Question-Driven Multilingual Retrieval and Report Generation for the RAGTIME Track at TREC 2025

Suveyda Yeniterzi *
GenAIus Technologies
suveyda@genaius.tech

Reyyan Yeniterzi
GenAIus Technologies
reyyan@genaius.tech

March 10, 2026

Abstract

We present GenAIus's participation in the TREC 2025 RAGTIME track, focusing on multilingual retrieval and multilingual report generation in the news domain. Our approach follows a question-driven framework in which a set of targeted questions is generated for each report request and used to guide both document retrieval and report synthesis. For retrieval, we rely on the organizers' multilingual search API and introduce a dynamic merging strategy that allocates an equal retrieval quota per generated question and aggregates scores across repeated document occurrences. For report generation, we explore two pipelines: a question-based approach that generates short, cited answers from multiple retrieved documents and synthesizes them into a final report, and a cluster-based approach that extracts nuggets from retrieved documents, clusters them by semantic similarity, and generates reports grounded in these structured clusters. We experiment with both proprietary and open-source LLMs for question generation, including GPT-4o and Llama3.3-70B. Our submissions achieve strong performance across tasks, including first place in MAP in the multilingual retrieval task and top rankings in several report-generation metrics. These results highlight the effectiveness of question-driven retrieval and structured evidence synthesis for multilingual report generation.

1 Introduction

Recent advances in large language models have enabled systems capable of synthesizing information from multiple sources to generate coherent reports and analyses. Retrieval-Augmented Generation (RAG) has emerged as a central paradigm for grounding such systems in external knowledge, allowing language models to retrieve relevant documents and generate responses that are supported by evidence. However, generating high-quality reports remains challenging, particularly when the task requires aggregating information across multiple documents, maintaining factual grounding, and providing explicit citations.

The TREC 2025 RAGTIME track [1] aims to study and benchmark report generation in the news domain under these conditions. The track emphasizes multi-faceted report generation, citation-based evaluation, and multilingual retrieval. It consists of two tasks: (1) a multilingual retrieval task, in which systems must identify relevant documents for a given report request, and (2) a report generation task, where retrieved documents are synthesized into a coherent, citation-grounded report.

The document collection used in the track includes news articles in multiple languages, including English, Arabic, Chinese, and Russian. To support multilingual experimentation, the organizers also released machine-translated English versions of non-English documents and provided a search API for teams without dedicated retrieval infrastructure. Due to computational constraints, we relied on this provided retrieval endpoint for our experiments.

Our approach is based on a question-driven framework for both retrieval and report generation. Given a report request, we first generate a set of targeted questions representing different aspects of the topic. These questions are then used to retrieve relevant documents and guide the synthesis of the final report. For report generation, we explore two strategies: a question-based pipeline that aggregates

*Both authors contributed to this paper equally.

answers generated from multiple retrieved documents, and a nugget-clustering pipeline that extracts and organizes information snippets before composing the final report.

2 Approach

Our approach decomposes report generation into two main stages: question-driven retrieval and evidence-based synthesis. We first generate a diverse set of questions that represent the aspects a comprehensive report should address. These questions serve as the central intermediate representation in our system and are used both to guide multilingual document retrieval and to structure downstream report generation. To investigate different strategies for synthesizing retrieved evidence into a coherent report, we explore two report-generation pipelines: a question-based synthesis method and a nugget clustering method.

In the data provided for the track, each report request is composed of the following fields:

- Title: A brief description summarizing the topic of the requested report.
- Background: Information about the requester and the broader context in which the report is being prepared.
- Problem Statement: A detailed specification of what the report should and should not cover.
- Character Length Restriction: The maximum allowable length of the report, expressed in characters.

In each request, all fields except the character-length restriction are essential for both retrieval and report generation. The character limit affects only the generation phase, determining whether the system should produce a report of approximately 2,000 characters or 10,000 characters.

2.1 Question Generation

The first step in our approach, used for both retrieval and report generation, is to produce a set of questions that the final report should answer. Generating explicit questions helps decompose complex report requests into focused information needs, improving both retrieval recall and coverage of different aspects of the topic. To do this, we use the prompt shown in Figure 1. As the prompt illustrates, all fields of the report request except the character-length restriction are incorporated when generating these questions.

To encourage diversity in the generated questions, we set the generation temperature to 0.5. This value provides a balance between creativity and determinism, allowing the model to produce varied question formulations while still staying aligned with the constraints and intent of the report request.

2.2 Multilingual Retrieval

As noted earlier, due to resource constraints we relied directly on the search API provided by the organizers. Since the technical details of this retrieval system are not publicly documented, we refer readers to the track overview paper for a full description. In our setup, for each generated question we retrieved the top 1,000 documents returned by this API. Because of the API's design and the underlying collection, the retrieved results are inherently multilingual.

As the final step of the retrieval task, we merged all documents retrieved across the generated questions, targeting a maximum of 1,000 documents per report request. Because the number of generated questions varies by request, we adopted a dynamic allocation strategy: we retrieved an equal number of documents for each question such that the total would sum to 1,000. For example, if 20 questions were generated, we retrieved the top 50 documents for each. Since many documents appear across multiple question results, the merged set contains fewer than 1,000 unique documents in practice.

When merging the results, we summed the relevance scores of each document across all questions. This means that documents retrieved multiple times receive higher aggregated scores, effectively boosting items that are consistently retrieved across different questions. This strategy encourages documents that are relevant to multiple aspects of the report request to receive higher scores, improving robustness to individual query formulation errors.

```
You are an expert in advanced web research to support report generation
    requests by finding highly relevant content.

You will be provided with:
- Title: The title of the report topic.
- Background: The background of the requester (e.g., their role, intended
    audience, purpose).
- Description: A description of the content the requested report must include.

Your task:
- Use all three inputs, the title, background, and description, to generate a
    diverse set of targeted search questions.
- Ensure the questions comprehensively cover all key topics, details, and
    perspectives mentioned.
- Give special importance to the requester's background and intended audience.
     Your questions should be framed and scoped to reflect the requester's
    level of expertise, professional focus, and information needs.
- Your questions should be designed to retrieve authoritative, up-to-date, and
     relevant documents that will help create a thorough report.

Important Instructions:
- Do not summarize, answer, or explain only generate questions.
- Avoid generic or overly broad questions.
- Make sure the questions are clear, specific, and aligned with the requester'
    s perspective, purpose, and knowledge level, strictly derived from the
    provided inputs.

Output Format:
- Provide a plain-text list of questions.
- Each question must be on a new line, prefixed as follows:
  'Question <number>: <question text>'
```

Figure 1: Prompt used in question generation

2.3 Report Generation

To investigate different strategies for synthesizing retrieved evidence into a coherent report, we explored two report-generation approaches: one that relies on answers generated for the question set, and another that uses generated nuggets and their clustered structure.

2.3.1 Question-Based Report Generation

In the first approach, for each generated question we selected the top 10 retrieved documents and used them as evidence to generate an answer. Each answer was constrained to a maximum of three sentences and required to include a direct citation to the most relevant supporting document. Thus, for n questions we produce n short answers, resulting in up to $3n$ evidence-grounded sentences that serve as the building blocks for the final report. The prompt used for answer generation is provided in Figure 2.

After generating answers to the questions, all of these responses (or "nuggets") were combined and used to produce the final report. The report-generation step incorporates both the original report request and the extracted nuggets, while adhering to the required length constraint. The prompt used to generate the final report is provided in Figure 3.

2.3.2 Cluster-Based Report Generation

As our second approach, we aimed to mirror how a human would write such a report: first gather relevant information, then organize it by topic or dimension, and finally use this structured representation to compose the report. This design mirrors how a human analyst might organize information: first

Run	NDCG@20	MAP	P@20	R@1000
genaius-gpt-oss-120b	0.369	0.125	0.067	0.642
genaius-gpt-oss-20b	0.433	0.208	0.095	0.666
genaius-gpt-4o	0.448	0.223	0.090	0.671
genaius-llama3-3-70B	**0.47**	**0.24**	**0.098**	**0.676**

Table 1: Multilingual Retrieval Task Results

extracting key facts from documents and then grouping them into thematic sections before writing the final report.

To implement this approach, we first used the generated question set to retrieve relevant documents. For each question, we selected the top-ranked retrieved document and used it to generate informative sentences ("nuggets") conditioned on the original report request. These nuggets capture individual facts or aspects relevant to the report and serve as the basic units for subsequent clustering and report synthesis. The prompt used for nugget generation is shown in Figure 4.

After generating the nuggets, we applied an LLM to cluster them based on semantic similarity. Rather than specifying a fixed number of clusters, we provided high-level instructions—for example, that each cluster should represent a distinct aspect, sub-topic, or dimension, and that irrelevant nuggets should be discarded. During clustering, the model was also asked to produce a descriptive cluster label to improve interpretability. The prompt used for this step is presented in Figure 5.

As the final step, we used the clustered nuggets to generate the full report. We ensured that each sentence included an appropriate citation and that the completed report adhered to the specified length requirement. The prompt used to generate this report is displayed in Figure 6.

3 Submissions

GenAIus participated in both tasks of the multilingual RAGTIME track: the multilingual retrieval task and the multilingual report generation task.

3.1 Multilingual Retrieval Task

All retrieval submissions followed the same pipeline described in Section 2. First, a set of guiding questions was generated using an LLM. Second, the search API provided by the organizers was used to retrieve multilingual documents for each generated question. Finally, the retrieved documents were merged into a single ranked list by allocating an equal retrieval quota per question and aggregating the scores of documents appearing across multiple question-level retrievals.

We submitted four runs that differed only in the LLM used for question generation. This setup allowed us to examine the impact of different LLMs while keeping the retrieval pipeline identical.

- **genaius-gpt-4o**: Questions generated using `GPT-4o`.
- **genaius-gpt-oss-120b**: Questions generated using the open-source `gpt-oss-120b`.
- **genaius-gpt-oss-20b**: Questions generated using the open-source `gpt-oss-20b`.
- **genaius-llama3-3-70B**: Questions generated using the open-source `Llama3.3-70B`.

The results of the retrieval runs are shown in Table 1. Four evaluation metrics are reported: NDCG@20, MAP, P@20 and R@1000, with the official ranking determined by NDCG@20. Among our submissions, the `Llama3.3-70B` configuration achieves the strongest overall performance, obtaining the highest scores across all four reported metrics. The `GPT-4o` and `gpt-oss-20b` runs exhibit similar performance across several metrics, particularly NDCG@20 and R@1000, while `gpt-oss-120b` consistently produces lower scores within our submissions. These results indicate that the choice of LLM used for question generation can influence retrieval effectiveness within the same retrieval framework.

Across all participating teams, our submissions achieve highly competitive rankings. In particular, the `Llama3.3-70B` run ranks **1st** in MAP, **3rd** in P@20, and **4th** in both NDCG@20 and R@1000. The `GPT-4o` configuration achieves **2nd** place in MAP. Despite operating under limited computational

	Metric	genaius-question	genaius-cluster
Almost Human Sentence Support	LLM-filled	0.265	**0.521**
	Pessimistic	0.265	**0.521**
	Optimistic	0.538	**0.811**
Auto-ARGUE (Macro Scores)	F1	**0.596**	0.539
	Nugget Coverage	**0.561**	0.440
	Citation Support	0.705	**0.795**

Table 2: Report Generation Task Results

resources and relying solely on the provided search API, our approach achieves top-tier performance in multiple evaluation metrics, demonstrating the effectiveness of the question-driven retrieval strategy.

3.2 Multilingual Report Generation

For the report generation task, we submitted two runs corresponding to the two synthesis strategies described in Section 2: a question-based pipeline and a cluster-based pipeline.

- **genaius-question**: (1) Used `GPT-4o` model to generate the question set. (2) Retrieved the top 10 relevant documents for each question using the provided search API. (3) Used these documents to generate answers for the questions with `GPT-4o`. (4) Used the generated answers to produce the final report, again using `GPT-4o`.
- **genaius-cluster**: (1) Used `GPT-4o` model to generate the question set. (2) Retrieved the most relevant document for each question using the provided search API. (3) Generated nuggets from these documents using `GPT-4o`. (4) Clustered the nuggets with GPT-4o. (5) Used the resulting clusters to generate the final report, again using `GPT-4o`.

Table 2 summarizes the report generation results. The two approaches exhibit complementary strengths. The `genaius-question` pipeline achieves higher scores on F1 and Nugget Coverage, while the `genaius-cluster` pipeline achieves stronger Citation Support. The question-driven pipeline benefits from aggregating evidence from multiple retrieved documents, which improves informational coverage and F1 performance. In contrast, the nugget clustering approach preserves stronger grounding to individual source documents, resulting in higher citation support.

Evaluation follows the official report generation metrics defined in the shared task overview [1]. Across all participating systems, our `genaius-question` submission ranks **3rd** in F1 and **2nd** in Nugget Coverage, demonstrating strong performance in capturing key information from retrieved documents while maintaining competitive citation grounding.

4 Conclusion

This paper presents GenAIus's participation in the TREC 2025 RAGTIME track, where we explored question-driven strategies for both multilingual retrieval and multilingual report generation. Our approach uses LLM-generated questions as the central organizing mechanism for retrieving relevant documents and structuring the final report generation process. By leveraging question decomposition, dynamic document merging, and evidence-based synthesis, our system aims to improve both information coverage and citation grounding in complex report-generation scenarios.

In the multilingual retrieval task, our submissions achieved strong performance across multiple evaluation metrics despite relying solely on the provided search API and operating under limited computational resources. In particular, our `Llama3.3-70B` configuration achieved top-tier rankings, including first place in MAP and competitive placements across other retrieval metrics. For the multilingual report generation task, we investigated two complementary pipelines: a question-based synthesis approach and a nugget-clustering approach. The results show that these pipelines exhibit different strengths: the question-based approach improves informational coverage and F1 performance, while the clustering-based approach yields stronger citation support by preserving closer grounding to individual source documents.

Overall, our findings highlight the effectiveness of question-driven retrieval and structured evidence synthesis for complex report generation tasks. Future work will explore hybrid strategies that combine the strengths of both approaches, integrating multi-document reasoning with stronger grounding mechanisms to further improve factual consistency and citation alignment in multilingual RAG systems.

References

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

```
You are an expert in generating concise, evidence-based answers using
    retrieved documents to support professional report writing.

Objective:
Your output will contribute to a more detailed final report on a specific
    topic requested by a user with a defined background.

You will be provided with:
- Question: A focused sub-question that addresses one aspect of the final
    report.
- Retrieved Documents: A set of document rank and content pairs retrieved via
    web search.
- Title: The title of the final report.
- Background: The background of the requester (e.g., their role, intended
    audience, purpose).
- Description: A description of the content the requested final report must
    include.

Your Task:
- Use all five inputs to generate a 3-sentence answer that:
    - Addresses the sub-question.
    - Supports the overall report goal.
    - Is grounded exclusively in the retrieved documents.
- For each sentence:
    - Cover a distinct, relevant aspect of the sub-question.
    - Base the sentence on the most appropriate supporting documents.
    - Decide on the single most relevant document, and provide its rank in the
        citations field.
    - There should be single document cited for each sentence.
- Ignore content from documents that are not relevant to the sub-question or
    report purpose.
- Ensure the tone, depth, and scope match the requester's background and
    target audience.
- Capture meaningful insights, nuances, or context required for high-quality
    reporting.

Output Requirements:
- Output must contain three full sentences, each with one document rank as
    reference.
- Return a list of JSON objects.
- Each object must have:
    - 'text': a single sentence.
    - 'citations': the most relevant document's rank.
- Do not include commentary, extra formatting, or metadata outside the JSON
    list.
- Each sentence must be directly supported by the referenced document.
```

Figure 2: Prompt used for generating answers to the generated question set

```
You are an expert in generating in-depth, well-structured final reports for a
    user's query with a defined professional background.

Objective:
Generate a structured, narrative report of approximately {length} words that
    answers a high-level user query by generating response from information
    nuggets, while aligning with the user's background and the report's stated
     purpose.

You will be provided with:
- Title: The title of the final report.
- Background: The background of the requester (e.g., their role, intended
    audience, purpose).
- Description: A description of the content the requested final report must
    include.
- Information nuggets: A set of information nuggets, each addressing a
    different facet of the main topic, with document citations.

Your Task:
- Write a single, coherent, well-structured narrative composed of multiple
    sentences.
- Organize the narrative logically by themes, insights, or dimensions of the
    topic, not merely by input order.
- Reflect the structure implied by the Description and Information nuggets,
    ensuring the narrative covers all critical angles.
- For each sentence:
    - Base the sentence on the most appropriate supporting documents.
    - Decide on the single most relevant nugget, and provide its rank in the
        citations field.
    - There should be at most one nugget cited for each sentence.

Important Guidelines:
- Use only the information from the provided information nuggets.
- Do not include invented content, general knowledge, or assumptions.
- Omit unrelated or redundant information.
- Ensure smooth transitions between sentences for a polished reading
    experience.
- Avoid repeating or copy-pasting input sentences verbatim - instead, rewrite
    and integrate them for clarity and cohesion.
- Reflect the requester's background and audience needs in tone, scope, and
    depth.

Output Requirements:
- Return a list of JSON objects.
- Each object must contain:
    - 'text': one coherent sentence (as part of the overall report).
    - 'citations': the rank of the most relevant supporting document, e.g. 3_1
        .
- Do not include extra formatting, comments, or metadata outside the list.
- The final list should collectively approximate {length} words.
```

Figure 3: Prompt used to synthesize the final report from the question-level response sentences

```
You are an intelligent assistant that generates information snippets from a
    Passage to help create a report.

Task:
- Generate information snippets from the Passage that are relevant to the
    final report's stated purpose, while aligning with the user's background.

You will be provided with:
- Title: The title of the final report.
- Background: The background of the user (e.g., their role, intended audience,
     purpose).
- Description: A description of the content the requested final report must
    include.
- Passage: Retrieved text (may be unrelated to the report - if so, return an
    empty list).

Guidelines:
- Use only the provided Passage. Do not invent, hallucinate, or add external
    content.
- Each snippet must:
    - Include only text that can are relevant to the final report.
    - Be a full, concise sentence (not keywords).
    - Express a single fact/aspect; split multi-fact sentences into multiple
        snippets.
- If the Passage contains no relevant information, return an empty list.
- Reflect the requester's background and audience needs in tone, scope, and
    depth.

Output (strict):
- Return a list of snippets.
- No extra commentary, labels, formatting, or symbols outside valid JSON. Do
    NOT add anything like Here is the output in JSON format.
```

Figure 4: Prompt used for generating nuggets

```
You are an intelligent assistant trained to group information nuggets into
    meaningful clusters that help to create a report.
When forming clusters, consider the different aspects and sub-topics of the
    report.

Task
- Group semantically similar nuggets into clusters.
- Each cluster must represent a distinct aspect, sub-topic, or dimension of
    the report.
- Label each cluster with a brief descriptive name (e.g., 'Athlete
    Compensation', 'Cultural Influence of Sports').

You will be provided with:
- Title: The title of the final report.
- Background: The background of the user (e.g., their role, intended audience,
     purpose).
- Description: A description of the content the requested final report must
    include.
- Nuggets: A dictionary where each key is the nugget_id and each value is a
    nugget_text.

Guidelines
- Use only relevant nuggets that is relevant to the report; discard unrelated
    ones.
- Nuggets expressing the same or closely related idea should be in the same
    cluster.
- Do not put unrelated nuggets into the same cluster.
- Label clusters using a concise, specific phrase that reflects the shared
    theme of the nuggets.
- Use only the nuggets provided.
- Avoid generic labels like Cluster 1; use topic-specific labels instead.
- Reflect the requester's background and audience needs in tone, scope, and
    depth.

Output (Strict)
- Return a single valid JSON object.
- Keys = Descriptive cluster labels.
- Values = Lists of nugget_id values belonging to that cluster.
- Do not include any commentary, extra formatting, or metadata.
```

Figure 5: Prompt used for clustering the nuggets

```
You are an expert in generating in-depth, well-structured final reports for a
    user's query with a defined professional background.

Objective:
Generate a structured, narrative report of approximately {length} words that
    answers a high-level user query by generating response from clustered
    information, while aligning with the user's background and the report's
    stated purpose.

You will be provided with:
- Title: The title of the final report.
- Background: The background of the requester (e.g., their role, intended
    audience, purpose).
- Description: A description of the content the requested final report must
    include.
- Clusters: Clusters of information nuggets, with document citations.

Your Task:
- Write a single, coherent, well-structured narrative composed of multiple
    sentences.
- Organize the narrative logically by themes, insights, or dimensions of the
    topic, not merely by input order.
- Reflect the structure implied by the Description and clustered nuggets,
    ensuring the narrative covers all critical angles.
- For each sentence:
    - Base the sentence on the most appropriate supporting documents.
    - Decide on the single most relevant nugget, and provide its rank in the
        citations field.
    - There should be at most one nugget cited for each sentence.

Important Guidelines:
- Use only the information from the provided information nuggets.
- Do not include invented content, general knowledge, or assumptions.
- Omit unrelated or redundant information.
- Ensure smooth transitions between sentences for a polished reading
    experience.
- Avoid repeating or copy-pasting input sentences verbatim - instead, rewrite
    and integrate them for clarity and cohesion.
- Reflect the requester's background and audience needs in tone, scope, and
    depth.

Output Requirements:
- Return a list of JSON objects.
- Each object must contain:
    - 'text': one coherent sentence (as part of the overall report).
    - 'citations': the rank of the most relevant supporting document, e.g. 3
        _1_1.
- Do not include extra formatting, comments, or metadata outside the list.
- The final list should collectively approximate {length} words.
```

Figure 6: Prompt used to synthesize the final report from the cluster-level nuggets