GenAIus at RAG4Reports 2026: Citation-Aware Compression for Multilingual Report Generation

Reyyan Yeniterzi
GenAIus Technologies
reyyan@genaius.tech

Suveyda Yeniterzi
GenAIus Technologies
suveyda@genaius.tech

Abstract

This paper describes the GenAIus submission to RAG4Reports 2026 Multilingual Report Generation Task. Our system builds on our earlier TREC RAGTIME pipeline, reusing the evidence preparation stages for overlapping topics, including question generation, multilingual retrieval, nugget generation, and nugget clustering. For RAG4Reports, we focused on the final generation stage and tested a citation-aware compression strategy: generating the long report first from clustered evidence nuggets and then deriving the short report from it, rather than generating both length conditions independently. Our baseline run, which followed the original TREC-style setup, ranked third overall. Our best run, `genaius-cluster-gpt4`, ranked second overall with an F1 score of 0.5456, achieving the best balance among our submissions between nugget coverage and sentence support. The results suggest that citation-aware compression is a promising strategy for length-constrained, citation-grounded report generation.

1 Introduction

Retrieval-augmented generation (RAG) is increasingly used for producing grounded responses from large document collections. However, long-form report generation poses challenges beyond short-form question answering: systems must retrieve evidence across many documents, organize it into a coherent narrative, cover multiple aspects of the information need, and provide reliable citations for generated claims. These challenges are further amplified in multilingual settings, where relevant evidence may appear in different languages.

The RAG4Reports 2026 Multilingual Report Generation Task directly targets this setting by evaluating systems that generate citation-grounded reports from a multilingual news collection. Our system builds on the question-driven retrieval and evidence-based synthesis framework developed for our TREC 2025 RAGTIME submission (Yeniterzi and Yeniterzi, 2025). Since the RAG4Reports topics overlapped with topics we had previously processed, we reused the earlier question-generation and retrieval outputs and focused on testing generation-side modifications that were not fully explored in our TREC submission.

In particular, we experimented with generating the long report first and then producing the short report through citation-aware compression, rather than generating both length conditions independently. Our best run, `genaius-cluster-gpt4`, ranked second on the official task leaderboard, suggesting that this strategy is effective for balancing nugget coverage and sentence-level citation support.

2 Task Description

RAG4Reports 2026 focuses on retrieval-augmented generation for long-form, citation-grounded report generation. It is a direct continuation of the TREC 2025 RAGTIME track (Lawrie et al., 2025), which studied multilingual, citation-grounded report generation over news documents. As in RAGTIME, the task requires systems to address cross-lingual evidence access, evidence selection, long-form synthesis, and faithful attribution. Compared with short-form question answering, this setting places greater emphasis on retrieving and organizing evidence across multiple documents, producing coherent and information-dense reports, and maintaining explicit sentence-level grounding in source evidence.

Multilingual Report Generation Task uses a news document collection in English, Chinese, Russian, and Arabic, sampled from Common Crawl News from 2021 to 2024. In this task, each report request is written in English and includes contextual back-

Proceedings of the First Workshop on Multilingual Report Generation via Retrieval Augmented Generation (RAG4Reports), pages 83–88

ground about the requester as well as a problem statement describing the information need. Systems are required to generate a report in the same language as the request, with every sentence supported by citations to source documents.

3 System Overview

Our system builds on the question-driven retrieval and evidence-structuring framework developed for our TREC 2025 RAGTIME submission (Yeniterzi and Yeniterzi, 2025). Since RAG4Reports is a direct continuation of the RAGTIME setting, it provided an opportunity to examine generation-side design choices that we were not able to fully explore in the original TREC submission.

For RAG4Reports, we reused the intermediate artifacts produced by our RAGTIME pipeline through the evidence-organization stage. Specifically, we reused the generated search questions, multilingual retrieval results, generated information nuggets, and nugget clusters for the overlapping report requests. Our RAG4Reports-specific modifications therefore begin only at the final report generation stage. Figure 1 provides an overview of the resulting pipeline.

3.1 Evidence Preparation Pipeline

The reused part of the pipeline consists of four stages: question generation, multilingual retrieval, nugget generation, and nugget clustering. We summarize these stages briefly for completeness and refer readers to our TREC RAGTIME system paper for the full prompts and implementation details (Yeniterzi and Yeniterzi, 2025).

First, each broad report request is decomposed into a set of focused search questions. This step transforms a complex long-form report request into multiple targeted information needs, helping the system cover different aspects of the topic more systematically. The generated questions serve as the main intermediate representation for retrieval.

Second, multilingual evidence is retrieved for each generated question. The retrieved documents were then merged across questions, with documents appearing for multiple questions receiving higher aggregate scores. This aggregation emphasizes sources relevant to several dimensions of the report request.

Third, the retrieved evidence is converted into concise factual nuggets. For each generated question, the system uses the top-ranked retrieved document to generate information nuggets conditioned on the original report request. Each nugget captures a single relevant fact or aspect of the topic and serves as an atomic evidence unit for downstream synthesis.

Fourth, the generated nuggets are grouped into thematic clusters using an LLM. The clustering step groups semantically related nuggets into distinct aspects, subtopics, or dimensions of the report. Irrelevant nuggets may be discarded during this step, and each cluster is assigned a descriptive label to improve interpretability. These clustered nuggets provide the structured evidence representation used by the final generation stage.

In RAG4Reports, the report requests were a subset of the topics we had already processed in our TREC RAGTIME experiments. We therefore reused the corresponding outputs from all four stages above. This allowed us to hold evidence preparation fixed and focus on the effect of modifying the final report generation strategy.

3.2 RAG4Reports-Specific Report Generation

Both TREC RAGTIME and RAG4Reports require reports under two length conditions: a shorter report and a longer report. In our TREC experiments, we generated both versions independently from the clustered evidence. We observed, however, that our longer reports generally performed better than our shorter reports, likely because the longer setting allowed the system to preserve more supporting evidence, cover more facets of the topic, and maintain smoother narrative organization.

For RAG4Reports, we therefore changed only the final generation strategy. Starting from the reused nugget clusters, we first generated the long report. Each generated sentence was required to be grounded in the clustered evidence and associated with supporting citations. We then produced the short report by compressing the generated long report, rather than generating the short report directly from the clustered nuggets.

The compression step is designed as citation-aware editing rather than generic summarization. The model is instructed to preserve only the highest-priority factual claims needed to answer the request, merge related claims into compact sentences, remove lower-priority details, and retain only citations that support the preserved claims. This separates comprehensive evidence synthesis from length-constrained content selection. The full prompt used for this compression step is shown in

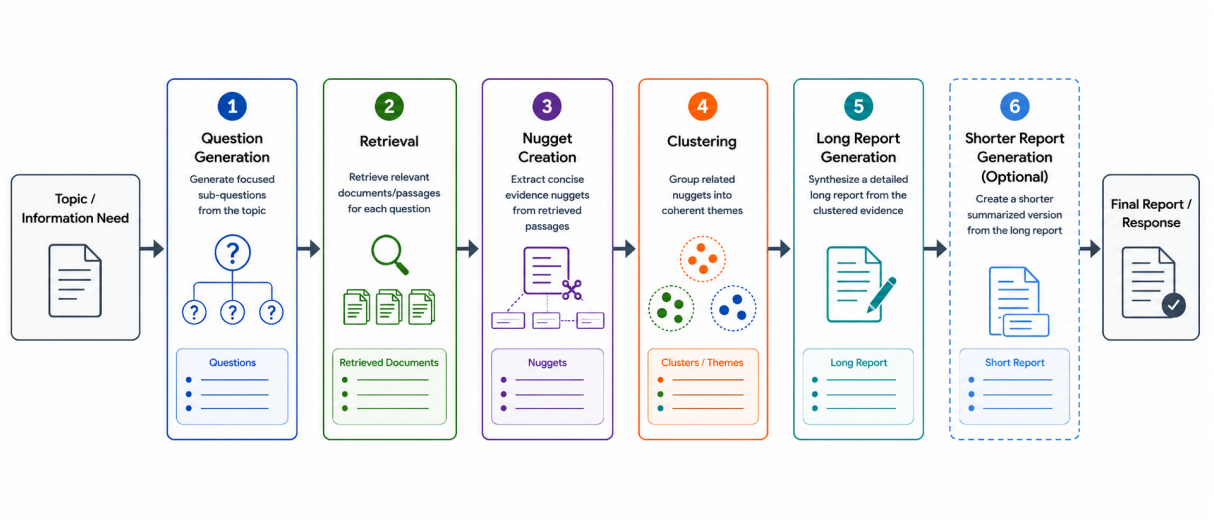

Figure 1: Overview of our RAG4Reports pipeline

Appendix A, Figure 2.

4 Experimental Configuration

We submitted three runs to the official RAG4Reports Multilingual Report Generation Task leaderboard. All runs reused the same evidence-preparation artifacts from our TREC RAGTIME pipeline: generated questions, multilingual retrieval results, information nuggets, and nugget clusters. The submissions therefore differed only in the final report-generation strategy and in the model used for the RAG4Reports-specific generation stages.

4.1 Submitted Runs

`trial`: The `trial` run serves as a baseline approach. It corresponds to the same generation setup used in our earlier TREC system run, `genaius-cluster` (Yeniterzi and Yeniterzi, 2025). In this configuration, the short and long reports were generated independently from the retrieved evidence using GPT-4o (`2024-11-20`) (OpenAI, 2024). We use this run as a comparison point for evaluating the compression-based strategy.

`genaius-cluster-gpt4`: This run also used GPT-4o (`2024-11-20`) for the RAG4Reports-specific generation stages. The long reports were the same as those produced in the `trial` run, generated from clustered evidence nuggets. The short reports, however, were derived from the long reports using citation-aware compression.

`genaius-cluster-gpt5`: This run used the same cluster-based generation and compression strategy as `genaius-cluster-gpt4`, but replaced GPT-4o with GPT-5.4 (`2026-03-05`) (OpenAI, 2026) in the generation stages. Since the clustered nuggets were fixed across runs, differences between these two submissions primarily reflect differences citation-aware generation and compression behavior.

No task-specific fine-tuning or parameter updates were performed in any run. All model calls were conducted in a prompting-based setting, and no additional external datasets were used.

4.2 Evaluation Metrics

RAG4Reports Multilingual Report Generation Task uses the Auto-ARGUE framework (Walden et al., 2026) to evaluate multilingual report generation with citation-aware quality metrics. The main reported measures are nugget coverage, sentence support, and their combined F1 score.

Nugget coverage measures how well the generated report captures relevant information units expected for the report request. Sentence support measures whether generated sentences are supported by the cited source documents. The F1 score combines these two dimensions, rewarding systems that balance information coverage with citation-grounded faithfulness.

These metrics directly reflect the goals of our system design: the cluster-based long-report generation stage aims to improve coverage by organizing evidence into factual units and themes, while the compression-based short-report generation stage aims to preserve high-priority supported claims under a strict length constraint.

Rank	Run	Sent. Supp.	Nugget Cov.	F1
3	`trial`	0.7936	0.4395	0.5382
2	`cluster-gpt4`	**0.8140**	0.4428	**0.5456**
5	`cluster-gpt5`	0.6620	**0.4487**	0.5006

Table 1: Official RAG4Reports Multilingual Report Generation Task results for our submitted runs. "Rank" refers to the rank within the overall task leaderboard. "Sent. Supp." denotes sentence support and "Nugget Cov." denotes nugget coverage.

5 Results

Table 1 reports the official RAG4Reports Multilingual Report Generation Task leaderboard results for our three submissions. The `trial` run serves as our baseline because it follows the original TREC-style setup, where the short and long reports were generated independently from the evidence. Despite not using the new compression-based strategy, this run ranked third overall with an F1 score of 0.5382. This strong result shows that the reused question-driven retrieval, nugget generation, clustering, and independent report-generation pipeline remained highly competitive in the RAG4Reports setting.

The `genaius-cluster-gpt4` run achieved our best overall performance, ranking second with an F1 score of 0.5456. Compared with the `trial` baseline, it improved both sentence support, from 0.7936 to 0.8140, and nugget coverage, from 0.4395 to 0.4428. This suggests that generating the long report first and then deriving the short report through citation-aware compression helped preserve important information while improving sentence-level grounding. Among our submissions, this run provided the best balance between coverage and citation faithfulness.

The `genaius-cluster-gpt5` run ranked fifth with an F1 score of 0.5006. It achieved the highest nugget coverage among our submissions, with a score of 0.4487, but its lower sentence support score of 0.6620 reduced the final F1. This indicates that the GPT-5 variant preserved or introduced more expected information units, but its generated sentences were less consistently supported by the cited evidence. The result highlights an important tradeoff in citation-grounded report generation: increasing coverage is not sufficient if citation support decreases.

Overall, our results show that the original TREC-style baseline was already strong, while the compression-based variant provided the best improvement over it. The comparison across runs suggests that effective RAG4Reports systems must balance two competing goals: covering relevant nuggets and maintaining reliable sentence-level citation support.

6 Conclusion

We described the GenAIus system for RAG4Reports 2026 Multilingual Report Generation Task. Our approach reused the evidence preparation stages from our TREC RAGTIME pipeline, including question generation, multilingual retrieval, nugget generation, and clustering, and focused on modifying the final report-generation stage.

Our main experiment was to generate the long report first and then derive the short report through citation-aware compression. The original TREC-style baseline already performed strongly, ranking third overall, while the compression-based variant ranked second and achieved the best balance between nugget coverage and sentence support among our submissions.

These results suggest that citation-aware compression can be an effective strategy for length-constrained, citation-grounded report generation. In future work, we plan to study stronger citation-verification methods, and more systematic ablations of the final generation stage, with the goal of improving nugget coverage and sentence support jointly rather than optimizing one at the expense of the other.

References

Dawn Lawrie, Sean MacAvaney, James Mayfield, Luca Soldaini, Eugene Yang, and Andrew Yates. 2025. Overview of the TREC 2025 RAGTIME track. In *Proceedings of the 34th Text REtrieval Conference (TREC 2025)*.

OpenAI. 2024. Gpt-4o. https://developers.openai.com/api/docs/models/gpt-4o. Accessed: 2026-03-10.

OpenAI. 2026. Gpt-5.4. https://developers.openai.com/api/docs/models/gpt-5.4. Accessed: 2026-03-10.

William Walden, Marc Mason, Orion Weller, Laura Dietz, John Conroy, Neil Molino, Hannah Recknor, Bryan Li, Gabrielle Kaili-May Liu, Yu Hou, Dawn Lawrie, James Mayfield, and Eugene Yang. 2026. Auto-argue: Llm-based report generation evaluation. *Preprint*, arXiv:2509.26184.

Suveyda Yeniterzi and Reyyan Yeniterzi. 2025. Question-driven multilingual retrieval and report generation for the ragtime track at trec 2025. In *Proceedings of the 34th Text REtrieval Conference (TREC 2025)*.

A Appendix

```
You are an expert editor for evidence-grounded RAG reports.

Objective:
Rewrite a previously generated report into a substantially shorter version, preserving only the
↪  highest-priority factual content, narrative coherence, and citation faithfulness.

The input report is approximately {original_length_words} words.
The maximum allowed output length is {length} words.
The compression ratio is approximately {compression_ratio}.

Your Task:
Create a shorter report by selecting, merging, deleting, and rewriting only the most important
↪  supported claims.

Mandatory Compression Policy:
- This task is not light editing; it is aggressive selection under a strict word budget.
- You must discard at least 50% of the original report content.
- The final output must contain no more than {length} words.
- If there is a conflict between coverage and length, length compliance wins.
- Do not preserve all claims.
- Do not preserve all themes equally.
- Do not preserve all citations.
- If uncertain whether to keep a claim, omit it.

Content Selection Rules:
- Use only information already present in the Generated Report.
- Do not add new facts, assumptions, explanations, or outside knowledge.
- Preserve only claims directly needed to answer the Description.

Compression Strategy:
1. Identify the main themes required by the Description.
2. Select only the highest-value claims under each theme.
3. Merge closely related claims into compact sentences.
4. Rewrite sentences with high information density.
5. Stop adding content once the report approaches {length}.
6. If the report is still too long, delete lower-priority claims rather than making sentences
↪  longer.

Citation Rules:
- Each output sentence may contain one or more citations.
- If a sentence merges claims from multiple original sentences, include only the citations that
↪  support the retained claims.
- Rank citations by importance, placing the citation supporting the main claim first.
- Remove duplicate citations.
- Do not invent, alter, or normalize citation IDs.
- Do not include citations for claims that were removed.

Final Deletion Pass:
Before output, silently perform a deletion pass:
- Remove any sentence that is not directly necessary to answer the Description.
- Remove any sentence whose main point is already implied by another sentence.
- Remove any low-value detail included mainly for completeness.
- Remove any citation whose supported claim was deleted.
- If the result is above {length}, revise again until it is within {length} words.

Output Requirements:
- Return only a valid JSON list.
- Each object must contain:
  - "text": one coherent sentence in the shortened report.
  - "citations": a ranked list of citation IDs, for example ["3_1_1", "4_2_1"].
- Do not include comments, markdown, explanations, word counts, or metadata outside the JSON
↪  list.
- The final list must collectively contain no more than {length} words.
```

Figure 2: Prompt for citation-aware compression of long reports into short reports