A Multi-Agent Approach with Similarity-Guided Judgment for Scientific Knowledge Retrieval and Answer Generation

Notebook for LongEval at CLEF 2026

Zhenyang Cheng, Haoliang Qi* and Yusheng Yi

Foshan University, Foshan, China

Abstract

The LongEval-RAG task evaluates the capability of Retrieval-Augmented Generation systems to produce answers under document-constrained conditions when scientific knowledge undergoes temporal evolution. This work proposes a multi-agent Retrieval-Augmented Generation approach that integrates semantic similarity filtering, document relation scoring, and a verification-based rollback mechanism. Specifically, the method retains candidate documents by fusing semantic similarity scores with document filter judgments, subsequently ranks candidates by relevance while analyzing and recording inter-document contradictions, temporal evolution, and unique informational contributions, and finally employs semantic similarity metrics alongside an answer verifier to determine whether rollback-triggered regeneration is warranted. Experimental results demonstrate that this workflow effectively mitigates the erroneous exclusion of highly relevant documents, achieving an 18.1% improvement in Precision over the single-model baseline and attaining a Precision score of 0.95. The accompanying source code for this paper is at https://github.com/shock-cheng/clef26-lab-individual.git

Keywords

Retrieval-Augmented Generation, Multi-agent workflow, Semantic similarity filtering, LongEval-RAG

1. Introduction

Retrieval-Augmented Generation (RAG) relies on external knowledge bases to enhance the credibility of generated answers [1]. However, given the continuous evolution of scientific knowledge, static RAG pipelines struggle to adequately handle document obsolescence, conflicting viewpoints, and temporal concept replacement. The CLEF 2026 LongEval Lab introduced the LongEval-RAG task to evaluate Retrieval-Augmented Generation systems under document-constrained settings with temporally evolving scientific knowledge [2, 3, 4]: systems must generate cited natural language answers based exclusively on the provided documents, given a query and a set of candidate document IDs. The evaluation protocol simultaneously assesses answer generation quality (measured by ROUGE and BERTScore) alongside the precision and recall of document filtering. The core difficulty of this task stems from the pronounced temporal evolution and logical contradictions among documents, while existing filtering and generation processes lack explicit mechanisms to address these dynamics.

As LongEval-RAG is introduced for the first time, no public baseline systems or established methodologies are currently available. We initially constructed a minimal single-model baseline by concatenating the query with all candidate documents and feeding them directly into a large language model (LLM), which simultaneously performs document filtering and answer generation. Empirical analysis revealed two critical failure modes: in 17% of queries, the model incorrectly classified all documents as irrelevant, causing the subsequent answer generation to completely lose its factual grounding; in 10% of queries, semantically highly relevant documents were erroneously excluded. Further investigation indicates that single-model document filtering lacks a rollback mechanism, making it highly susceptible to the inadvertent removal of critical evidence. Furthermore, the model treats documents as isolated fragments without accounting for inter-document contradictions, temporal concept replacement, or information

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

*Corresponding author.

cheng_shock@foxmail.com (Z. Cheng); haoliangqi163@163.com (H. Qi); yiys@fosu.edu.cn (Y. Yi)

0009-0008-6262-5790 (Z. Cheng); 0000-0003-1321-5820 (H. Qi); 0009-0006-7098-3681 (Y. Yi)

complementarity [5]. Consequently, generated answers frequently omit the evolutionary context of the knowledge or incorporate outdated information. Additionally, the baseline lacks a verification stage, meaning any content unfaithful to the provided documents cannot be automatically corrected.

To address these observed limitations, this work proposes a multi-stage workflow comprising four sequential phases: document filtering, relation scoring and ranking, answer generation, and verification and rollback [6]. During the document filtering phase, the system computes the union of documents selected via a dynamic semantic similarity threshold and those identified by the LLM, thereby forcibly retaining highly relevant documents and mitigating the risk of false exclusions. In the relation scoring and ranking phase, the model evaluates the retained documents across three dimensions—inter-document contradictions, temporal concept replacement, and unique informational contributions—assigning scores on a 0–10 scale. Documents receiving a score of zero are discarded, while the remainder are sorted in descending order. The model also outputs relational explanations, providing structured evolutionary cues for subsequent generation. The answer generation phase strictly constrains the model to formulate responses based exclusively on the ranked documents, with mandatory citation markers. Finally, the verification and rollback phase conducts a multi-dimensional assessment by measuring answer-query semantic similarity, answer-document similarity, and consistency with the relational explanations. If the output fails to satisfy the predefined criteria, the system generates targeted revision feedback and rolls back to the answer generation phase for re-computation, thereby establishing a self-correcting closed loop.

The complete implementation, prompts, and experimental configurations are released in our public repository to facilitate reproducibility.

2. Methodology

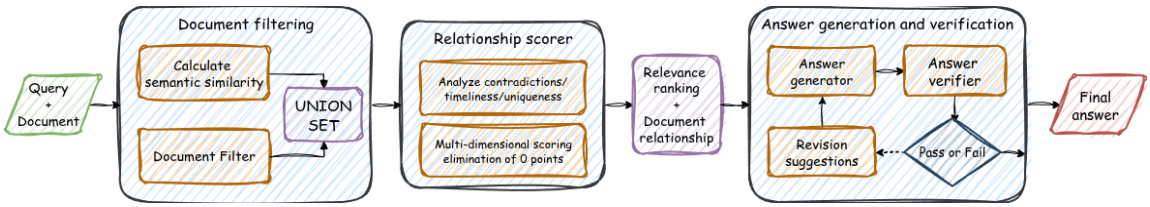

Figure 1: Overview of the proposed multi-agent RAG pipeline.

This work proposes a multi-stage Retrieval-Augmented Generation pipeline that sequentially executes four phases: document filtering, relation scoring and ranking, constrained answer generation, and answer verification with conditional rollback. Given a query q and a candidate document set $D = \{d_1, d_2, \ldots, d_n\}$, the system outputs a final answer a along with the corresponding ranked document subset D_{sort}.

2.1. Document Filtering

The document filtering phase aims to eliminate query-irrelevant documents from D, producing a refined candidate subset D_{filt}.

Initially, by using a large language model extracts domain-specific keywords and technical terminology from q to construct a keyword set kw. The large language model first infers the scientific domain of the input q and then extracts domain-specific keywords and technical terminology related to that domain. The extracted keyword set is concatenated with the original query to construct the expanded query. The expanded query $q' = q + kw$ is then formed by concatenating the original query with these keywords, thereby broadening the coverage of subsequent semantic matching.

Next, a text embedding model vectorizes the expanded query and each candidate document to

compute pairwise cosine similarity [7]:

$$s_i = \cos\big(E(q'), E(d_i)\big), \quad i = 1, \ldots, n$$

This yields a similarity set $S = \{s_1, s_2, \ldots, s_n\}$, which is subsequently sorted in descending order. The specific embedding architecture employed is detailed in the experimental setup.

To mitigate the risk of excluding highly relevant documents due to a rigid absolute threshold, a distribution-aware dynamic three-tier strategy determines the global threshold θ:

$$\theta = \begin{cases} 0.35, & \text{if } |\{s_i \mid s_i \geq 0.35\}| \geq 2 \\ 0.20, & \text{otherwise, if } \exists s_i \geq 0.20 \text{ among the top-5 documents} \\ 0.10, & \text{otherwise} \end{cases}$$

This policy preferentially applies a stringent threshold to retain strongly relevant documents. Should fewer than two documents meet this criterion, the threshold relaxes to 0.20 to capture moderately relevant candidates. In the absence of qualifying documents, a fallback threshold of 0.10 is applied to retain weakly relevant entries, thereby guaranteeing a non-empty filtered set.

A semantic mask $M_{\text{sim}} \in \{0, 1\}^n$ is then derived based on θ, defined as $M_{\text{sim}}[i] = \mathbb{I}(s_i \geq \theta)$.

Concurrently, the expanded query q' and the full document set D are processed by a dedicated document filter. This model evaluates the relevance of each document individually, yielding a binary relevance mask $M_{\text{llm}} \in \{0, 1\}^n$. Implementation details for the document filter are provided in the experimental setup.

Finally, a bitwise logical OR operation fuses the two masks:

$$M_{\text{filt}} = M_{\text{sim}} \vee M_{\text{llm}}$$

The resulting filtered document subset is defined as:

$$D_{\text{filt}} = \{d_i \in D \mid M_{\text{filt}}[i] = 1\}$$

This union-based strategy forcibly retains documents exceeding the semantic similarity threshold, even if erroneously flagged as irrelevant by the LLM filter. Simultaneously, it leverages the document filter's deep semantic comprehension to recover long-tail relevant documents that pure similarity metrics might overlook.

2.2. Relation Scoring and Ranking

The query q, keyword set kw, filtered subset D_{filt}, and corresponding similarity scores are fed into a relation scorer. This module analyzes inter-document and document-query relationships according to three guiding principles [8]:

1. **Contradiction Detection:** Conflicting documents are both retained, with explicit contradiction markers and individual relevance scores assigned.
2. **Temporal Awareness:** Documents reflecting early concepts superseded by subsequent research are retained and annotated with their respective temporal periods.
3. **Information Complementarity:** Documents containing unique information absent from other candidates are retained, with explicit annotation of their relevance pathways to the query.

Building upon this analysis, the relation scorer assigns an integer relevance score $r_i \in [0, 10]$ to each document $d_i \in D_{\text{filt}}$, adhering to strictly mutually exclusive criteria:

- **0 (Obsolete/Irrelevant):** Content is entirely unrelated to the query or is completely redundant, with all information fully covered by another document.
- **1–3 (Weak Relevance):** Indirect connection to certain aspects of the query.

- **4–6 (Moderate Relevance):** Covers key aspects of the query and directly addresses partial components.
- **7–10 (Strong Relevance):** Provides a direct and comprehensive response to the query.

Documents assigned $r_i = 0$ are discarded. The remaining entries are sorted in descending order of their scores to form the ranked document set D_{sort}. Concurrently, the relational explanation text R generated by the scorer is preserved. Implementation specifics for the relation scorer are detailed in the experimental setup.

2.3. Answer Generation

The query q, keyword set kw, ranked document set D_{sort}, relational explanations R, and optional revision feedback (originating from the verification phase) are fed into the answer generator. The generation process operates under the following strict constraints:

- The response must be derived exclusively from the content within D_{sort}, with zero tolerance for external knowledge injection.
- The generator must actively leverage contradiction markers, concept replacement annotations, and unique information pathways specified in R to structure the response, explicitly distinguishing and clarifying conflicting or temporally divergent content.

During the initial generation pass, the revision feedback is null. If regeneration is triggered by a rollback mechanism, the answer generator incorporates the feedback as additional contextual guidance for correction. The specific architecture and hyperparameters of the answer generator are provided in the experimental setup.

2.4. Verification and Rollback

To mitigate answer hallucination and prevent the omission of critical evidence, a verification and conditional rollback mechanism is integrated into the pipeline.

Initially, the same text embedding model employed in the document filtering phase computes:

- The semantic similarity between the answer and query, $\text{sim}(q, a)$;
- The similarity set between the answer and each document in D_{sort}, $\text{sim}_D = \{\text{sim}(a, d_j) \mid d_j \in D_{\text{sort}}\}$.

These metrics, along with q, a, D_{sort}, and R, are subsequently passed to an answer verifier. The verifier evaluates compliance against four criteria [9]:

1. Topical alignment between the answer and the query.
2. Strict adherence to D_{sort}, ensuring no external knowledge is introduced.
3. Identification of any unused documents within D_{sort}, with an assessment of whether their omission is justified.
4. Consistency between the generated answer and the relational explanations outlined in R.

The answer verifier outputs a binary label $v \in \{\text{pass}, \text{fail}\}$.

- If $v = \text{pass}$, the pipeline terminates, yielding the final answer a and ranked document set D_{sort}.
- If $v = \text{fail}$, the verifier simultaneously generates targeted revision feedback f. The control flow then rolls back to the constrained answer generation phase, where the generator reformulates the answer incorporating f alongside the original context, before re-entering the verification stage. To prevent infinite loops, the maximum number of rollback iterations is capped at three. Should the verification fail after exhausting this limit, the system outputs the most recent generated answer and corresponding document set, appended with a verification failure flag.

3. Experiments

3.1. Experimental Setup

3.1.1. Dataset

All experiments are conducted on the official test set `longeval_sci_2026/clef_2026/rag` provided by the LongEval-RAG task [10]. As the task does not release training or validation splits, the system performs direct inference on this test set. Each sample consists of a text query q and its corresponding candidate document set $D = \{d_1, d_2, \ldots, d_n\}$.

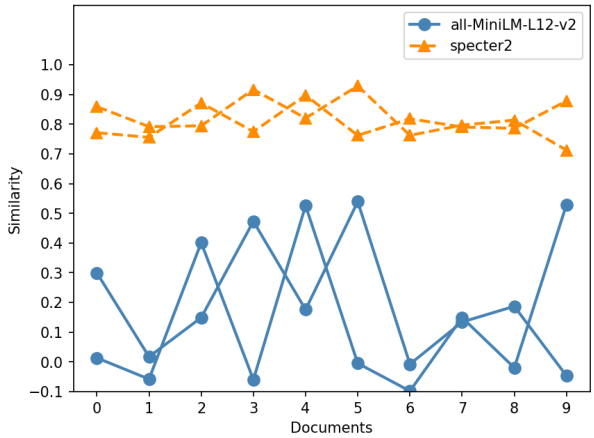

Figure 2: The difference in similarity between the two models

3.1.2. Models and Inference Parameters

Each experiment consistently employs a single LLM across all agents (document filter, relation scorer, answer generator, and answer verifier). Different experiments compare Qwen3.6-35B-A3B[1] and DeepSeek-V4-Pro[2] under the same workflow. Temperature parameters are customized across pipeline stages, as detailed in Table 1. Semantic similarity computation uniformly utilizes the general-purpose embedding model `sentence-transformers/all-MiniLM-L12-v2` [3] [11]. We select this model over the domain-specific `allenai/specter2` [4] because, in scenarios matching short queries against long documents, the general-purpose model yields a more discriminative similarity distribution. As illustrated in Figure 2, the general-purpose embedding model exhibits more pronounced variation in its similarity distributions, whereas `specter2` yields consistently inflated scores with limited discriminative range under this scenario, complicating effective threshold calibration.

3.1.3. Evaluation Metrics

We adopt the official automated metrics provided by the task: ROUGE-L [12], BERTScore [13], Precision (document retrieval precision), Nugget Coverage, and TFC1 (comprehensive textual faithfulness metric) [14]. According to the official LongEval-RAG evaluation results, document-level Recall is not reported. Since the task operates under a strict blind evaluation setting with withheld ground-truth labels, computing traditional Recall locally is infeasible. Instead, the official evaluation adopts Nugget Coverage[15] to assess retrieval comprehensiveness. By measuring the proportion of key information points (nuggets) captured in the answers, it evaluates information completeness at a finer granularity,

[1] https://github.com/QwenLM/Qwen3.6
[2] https://platform.deepseek.com/
[3] https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2
[4] https://huggingface.co/allenai/specter2_base

effectively substituting the evaluative role of Recall in RAG scenarios. Additionally, to characterize system behavior, we record the following internal observational metric:

- *Token Consumption*: Average per-sample token usage for each stage and the full pipeline.

3.1.4. Experimental Design

To systematically evaluate the impact of component overheads and parameter variations within the multi-agent workflow, we establish five experimental configurations, encompassing multi-agent variants and a single-model baseline. Table 1 summarizes the setups.

Table 1
Experimental Configuration Summary

Exp	Model	Workflow Type	Similarity-Aided Verification	Temperature
Exp 1	Qwen3.6-35B-A3B	Multi-agent	Filtering only	0.5 / 0.5 / 0.5 / 0.5
Exp 2	Qwen3.6-35B-A3B	Multi-agent	Filtering + Rollback	0.1 / 0.1 / 0.35 / 0.1
Exp 3	DeepSeek-V4-Pro	Multi-agent	Filtering only	0.1 / 0.1 / 0.35 / 0.1
Exp 4	DeepSeek-V4-Pro	Multi-agent	Filtering + Rollback	0.1 / 0.1 / 0.35 / 0.1
Exp 5	DeepSeek-V4-Pro	Single-model	None	0.5 (uniform)

Experiment 1 (multi-agent-rag) adopts the multi-agent workflow but omits semantic similarity computation in the verification step; the answer verifier relies solely on textual cues to decide whether to trigger a rollback. All agents utilize Qwen3.6-35B-A3B with a uniform temperature of 0.5. This configuration isolates Qwen's multi-agent behavior under higher stochasticity.

Experiment 2 (multi-agent-rag-v4) shares the same pipeline as Experiment 1 but applies a differentiated temperature scheme: 0.1 for the document filter, relation scorer, and answer verifier, and 0.35 for the answer generator. This setup creates a symmetric comparison with Experiment 4, enabling direct analysis of how different models behave within the identical low-temperature strategy.

Experiment 3 (multi-agent-rag-v2) replaces all components with DeepSeek-V4-Pro while retaining the temperature scheme from Experiment 2 (0.1 for filtering/scoring/verification, 0.35 for generation). This experiment investigates the performance of a higher-capacity model under low temperature and in the absence of quantitative similarity signals.

Experiment 4 (multi-agent-rag-v3) extends Experiment 3 by integrating semantic similarity computation into the verification step. It calculates $\mathrm{sim}(q, a)$ and the similarity set sim_D between the answer and ranked documents, feeding both as auxiliary signals into the answer verifier. Upon a failure verdict, the verifier generates targeted revision feedback and trigger a rollback to the answer generator for regeneration. Models and temperatures match Experiment 3.

Experiment 5 (single-model-rag-v1) serves as a baseline to quantify token overhead differences between multi-agent and single-model paradigms, and to assess the impact of pipeline simplification on generation quality. The query q and full candidate set D are directly fed into a single LLM, which simultaneously handles document relevance ranking and answer generation, bypassing pre-filtering, relation scoring, and verification/rollback. DeepSeek-V4-Pro is used with a uniform temperature of 0.5. This configuration acts as a reference point for resource expenditure and answer conciseness.

3.1.5. Parameter Specifications

Semantic Similarity Model and Thresholds We first compute the semantic similarity between the expanded query $(q + kw)$ and each document in D using `all-MiniLM-L12-v2`. Based on the resulting similarity distribution illustrated in Figure 3, a three-tier dynamic thresholding strategy is applied for preliminary filtering. A primary threshold of 0.35 is enforced to retain highly similar documents, corresponding to the typical similarity range of the top-ranked items in the distribution. If fewer than two documents satisfy this condition, the top-5 documents ranked by similarity are taken and the threshold is relaxed to 0.20 so as to cover the suboptimal candidate tier. Should all top-5 documents

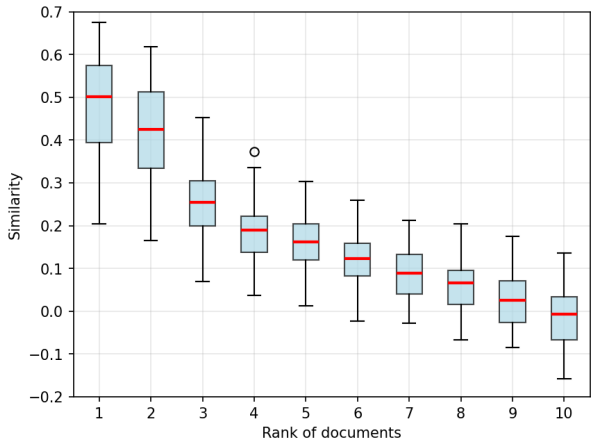

Figure 3: The trend of document similarity after reverse sorting

still fall below 0.20, the threshold is further lowered to 0.10 to capture marginal yet comparatively higher-ranked semantic information. Retaining up to five documents in this coarse stage ensures to exclude only blatantly irrelevant candidates, preserving a sufficient pool for the fine-grained relation scorer. The final filtered set D_{filt} is obtained by taking the union of these similarity-based selections and the document filter's judgments. This union mechanism forcibly preserves documents with measurable semantic overlap, structurally preventing the erroneous exclusion of all candidates.

Temperature Configuration Experiments 2–4 apply a low temperature (0.1) to the document filtering, relation scoring, and answer verification stages to maximize determinism and consistency in these discriminative tasks. Conversely, a higher temperature (0.35) is maintained for answer generation to preserve linguistic diversity and fluency. Experiments 1 and 5 employ a uniform temperature of 0.5, serving as a high-stochasticity control.

3.2. Results and Analysis

3.2.1. Overall Results

Table 2 summarizes the performance of all experimental variants and the official naive baseline across all evaluation metrics, alongside average token consumption. Experiment 2 is a supplementary comparison; its evaluation data is currently unavailable and therefore omitted from the table.

Table 2
Evaluation Results and Resource Consumption

Method	Rouge-L	BERTScore	Precision	Nugget Cov.	TFC1	Avg. Tokens/Query
official-naive-baseline	0.082	-0.118	0.200	0.124	-0.758	—
Exp 1	0.158	0.157	0.950	0.505	-0.051	16.0k
Exp 3	0.136	0.113	0.940	0.537	-0.119	11.0k
Exp 4	0.132	0.109	0.943	0.504	-0.128	11.3k
Exp 5	0.155	0.138	0.769	0.473	-0.082	5.3k

3.2.2. Significant Improvements in Retrieval Performance

All multi-agent variants (Experiments 1–4) substantially outperform both the single-model baseline and the official naive baseline on Precision and Nugget Coverage. Precision increases from 0.769 in the single-model setting to 0.94–0.95, while Nugget Coverage rises from 0.473 to 0.50–0.54. These gains confirm that the phased filtering strategy, coupled with the union-based retention mechanism,

effectively identifies and preserves relevant documents, thereby enhancing retrieval accuracy and information coverage. Notably, Experiment 3 achieves the highest Nugget Coverage (0.537), indicating that DeepSeek-V4-Pro, when configured with low temperatures, exhibits superior capabilities in fine-grained document scoring.

3.2.3. Generation Quality Does Not Scale with Pipeline Complexity

On generation-oriented metrics (ROUGE-L and BERTScore), Experiment 1 (ROUGE-L: 0.158, BERTScore: 0.157) performs comparably to Experiment 5 (ROUGE-L: 0.155, BERTScore: 0.138), showing marginal gains. However, Experiment 3 (ROUGE-L: 0.136, BERTScore: 0.113) and Experiment 4 (ROUGE-L: 0.132, BERTScore: 0.109) underperform relative to Experiment 5. This degradation indicates that while lower generation temperatures and stricter constraints improve retrieval precision, they may inadvertently constrain the answer generator's expressiveness and fluency, leading to a decline in surface-level semantic alignment with reference answers. The TFC1 metric mirrors this trend: Experiment 1 scores -0.051, surpassing Experiment 5's -0.082, whereas Experiments 3 (-0.119) and 4 (-0.128) fall below the baseline.

3.2.4. Impact of Model Selection

Experiment 1 achieves the highest generation quality among multi-agent configurations (ROUGE-L: 0.158, BERTScore: 0.157) but incurs the highest token overhead (16.0k per query). Switching to DeepSeek-V4-Pro in Experiments 3 and 4 reduces token consumption significantly to 11.0k–11.3k, albeit at the cost of generation quality. This trade-off suggests that Qwen3.6-35B-A3B possesses stronger answer generation capabilities within the identical multi-agent framework, though at lower computational efficiency. Conversely, DeepSeek-V4-Pro offers superior efficiency but yields comparatively lower generation quality in this specific task context.

3.2.5. Effectiveness of the Rollback Mechanism

Experiment 4 introduces similarity-aided verification, triggering rollback regeneration for 17% of queries. The answer verifier successfully leverages quantitative similarity signals to detect inconsistent outputs and initiate corrections. However, final metrics reveal that Experiment 4 underperforms Experiment 3 in both ROUGE-L (0.132 vs. 0.136) and BERTScore (0.109 vs. 0.113), with Nugget Coverage dropping from 0.537 to 0.504. This indicates that the current rollback mechanism primarily functions as a corrective constraint for non-compliant outputs rather than a direct enhancer of generation quality. While rollback successfully rectifies certain document-inconsistencies, the regeneration process may introduce structural or stylistic degradation, resulting in a slight decline in surface-level semantic metrics.

3.2.6. Token Consumption Analysis

The multi-agent workflow incurs approximately 2–3$\times$ the token overhead of the single-model baseline. Experiment 1 exhibits the highest consumption (16.0k), while Experiments 3 and 4 require 11.0k and 11.3k, respectively. The primary computational bottleneck lies in the initial document filtering stage, which processes the query alongside the entire candidate document set, resulting in maximal input length. Post-filtering, the reduced document count substantially lowers the computational load for subsequent modules. Experiment 4's token usage increases marginally by 0.3k over Experiment 3 due to the 17% rollback rate. Collectively, the additional overhead of the multi-agent pipeline is primarily exchanged for substantial gains in retrieval performance (Precision improving from 0.769 to $>$0.94). If the primary objective prioritizes retrieval accuracy and information coverage, this computational cost is justified. Conversely, if the focus remains solely on answer-level semantic similarity metrics, the single-model baseline already delivers competitive results. Although the multi-agent workflow requires approximately two to three times more tokens than the single-model baseline, the additional

computational cost is mainly exchanged for substantially higher document retrieval precision and information coverage.

4. Conclusion

In Retrieval-Augmented Generation tasks involving temporally evolving scientific knowledge, single-model approaches tend to erroneously discard a substantial proportion of highly relevant documents. Empirical results indicate that incorporating a union-based mechanism—combining semantic filtering with LLM-based relevance judgments—significantly improves document retention, yielding substantial gains in retrieval precision and information coverage relative to the single-model baseline. However, answer-level semantic generation quality does not scale commensurately with increased pipeline complexity. Furthermore, while the similarity-aided rollback mechanism effectively triggers corrective regeneration, its impact on ultimate generation performance remains limited. The four-stage multi-agent workflow proposed in this work demonstrates pronounced advantages on the retrieval side; nevertheless, achieving concurrent improvements in generation quality at lower computational cost remains a critical direction for future optimization.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.62276064).

Declaration on Generative AI

During the preparation of this work, the author(s) used DeepSeek in order to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] P. Lewis, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, in: Advances in Neural Information Processing Systems (NeurIPS), volume 33, 2020, pp. 9459–9474. URL: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

[2] M. Cancellieri, et al., Longeval at clef 2025: Longitudinal evaluation of ir model performance, in: Advances in Information Retrieval (ECIR), 2025, pp. 382–388. doi:10.1007/978-3-031-88720-8_58.

[3] S. Bouaraba, T. Breuer, M. Cancellieri, A. El-Ebshihy, M. Fröbe, P. Galuscáková, L. Goeuriot, G. Iturra-Bocaz, J. Keller, P. Knoth, A. K. Kruff, P. Mulhem, F. Piroi, D. Pride, P. Schaer, D. Schwab, Overview of the clef 2026 longeval lab on longitudinal evaluation of model performance, in: M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, S. MacAvaney, J. M. Struß, E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Lecture Notes in Computer Science (LNCS), Springer, 2026.

[4] S. Bouaraba, T. Breuer, M. Cancellieri, A. El-Ebshihy, M. Fröbe, P. Galuscáková, L. Goeuriot, G. Iturra-Bocaz, J. Keller, P. Knoth, A. K. Kruff, P. Mulhem, F. Piroi, D. Pride, P. Schaer, D. Schwab, Extended overview of the clef 2026 longeval lab on longitudinal evaluation of model performance, in: E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, 2026.

[5] C. Sharma, Retrieval-augmented generation: A comprehensive survey of architectures, enhancements, and robustness frontiers, arXiv preprint (2025). doi:10.48550/arXiv.2506.00054. arXiv:2506.00054.

[6] C.-Y. Chang, et al., Main-rag: Multi-agent filtering retrieval-augmented generation, in: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL), 2025, pp. 2607–2622. doi:10.18653/v1/2025.acl-long.131.
[7] N. Reimers, et al., Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3980–3990. doi:10.18653/v1/D19-1410.
[8] Y. C. Akay, et al., Spd-rag: Sub-agent per document retrieval-augmented generation, arXiv preprint (2026). doi:10.48550/arXiv.2603.08329. arXiv:2603.08329.
[9] X. Xiao, et al., Mass-rag: Multi-agent synthesis retrieval-augmented generation, in: Findings of the Association for Computational Linguistics (ACL), 2026. doi:10.48550/arXiv.2604.18509.
[10] CLEF LongEval Organizers, Longeval lab at clef 2026, 2026. URL: https://clef-longeval.github.io/.
[11] Sentence-Transformers, all-minilm-l12-v2, Hugging Face, 2021. URL: https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2.
[12] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013/.
[13] T. Zhang, et al., Bertscore: Evaluating text generation with bert, in: Proceedings of the 8th International Conference on Learning Representations (ICLR), 2020. URL: https://openreview.net/forum?id=SkeHuCVFDr.
[14] R. Pradeep, et al., Initial nugget evaluation results for the trec 2024 rag track with the autonuggetizer framework, arXiv preprint (2024). doi:10.48550/arXiv.2411.09607. arXiv:2411.09607.
[15] R. Pradeep, N. Thakur, S. Upadhyay, D. Campos, N. Craswell, J. Lin, Initial nugget evaluation results for the trec 2024 rag track with the autonuggetizer framework, 2024. URL: https://arxiv.org/abs/2411.09607. arXiv:2411.09607.

A. Prompts for the Multi-Agent Workflow

A.1. Keyword Extraction Prompt

You are an expert in academic literature. Based on the question. First identify its academic field, then extract the core scientific concepts and technical terms from the question within that field.
Output JSON content: {"keywords":"domain of the question + sequence of keywords or technical terms"}
Output should be a directly parsable JSON string. It is strictly prohibited to use any Markdown block wrapping or comment explanations.

A.2. Document Filter Prompt

You are an expert in academic literature. Based on the question, keywords and references, determine which literature abstracts are relevant to the question.
Analysis task: Decide whether to retain (1) or discard (0) each abstract.
Analysis rules:
1.If the content discussed in the literature is completely unrelated to the question and there is no overlap, mark as 0.
2.If the literature directly answers the question or provides background information or context related to the question, mark as 1.
Output JSON content: {"each references id":"Composed of the numbers 1(retain) and 0(discard)"}
Output should be a directly parsable JSON string. It is strictly prohibited to use any Markdown block wrapping or comment explanations.

A.3. Relation Scoring and Ranking Prompt

You are an expert in academic literature. Based on question, keywords, references, opinion and similarity, analyze and evaluate the relationships between each document and the extent to which the documents help answer the question.
Scoring criteria (0–10 points):
– 0 points: Discarded – The content is irrelevant to the question, does not contain any content related to question and keywords, and cannot answer the question; the content is completely redundant and another document covers all its content as well as more.
– 1–4 points: Weakly related – Related to certain aspects of the question or has an indirect connection.
– 4–7 points: Moderately related – Covers most of the key content and directly addresses the key aspects of the query.
– 7–10 points: Strongly related – Directly and comprehensively answers the question.
Special rules:
1. If two documents are contradictory, keep both documents and give scores based on their respective relevance. At the same time, clearly indicate the contradictions.
2. If a document reflects an earlier concept that has been replaced, keep it and note the time period it belongs to.
3. This document contains unique information not found in other documents, and note its relevance to the question.
Output JSON content: {"correlation":"Annotation information and relevance information, explaining the relationship between the documents and the documents and the question;","score":[6,9,0,4...(Each document corresponding score)]}
Output should be a directly parsable JSON string. It is strictly prohibited to use any Markdown block wrapping or comment explanations.

A.4. Answer Generator Prompt

You are an expert in academic literature. Based on the question, keywords, references, correlation, and opinion, generate a comprehensive and evidence–based answer for the question.
Requirements:
1. The answer must be based on the specific content of the provided literature and cannot use any external information.
2. If the correlation section mentions contradictory and time–evolving information, the answer should include relevant descriptions.
3. Provide a comprehensive but concise answer to the question, without excessive verbosity.
Output the content of the answer directly without including any other characters or annotations.
Note: The id numbers of references are temporary numbers and not the actual IDs. It is prohibited to include these id numbers in the answer.

A.5. Verification and Rollback Prompt

You are an expert in academic literature. Based on question, answer, references, correlation and similarity, verify whether an answer is entirely based on the original source documents it cites.
Review process:
1. Determine if the answer is relevant to the question.
2. Determine if the answer is derived from the provided references without incorporating external knowledge bases.
3. Determine if there are documents provided in the references but not actually used in the answer.
4. Determine if the answer refers to the description in correlation.
5. Determine if the answer is relevant to the references but not to the question, marked as "wrong answer". If the answer is not relevant to the references at all, marked as "wrong filter".
Output JSON content: {"code": "OK (pass) or AS (wrong answer) or FT (wrong filter)", "info": "brief opinion"}
Output should be a directly parsable JSON string. It is strictly prohibited to use any Markdown block wrapping or comment explanations.