness, providing a simple and effective improvement for Task C.

5.4 Error analysis on the test set (Task A)

Score distribution. The score distribution is skewed: 61.1% of queries achieve nDCG@5 $>$ 0.5, while 16.8% obtain nDCG@5 = 0; 2.8% fall in (0, 0.2] and 19.3% fall in (0.2, 0.5]. Thus, errors concentrate in a relatively small subset of difficult instances.

Stage-level failures. For low-score cases (nDCG@5 $<$ 0.2), we categorize errors into two types: (i) *recall failures*, where no relevant passage appears in the top-10, and (ii) *rerank failures*, where a relevant passage is retrieved but not promoted to the top ranks. Under this diagnostic, 85.7% of hard cases are recall failures and 14.3% are rerank failures, indicating that insufficient recall in the final top-10 is the primary bottleneck.

Domain and turn effects. Recall failures are most frequent in the Cloud and FiQA domains of the MTRAG benchmark (Katsis et al., 2025), which contain a higher proportion of noisy text, underscoring the importance of effective data cleaning. Hard cases also concentrate in later conversation turns (especially after the fifth turn), indicating that query rewriting remains challenging as dialogue context grows.

In contrast, reranking failures are substantially less common, implying that improving first-stage recall is likely to yield larger gains than further reranker optimization.

6 Discussion

Alternative chunking did not help. We evaluated several segmentation strategies beyond the official fixed-window passages, including (i) semantic re-chunking on top of the official passages, (ii) document-level chunking with outputs mapped back to the official passage IDs, and (iii) passage packing. None consistently outperformed the official setup. We attribute the degradation mainly to: (i) over-fragmentation into small chunks that lack sufficient context; (ii) evidence being split across multiple units, weakening relevance signals; (iii)

a mismatch between chunk-level retrieval scores and passage-level evaluation; and (iv) redundancy introduced by overlap in packing. Together, these factors reduce discriminative power under passage-level ranking.

Reranking and fusion order matter. Hybrid retrieval with RRF improves recall but does not always improve top-rank precision (e.g., nDCG@1/3). Cross-encoder reranking helps refine the top results, yet very large candidate pools (e.g., 1000) introduce many near-duplicates and hard negatives, making the final ranking sensitive to small scoring differences and degrading nDCG@5. We find that both the fusion–reranking order and the candidate pool size affect performance. In practice, reranking a moderate pool (top-100) followed by α -rescoring ($\alpha=0.15$) yields the most stable results.

Keyword-style augmentation underperformed. Retrieval-oriented rewrites (e.g., keyword lists or hypothetical-document expansion) consistently hurt performance by distorting intent and over-amplifying rare terms. In contrast, controlled conversational rewriting combined with last-turn concatenation provides consistent gains across domains.

7 Conclusion

We presented an end-to-end RAG system for SemEval-2026 Task 8 (MTRAGEval), featuring a multi-stage hybrid retrieval pipeline for Task A and a lightweight generation pipeline for Task C. Our retrieval system combines qrels-protected corpus cleaning, conversational query rewriting with last-turn concatenation, dense and sparse retrieval, calibrated fusion, and cross-encoder reranking. Our final submission ranked 3rd in Task A, achieving 0.5453 nDCG@5. Ablations show that query reconstruction and hybrid retrieval deliver the largest gains, and concatenating the last-turn query with its rewrite consistently improves performance in both Tasks A and C at no additional training cost.

Future work will focus on more robust late-turn rewriting and evidence alignment, as well as improving response generation for both answerable and underspecified queries.

Ethical considerations. Our system is evaluated on the provided public benchmark and uses only the released conversations and corpora.

References

- Ashish Bansal. 2023. [Optimizing rag with hybrid search and contextual chunking](#). *Journal of Engineering and Applied Sciences Technology*.
- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2025a. [M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#). *Preprint*, arXiv:2402.03216.
- Zhiyu Chen, Biancen Xie, Sidarth Srinivasan, Qun Liu, Manikandarajan Ramanathan, and Raj Maragoud. 2025b. [Building multi-turn rag for customer support with llm labeling](#).
- Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal, Jonathan Ryrstrøm, Roman Solomatin, and 67 others. 2025. [Mmteb: Massive multilingual text embedding benchmark](#). *arXiv preprint arXiv:2502.13595*.
- Rishi Kalra, Zekun Wu, Ayesha Gulley, Airlie Hilliard, Xin Guan, Adriano Koshiyama, and Philip Treleaven. 2025. [Hypa-rag: A hybrid parameter adaptive retrieval-augmented generation system for ai legal and policy applications](#). *Preprint*, arXiv:2409.09046.
- 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.
- Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. 2024. [Splade-v3: New baselines for splade](#). *Preprint*, arXiv:2403.06789.
- Joohyun Lee and Minji Roh. 2024. [Multi-reranker: Maximizing performance of retrieval-augmented generation in the financerag challenge](#). *Preprint*, arXiv:2411.16732.
- Carlo Merola and Jaspinder Singh. 2025. [Reconstructing context: Evaluating advanced chunking strategies for retrieval-augmented generation](#). *Preprint*, arXiv:2504.19754.
- Priyaranjan Pattanayak, Amit Agarwal, Hansa Megh-wani, Hitesh Laxmichand Patel, and Srikant Panda. 2025. [Hybrid ai for responsive multi-turn online conversations with novel dynamic routing and feedback adaptation](#). *Preprint*, arXiv:2506.02097.
- 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.

Xiaohua Wang, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, Zhengyuan Wang, Shizheng Li, Qi Qian, Ruicheng Yin, Changze Lv, Xiaoqing Zheng, and Xuanjing Huang. 2024. [Searching for best practices in retrieval-augmented generation](#). *Preprint*, arXiv:2407.01219.

Dun Zhang, Ziyang Zeng, Yudong Zhou, and Shuyang Lu. 2025. [Jasper-token-compression-600m technical report](#). *Preprint*, arXiv:2511.14405.

Jihao Zhao, Zhiyuan Ji, Yuchen Feng, Pengnian Qi, Simin Niu, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2025. [Meta-chunking: Learning text segmentation and semantic completion via logical perception](#). *Preprint*, arXiv:2410.12788.

A Appendix

A.1 Prompt for query rewriting

Purpose: Rewrite a multi-turn conversational query into a single, clean, standalone English question, following the style of official rewrites.

Input: A conversation history with multiple user turns. The last line is the current user question; earlier lines provide context.

Task: Rewrite the last user question only into a standalone question that can be understood without the conversation history. Use earlier turns only to resolve ambiguity (pronouns, ellipsis, vague references).

Requirements:

1. Output exactly one complete English question.
2. Preserve the original intent and interrogative structure (who/what/where/when).
3. Do not introduce new concepts; do not merge multiple questions; do not copy the conversation history.
4. Remove conversational fillers (e.g., “No, I meant...”, “Thanks”, etc.).

Output format: Return a JSON object per example: {"task_id": ...

```
, "collection": ... , "text":
"|user|: ...". Copy task_id and
collection from the input; task_id
has the format hash<::>turn.
```

Example:

Conversation history:

```
> who takes photos of planes in
  the air
> No, I meant photos in the air.
```

Rewrite:

```
> Who takes photos of airplanes
  while they are in the air?
```

A.2 Prompt for Task C generation

Given the following documents and a user question, generate a concise answer (less than 150 words) that is grounded in the provided documents.

1. Use only information that can be supported by the documents.
2. You may paraphrase and summarize across documents.
3. Do NOT quote verbatim unless necessary.
4. If the documents do not contain enough information to answer the question, say: "I do not have specific information."

[Documents]

[Document 1]

```
{context_1}
```

...

[Document 5]

```
{context_5}
```

[Question]

```
last_turn_query || rewrite_query
```

[Answer]