UTRAG at SemEval-2026 Task 8: History-Aware Query Rewriting and LoRA-Finetuned Generation for Multi-Turn RAG

Yi-Shan Lin **Ke Zhou**
Computational Linguistics, University of Tübingen
{yi-shan.lin, ke.zhou}@student.uni-tuebingen.de

Abstract

This paper describes our system for SemEval-2026 Task 8: Evaluating Multi-Turn RAG Conversations (MTRAGEval), which evaluates retrieval-augmented generation (RAG) in multi-turn, context-dependent settings. We improve retrieval with history-aware query rewriting and enhance generation faithfulness with a LoRA-adapted model, integrating both into an end-to-end pipeline.

Our approach achieves competitive performance across all subtasks, with nDCG@5 of 0.4855 in Subtask A, a harmonic mean score of 0.6554 in Subtask B, and 0.5159 in Subtask C, outperforming strong baselines in Subtasks A and B while remaining competitive in Subtask C.

Our analysis shows that increasing dialogue length introduces cumulative errors in history selection and query formulation, leading to incomplete or drifting retrieval results and increasing the risk of hallucination.

1 Introduction

Large Language Models (LLMs) are increasingly deployed as conversational assistants for information-seeking tasks. To improve factual grounding and reliability, Retrieval-Augmented Generation (RAG) has become a widely adopted paradigm, where external documents are retrieved and used as evidence during generation (Lewis et al., 2020). While RAG has been extensively studied in single-turn question answering, recent work shows that multi-turn conversations introduce additional challenges that are not captured by single-turn benchmarks, such as non-standalone questions, reference to earlier turns, answerability, and changing information needs across a dialogue (Choi et al., 2018; Reddy et al., 2019; Dalton et al., 2020). For example, a user may first ask "Where does Doctor Strange get his powers from?" and later follow up with "How many films does he appear in?", where the latter question is non-standalone and relies on entities introduced earlier in the dialogue.

The recently-proposed MTRAG benchmark (Katsis et al., 2025) highlights these challenges by providing human-generated multi-turn conversations across multiple domains, and demonstrates that even state-of-the-art RAG systems struggle on later turns and context-dependent queries. To systematically evaluate these issues, SemEval-2026 Task 8: Evaluating Multi-Turn RAG Conversations (MTRAGEval) is organized into three subtasks on English multi-turn conversations: retrieval (Subtask A), generation conditioned on reference passages (Subtask B), and full pipeline evaluation (Subtask C) (Rosenthal et al., 2026).

Our submission addresses two key challenges in multi-turn RAG: unstable retrieval quality and unfaithful generation. For Subtask A, we improve retrieval quality via history-aware query rewriting, where relevant dialogue turns are selected and incorporated into a standalone query for retrieval. For Subtask B, we focus on improving faithfulness to the provided evidence passages by adapting a decoder-based generator using parameter-efficient fine-tuning with LoRA. For Subtask C, we combine our retrieval and generation components into an end-to-end RAG pipeline to evaluate whether improvements in both modules transfer to overall performance.

Through participation in this task, we observe that increasing dialogue length in multi-turn RAG introduces challenges for history selection and query formulation. Errors in these steps can lead to incomplete or drifting retrieval results, which degrade downstream generation quality.

Our code, data, and analysis scripts are publicly available on GitHub.[1]

[1] UTRAG repository

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 1882–1889
July 3-4, 2026 ©2026 Association for Computational Linguistics

2 Background

In the shared task setting, each instance consists of the dialogue history, the current user query, and a corpus of passages from the corresponding domain. Systems are required to retrieve evidence passages and/or generate responses depending on the subtask. Subtask A evaluates retrieval quality given a multi-turn query, Subtask B evaluates the generation conditioned on the gold evidence passages, and Subtask C evaluates the complete retrieval–generation pipeline (Rosenthal et al., 2026). For Subtask A, the organizers provide sparse and dense retrieval baselines, including BM25 (Robertson, 2009), ELSER (Elastic, 2023), and BGE-based dense retrieval (Xiao et al., 2023). For Subtasks B and C, the MTRAG benchmark paper reports the performance of several existing models as reference points (Katsis et al., 2025).

MTRAG is a human-generated multi-turn RAG benchmark designed to reflect real-world information-seeking dialogues (Katsis et al., 2025). It consists of 110 conversations with an average of 7.7 turns per conversation, resulting in 842 tasks. Each task includes the full dialogue history up to the current turn, together with the last user question, enabling evaluation under context-dependent conditions. Documents are segmented into 512-token passages and indexed for retrieval. As shown in Table 1, the four corpora cover diverse domains and vary substantially in scale.

A recent survey provide a comprehensive overview of RAG architectures and enhancement strategies, including improvements to retrieval, reranking, and robustness in evidence-grounded generation (Sharma, 2025). In parallel, prior work has highlighted hallucination as a fundamental challenge in natural language generation, where models may produce unsupported or fabricated content even when conditioned on external evidence (Ji et al., 2022). Motivated by these directions, our system focuses on improving retrieval robustness in the multi-turn setting and improving faithfulness to reference passages.

3 System Overview

Our system consists of three main components: (1) history-aware query rewriting, dense retrieval, and reranking for Subtask A, (2) a LoRA-adapted generator for Subtask B, and (3) an end-to-end pipeline that combines both for Subtask C. We implement these components mainly using Qwen3 embedding and generation models (Team, 2025). We describe each component in the following subsections.

Hyperparameters, implementation details, and evaluation settings are described in Section 4.

3.1 Retrieval and Reranking (Subtask A)

We consider three official baseline retrievers provided by the task, namely BM25, BGE-base, and ELSER. In our system, we adopt dense bi-encoder retrieval (Karpukhin et al., 2020), where corpus passages and queries are independently encoded into the same vector space and retrieved via nearest-neighbor search (Mitra et al., 2016). We use Qwen3-4B as our embedding model for encoding both queries and passages.

3.1.1 History-Aware Query Rewriting

Multi-turn user queries are often non-standalone and depend on entities or constraints introduced in earlier turns. To address this, we rewrite the current query into a standalone form by incorporating a compact set of relevant context from the dialogue history.

To select relevant context for query rewriting, we compute dense embeddings for the current query and all candidate history questions using the Qwen3-embedding-8B model (Team, 2025). Cosine similarity is used to measure semantic relatedness between the current query and each history turn. We always retain the most recent history question to preserve local context, and select the remaining history questions from earlier turns based on cosine similarity ranking. In the submitted system, we set k=3 and keep_recent=1, meaning that at most three history questions are used for rewriting: the most recent history question plus the top-2 most similar earlier questions.

The selected history questions are concatenated with the current query and rewritten into a standalone question using a constrained prompt with the Qwen3-30B generation model (Team, 2025). The rewritten query is then used as input to the retriever.

3.1.2 Pronoun Resolution

Embedding-based history selection can be degraded by pronouns, since they provide little lexical signal and their referents are often ambiguous across turns. To mitigate this, we apply a history-aware pronoun rewriting step before history selection. Specifically, we use a lightweight language model (Qwen2.5-7B) (Team, 2024) to rewrite his-

Corpus	Domain	Documents	Passages
ClapNQ	Wikipedia	4,293	183,408
Cloud	Technical Documentation	57,638	61,022
FiQA	Finance	7,661	49,607
Govt	Government	8,578	72,422

Table 1: Statistics of the MTRAG benchmark, based on data from the official MTRAG repository (Rosenthal et al., 2026).

tory turns by replacing pronouns with explicit entity mentions inferred from earlier context. This yields more explicit history questions and improves retrieval robustness.

3.1.3 Reranking

To improve evidence quality, we rerank retrieved passages using a cross-encoder (Nogueira and Cho, 2019a). The reranker scores each query–passage pair and selects the final top-k evidence passages.

Notably, for reranking we use a richer query formulation than the concise standalone rewrite used for dense retrieval. Concretely, we generate a reranking query that includes additional conversational background, which provides more context for relevance judgment.

We also experimented with decomposing the rewritten query into a small set of sub-queries. Each sub-query retrieves a candidate set of passages, and the union of candidates is then reranked to produce the final evidence set. However, this strategy did not yield consistent improvements on the development set and was therefore excluded from the final submission.

3.2 LoRA-Adapted Generator (Subtask B)

For generation, we adopt a decoder-only model and apply parameter-efficient fine-tuning with LoRA (Hu et al., 2021). The generator is trained via supervised fine-tuning (SFT) on the provided training data. At inference time, the generator conditions on the current user query, a short conversational context window (one to two previous turns), and the retrieved evidence passages to produce the final response.

We also experimented with training the generator using single-reference passages (Liu et al., 2024). However, this setting encourages the model to synthesize answers even when the evidence is incomplete or noisy, which conflicts with our focus on faithfulness. Therefore, we did not use this strategy in the final submission.

3.2.1 Answerability Classifier

We explore an auxiliary four-way answerability classifier to predict ANSWERABLE, PARTIAL, UNANSWERABLE, and CONVERSATIONAL turns. We experiment with both a RoBERTa-based classifier (Liu et al., 2019) and a LoRA-adapted decoder model, trained using cross-entropy loss on the provided training data.

However, development results show that the classifier is strongly biased toward the majority ANSWERABLE class, leading to frequent misclassification of PARTIAL cases. This imbalance degrades overall performance under the official lexical-overlap metric in Subtask B. Detailed confusion statistics are provided in Appendix A. Therefore, the classifier is not included in the final submission.

3.3 End-to-End RAG Pipeline (Subtask C)

The full pipeline integrates query rewriting, dense retrieval, reranking, and generation. Given a multi-turn dialogue, we first rewrite the current query, retrieve and rerank evidence passages, and then generate a response conditioned on the selected evidence. We evaluated different retrieval–reranking combinations and found that retrieving the top-50 passages and reranking to the top-5 passages achieved the best end-to-end performance on the development set.

4 Experimental Setup

We implement our system using Qwen3 embedding and generation models. Hyperparameters and prompt templates are described in the following subsections.

4.1 Data and Evaluation Methodology

We use the official dataset released by the task organizers. For Subtask A, retrieval quality is evaluated using the official ranking-based metrics nDCG@$\{1, 3, 5, 10\}$ and Recall@$\{1, 3, 5, 10\}$.

For Subtasks B and C, the official final score is computed as the harmonic mean of three metrics: RB_alg, RB_llm, and RL_F (Katsis et al., 2025). Due to limited computational and storage resources, we primarily rely on RB_alg for model selection and ablation comparisons on the development set, and report results from a single training run without multiple runs with different random seeds, leaving variance analysis for future work.

4.2 Model and Implementation

Dense retrieval and indexing. For retrieval, we pre-compute dense embeddings for all corpus passages and index them using FAISS (Douze et al., 2025) for efficient nearest-neighbor search.[2] At inference time, rewritten queries are encoded using the same embedding model, and top-k candidate passages are retrieved from the index. A BERT-based cross-encoder reranker (Nogueira and Cho, 2019b) is applied to select the final evidence passages for downstream generation, using the cross-encoder/ms-marco-MiniLM-L12-v2 model.[3]

Query rewriting prompts. We implement pronoun rewriting, retrieval query rewriting, and reranking query rewriting using prompt-based generation. The exact prompt templates are shown in Figure 1.

Generator fine-tuning. For Subtask B, we perform LoRA fine-tuning using the PEFT library.[4] We apply LoRA to the attention projection matrices with rank $r = 32$, $\alpha = 64$, and dropout 0.1, targeting q_proj, k_proj, v_proj, and o_proj. We train the generator using supervised fine-tuning with a maximum sequence length of 4096. To train the model in a causal language modeling setup, we construct each training instance using the chat template and mask the loss on the prompt tokens. Concretely, we compute the prefix length of all messages excluding the final assistant response, and set the corresponding label positions to -100 so that the loss is only computed on the target answer tokens.

We use HuggingFace Transformers and the Trainer API for training.[5] The training hyperparameters are: batch size 2 per device, gradient accumulation steps 4, learning rate 5×10^{-5}, 2 epochs, and optimization uses the default AdamW optimizer in Transformers.

Generator prompting. We use the same instruction prompt during fine-tuning and inference to encourage evidence-grounded and concise responses. The generator is instructed to answer using the provided reference documents and dialogue history, produce a 1–2 sentence response, and output "I don't know." when no evidence is available.

The input to the generator follows a structured format that concatenates REFERENCE, HISTORY, and the current QUESTION. The REFERENCE field contains the top-k reranked passages, while HISTORY contains a short conversational context window.

End-to-end pipeline. For Subtask C, we directly integrate the retrieval and generation components described in Section 3 without additional architectural modifications.

5 Results

The official test set results are summarized in Table 2. Our system achieves competitive performance across all three subtasks, outperforming the strongest official baselines in both retrieval and generation.

In Subtask A, our dense-only retrieval framework ranks 12th out of 38 submissions and surpasses the strongest official baseline, a retrieval system built on ELSER with large-model query rewriting. Although we do not incorporate sparse or hybrid components, the integration of history-aware pronoun rewriting, dialogue-conditioned standalone reformulation, and context-enriched reranking leads to stronger retrieval effectiveness.

Controlled comparisons in Appendix B (Table 4) show that removing history-aware query rewriting consistently degrades retrieval performance across models (e.g., Qwen2.5-7B: NDCG@10 drops from 0.4466 to 0.4287). Furthermore, Table 5 shows that this improvement is robust across different backbone models, indicating that structured dialogue-aware query rewriting is broadly beneficial for dense retrieval in multi-turn settings.

In Subtask B, our LoRA-adapted open-source model ranks 15th out of 26 submissions and outperforms the substantially larger gpt-oss-120b baseline. We conduct ablation experiments to compare different generator configurations, and Qwen3-14B with LoRA achieves the best development performance (see Appendix B). These results indicate

[2] https://github.com/facebookresearch/faiss

[3] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L12-v2

[4] https://github.com/huggingface/peft

[5] https://github.com/huggingface/transformers

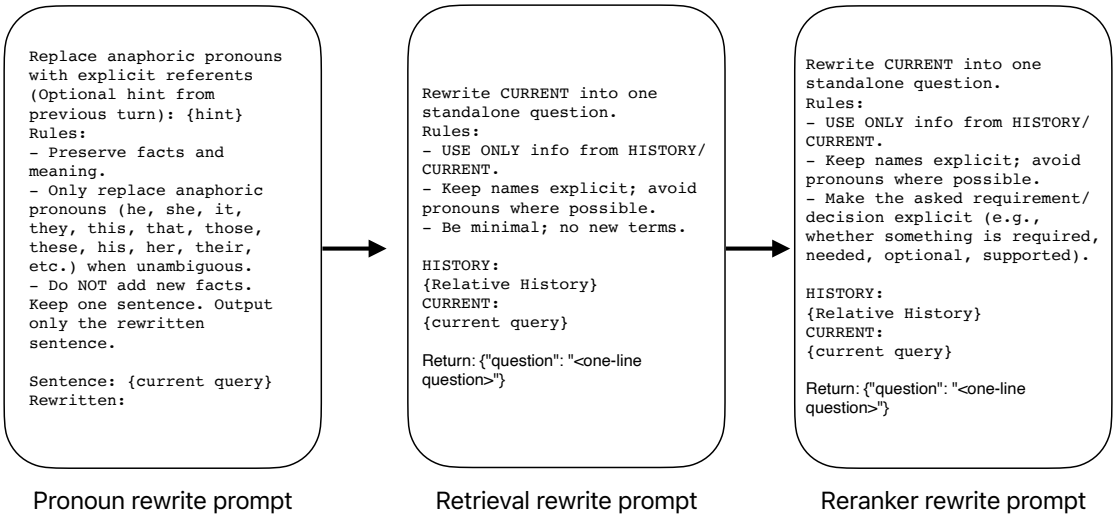

Figure 1: Prompt templates used for pronoun rewriting, retrieval query rewriting, and reranking query rewriting in Subtask A.

that parameter-efficient fine-tuning can yield competitive generation quality without relying on large proprietary models.

In Subtask C, the full RAG pipeline ranks 16th out of 29 submissions. Our performance is close to the strongest official baseline (qwen-30b-a3b-thinking). The remaining gap likely reflects limitations in retrieval accuracy and evidence selection, as well as differences in model capacity and reasoning-oriented architecture, given that the strongest baseline employs a larger model optimized for reasoning.

6 Conclusion

In this paper, we present a modular system for the MTRAG benchmark, consisting of history-aware query rewriting and dense retrieval with reranking (Subtask A), a LoRA-adapted generator (Subtask B), and an integrated end-to-end RAG pipeline (Subtask C). The system is built on Qwen-based embedding and generation models for multi-turn retrieval and response generation. For retrieval, we design a dense-only pipeline that integrates pronoun rewriting, dialogue-conditioned query reformulation, and cross-encoder reranking. This structure improves retrieval in multi-turn settings by modeling conversational dependencies before evidence selection. For generation, our LoRA-adapted open-source model achieves competitive performance relative to larger proprietary baselines, showing that parameter-efficient fine-tuning is sufficient for strong results. The end-to-end RAG pipeline combines these components within a unified framework. For future work, we plan to improve retrieval precision and evidence selection and explore reasoning-oriented generation strategies for more robust multi-turn RAG.

Limitations

Our system has several limitations. First, error propagation in the end-to-end RAG pipeline remains a primary challenge. Inaccuracies in query rewriting or retrieval can lead to partially relevant or missing evidence, which directly affects downstream generation quality. Since the generator conditions entirely on the retrieved passages, failures in early stages are difficult to recover from.

Second, although LoRA-based fine-tuning improves faithfulness, the generator is not explicitly optimized for multi-step reasoning across multiple passages. Complex queries that require aggregating dispersed evidence or integrating dialogue constraints may therefore result in incomplete or overly generalized responses.

Finally, due to computational constraints, model selection and ablation analyses rely primarily on RB_alg on a relatively small development split. While this provides practical guidance, it may not fully capture performance variations under the complete official evaluation metrics.

Task	Metric	Our System	Top Baseline
Subtask A (Retrieval)	nDCG@5	**0.4855**	0.4795 (ELSER + GPT-OSS-20B Rewrite)
Subtask B (Generation)	Harmonic Mean	**0.6554**	0.6390 (gpt-oss-120b)
Subtask C (RAG)	Harmonic Mean	0.5159	0.5366 (qwen-30b-a3b-thinking)

Table 2: Official test set results for all subtasks.

Acknowledgements

We sincerely thank Çağrı Çöltekin for his valuable guidance and support.

References

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

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

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2025. The faiss library. *Preprint*, arXiv:2401.08281.

Elastic. 2023. Elastic learned sparse encoder (elser). https://www.elastic.co/guide/en/elasticsearch/reference/current/semantic-search-elser.html. Accessed: 2026.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation. *CoRR*, abs/2202.03629.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. *Preprint*, arXiv:2004.04906.

Yannis Katsis, Sara Rosenthal, Kshitij Fadnis, Chulaka Gunasekara, Young-Suk Lee, Lucian Popa, Vraj Shah, Huaiyu Zhu, Danish Contractor, and Marina Danilevsky. 2025. Mtrag: A multi-turn conversational benchmark for evaluating retrieval-augmented generation systems. *Transactions of the Association for Computational Linguistics*, 13:784–808.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *CoRR*, abs/2005.11401.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Ziwei Liu, Liang Zhang, Qian Li, Jianghua Wu, and Guangxu Zhu. 2024. Invar-rag: Invariant llm-aligned retrieval for better generation. *arXiv preprint arXiv:2411.07021*.

Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2016. Learning to match using local and distributed representations of text for web search. *Preprint*, arXiv:1610.08136.

Rodrigo Nogueira and Kyunghyun Cho. 2019a. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Rodrigo Nogueira and Kyunghyun Cho. 2019b. Passage re-ranking with BERT. *CoRR*, abs/1901.04085.

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

S. Robertson. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Sara Rosenthal, Yannis Katsis, Vraj Shah, Lihong He, Lucian Popa, and Marina Danilevsky. 2026. Mtrag-un: A benchmark for open challenges in multi-turn rag conversations. *Preprint*, arXiv:2602.23184.

Chaitanya Sharma. 2025. Retrieval-augmented generation: A comprehensive survey of architectures, enhancements, and robustness frontiers. *arXiv preprint arXiv:2506.00054*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

A Answerability Classifier Error Analysis

Table 3 shows the confusion matrix of the four-way answerability classifier on the development set. The classifier exhibits a strong bias toward the majority ANSWERABLE class, predicting it for most instances. In particular, all PARTIAL instances (23/23) are misclassified as ANSWERABLE, resulting in zero recall for this category. Although UNANSWERABLE cases are correctly identified (8/8), the extremely small support for minority classes leads to a macro F1 score of 0.45 despite an overall accuracy of 0.83. This imbalance explains the degradation in RB_alg observed when incorporating the classifier into the generation pipeline.

Gold \ Pred	Ans.	Part.	Unans.	Conv.
Ans.	133	3	0	0
Part.	23	0	0	0
Unans.	0	0	8	0
Conv.	0	0	2	0

Table 3: Confusion matrix of the four-way answerability classifier on the development set (rows = gold labels, columns = predicted labels).

B Model and Strategy Selection

For Subtask A, we conduct additional comparisons to analyze the effect of history-aware query rewriting on retrieval performance. Specifically, we compare retrieval results with and without history selection under otherwise identical settings. The results (Table 4) show that removing rewriting leads to consistent drops in retrieval performance across models. This observation motivates the use of history-aware rewriting in our final retrieval pipeline. We further compare different backbone models for history-aware query rewriting (Table 5). The results show that rewriting performance is relatively stable across models, with Qwen3-30B achieving the best overall retrieval effectiveness.

For Subtasks B and C, we partition the official training data into 673 training instances and 169 development instances (approximately an 80/20 split) for model and strategy selection. Due to limited computational and storage resources, we use RB_alg on the development set as the primary criterion for comparing configurations.

Table 6 summarizes the configurations evaluated to select the final generator (Subtask B) and the retrieval–reranking setup (Subtask C). For Subtask B, we compare several generator variants. "Oversampling" increases the proportion of minority answerability types during training. "Single Ref" trains the generator using only one reference passage per instance. "Inference Prompt" introduces additional instruction constraints at inference time. We also experiment with integrating an auxiliary answerability classifier (either LoRA-adapted or RoBERTa-based) into the generation pipeline. For Subtask C, we compare different retrieval–reranking configurations by varying the number of retrieved candidates before reranking and the final number of passages provided to the generator. This analysis examines the trade-off between retrieval recall and evidence precision in the end-to-end RAG pipeline.

Model	N@1	N@3	N@5	N@10	R@1	R@3	R@5	R@10
Qwen2.5-7B (w/ rewrite)	0.394	**0.367**	0.388	0.447	0.151	**0.339**	0.414	0.552
Qwen2.5-7B (w/o rewrite)	0.375	0.348	0.381	0.429	0.144	0.319	0.418	0.528
Qwen3-14B (w/ rewrite)	**0.418**	0.361	**0.397**	**0.458**	**0.154**	0.326	**0.431**	**0.576**
Qwen3-14B (w/o rewrite)	0.413	0.365	0.393	0.451	**0.154**	0.327	0.409	0.553

Table 4: Effect of history-aware query rewriting on retrieval performance (ClapNQ).

Model	N@1	N@3	N@5	N@10	R@1	R@3	R@5	R@10
Mixtral 8x7B	0.409	0.370	0.407	0.457	0.160	0.347	**0.451**	0.565
Qwen3-14B	0.418	0.361	0.397	0.458	0.154	0.326	0.431	**0.576**
Qwen3-30B	0.423	**0.389**	**0.410**	**0.463**	0.160	**0.360**	0.433	0.563
Gemma-12B	**0.428**	0.368	0.395	0.456	**0.171**	0.325	0.412	0.554

Table 5: Comparison of different models for history-aware query rewriting on ClapNQ.

Subtask B: Generator Selection	
Method	Dev RB_alg
Baseline	0.361
Qwen2.5-7B + LoRA	0.423
Qwen2.5-7B + LoRA + Chat mode	0.425
Qwen3-14B	0.451
Qwen3-14B + LoRA	**0.526**
Qwen3-14B + LoRA (Oversampling)	0.516
Qwen3-14B + LoRA (Single Ref)	0.517
Qwen3-14B + LoRA + Inference Prompt	0.518
Qwen3-14B + LoRA + Classifier (LoRA)	0.383
Qwen3-14B + LoRA + Classifier (RoBERTa)	0.493
Subtask C: Retrieval–Reranking Strategy Selection	
Method	Dev RB_alg
Retriever + Generator (Baseline)	0.384
Retriever (top-10) + Reranker (top-5) + Generator	0.421
Retriever (top-50) + Reranker (top-5) + Generator	**0.435**

Table 6: Model and strategy selection on the development set using RB_alg for Subtasks B and C.