From Nuggets to Clusters: Multi-Level Evidence Structuring for the TREC 2025 RAG Track

Reyyan Yeniterzi *
GenAIus Technologies
reyyan@genaius.tech

Suveyda Yeniterzi
GenAIus Technologies
suveyda@genaius.tech

February 15, 2026

Abstract

We present GenAIus's participation in the TREC RAG track, focusing on the augmented generation and relevance judgment tasks. For augmented generation, we build two LLM-based pipelines: a nugget-based approach that converts passages into concise evidence units for response generation, and a cluster-based approach that groups nuggets by subtopic before synthesizing a citation-grounded answer. For the relevance judgment task, we reuse the nuggets, clusters, and generated responses to automatically score passage relevance. We develop five ranking methods based on nugget counts, length-normalized nugget counts, cluster membership, unique cluster coverage, and citation frequency.

1 Introduction

The goal of TREC RAG is to advance innovation and research in retrieval-augmented generation (RAG) systems. The track uses the MS MARCO V2.1 dataset [1] as its primary search corpus and is organized into four tasks: (1) Retrieval: Participants must rank and retrieve the most relevant segments from the corpus for a given set of input topics. (2) Augmented Generation: Participants generate RAG answers that include explicit attributions to supporting segments from the collection. (3) Retrieval-Augmented Generation: Participants produce RAG answers together with attributions for the supporting retrieved segments. (4) Relevance Judgment: Participants create their own QREL files by generating relevance judgments for topic–segment pairs.

Due to limited computational resources, we were unable to participate in the retrieval tasks; however, we did participate in both the augmented generation and relevance judgment tasks.

The goal of the augmented generation task is to evaluate the generation component of RAG systems. To isolate this step, the organizers provided a strong baseline retrieval run. For full details of this baseline, we refer readers to the overview paper [3].

The relevance judgment task was introduced as an experimental subtask this year, with the goal of evaluating whether participants could automatically generate QREL files. To ensure a consistent assessment pool across teams, participants were instructed to use the top 20 passages from the same baseline retrieval run that was provided for the augmented generation task.

2 Approach

2.1 Augmented Generation

In this task, we relied on the strong baseline retrieval results provided by the organizers. The baseline includes the top 100 retrieved passages for each topic; in our approach, we used only the top 20 passages.

*Both authors contributed to this paper equally.

```
You are an intelligent assistant that generates information snippets from a
    Passage to help answer a given Query.

Task:
- Generate information snippets from the Passage that are relevant to the
    Query.

Input:
- Query: A user question.
- Passage: Retrieved text (may be unrelated to the Query - 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 be used to answer the Query.
    - 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 the Rank key with an
     empty list.

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 1: Prompt used for generating nuggets

Rather than feeding these passages directly into the LLM, we adopted a nugget-based strategy. For each passage, we first generated concise informational nuggets using an LLM. The prompt used for nugget generation is shown in Figure 1.

We explored two different approaches for the next stage. In the first, which we refer to as the *nugget-based approach*, we used all generated nuggets as the evidence base for producing the final response. The prompt used for this step is shown in Figure 2. As illustrated in the figure, the prompt includes a length constraint and requires that each sentence contain an appropriate citation.

In the second approach, referred to as *cluster-based approach*, we first clustered all generated nuggets based on their semantic similarity using an LLM. We did not specify the number of clusters; instead, we allowed the LLM to determine an appropriate grouping. The goal was to form clusters that capture distinct aspects or subtopics relevant to the final response. To improve interpretability, we also asked the LLM to assign descriptive labels to each cluster. The full prompt used for this step is provided in Figure 3.

The final step of the *cluster-based approach* is to use the generated clusters as the evidence source for producing the final response. As in the previous method, we enforce both the length constraint and the citation requirement. The full prompt used for this step is shown in Figure 4.

2.2 Relevance Judgment

The relevance judgment task relied on the same top 20 passages used in the augmented generation task, so our approach to relevance assessment was naturally shaped by our generation pipeline. We aimed to examine how our nugget-based methodology influenced passage-level relevance judgments. To this end, we reused the intermediate outputs from the augmented generation step, specifically the generated nuggets, clusters, and final response, to inform our relevance assessments in this task.

We developed five different scoring and ranking strategies for assessing passage relevance. The first two rely directly on the generated nuggets, the next two make use of the clustered nuggets, and the final strategy is based on the citations appearing in the generated final response.

```
You are an intelligent assistant trained to write concise and informative
    answers to a given Query using provided information nuggets.

Task: Generate a response using the provided nuggets

Input:
- Query: A user question.
- Nuggets: A dictionary where keys are nugget IDs and values are the
    corresponding nugget texts.

Guidelines:
- Questions may contain subquestions, or ask about different aspects of the
    same topic. Make sure the response covers all these different aspects in a
     coherent way.
- Use only the provided nuggets. Do not invent, infer, or add any new
    information.
- Write a clear and coherent answer to the Query using the nuggets.
- Each sentence must address part of the Query and be fully supported by the
    nugget(s) used.
- After each sentence, cite the nugget IDs in square brackets (e.g., [2_1, 2_4
    ]).
- If multiple nuggets support a single sentence, list all relevant nugget IDs.
- Do not include any nuggets that are irrelevant to the Query.
- The output must be in plain text only, with no extra commentary, formatting,
     or metadata.
- The response should be comprehensive and detailed.
- The response length should be less than {length} words.

Output (Strict):
- Return a list of JSON objects.
- Each object must have:
    - 'text': a single sentence.
    - 'citations': a list of reference IDs supporting that sentence.
- Do not include commentary, extra formatting, or metadata outside the JSON
    list.
- Each sentence must be directly supported by the referenced nuggets.
```

Figure 2: Prompt used to generate the final response from the generated nuggets

Our first approach, *nugget_cnt*, ranks passages based on the number of nuggets generated from each one. Since the query is explicitly incorporated during nugget generation (as shown in the prompt in Figure 1), the number of nuggets reflects how much relevant information a passage contains. Passages that yield more nuggets are therefore assumed to be more relevant than those that yield fewer.

The second approach, *norm_nugget_cnt*, builds on *nugget_cnt* while addressing several limitations. In the original method, ties are common because multiple passages may produce the same number of nuggets. Additionally, longer passages may generate more nuggets simply due to their length, even when they are less relevant than shorter but more informative passages. To mitigate these issues, we compute a normalized score by dividing the number of generated nuggets by the passage's character length. This normalization helps distinguish between genuinely informative passages and those that appear relevant only because they are longer.

Our third approach leverages the generated clusters for scoring. During the clustering step (as shown in the prompt in Figure 3), nuggets are already evaluated for relevance to the query, and irrelevant ones are discarded. In the *cluster_cnt* method, we therefore score passages by counting how many of their nuggets remain after this relevance filtering and clustering process.

In the clustering step, we allow the LLM to organize nuggets into clusters representing distinct subtopics or dimensions of the query. A passage that covers only a single aspect of the topic is fundamentally different from one that contributes to multiple aspects. Our hypothesis is that passages touching on a broader range of dimensions are more likely to be relevant. Based on this assumption,

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

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

Input
- Query: A user question, possibly involving multiple sub-questions or themes.
- Nuggets: A dictionary where each key is the descriptive name and each value
    is a nugget_text.

Guidelines
- Use only relevant nuggets that help answer the Query; 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.

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 3: Prompt used for clustering the nuggets

the *unique_cluster_cnt* method scores each passage by counting the number of distinct clusters in which its nuggets appear. Passages whose nuggets span more unique clusters are thus ranked higher.

Our final approach uses the generated response and its citations to score passages. When producing the response from the clustered nuggets (as shown in Figure 4), the LLM effectively performs an additional layer of relevance assessment by selecting which nuggets and therefore which passages to cite. Based on this intuition, the *citation_cnt* method ranks passages according to how many times they are cited in the final response.

3 Submissions

As GenAIus, we participated in both the augmented generation task and relevance judgment task.

3.1 Augmented Generation Task

For the augmented generation task, we submitted two runs that differed in how extracted evidence nuggets were structured prior to response synthesis:

- **nugget-generation**: (1) Used the `GPT-4o` model to generate nuggets from the top-20 retrieved passages. (2) Used these generated nuggets to generate the final response with `GPT-4o`.
- **cluster-generation**: (1) Used the `GPT-4o` model to generate nuggets from the top-20 retrieved passages. (2) Clustered the nuggets using GPT-4o. (3) Used the resulting clusters to generate the final response, again with `GPT-4o`.

```
Input:
- Query: A user question. A question may contain subquestions, or ask about
    different aspects of the same topic.
- Clusters: Clusters of nuggets

Task: Generate a Response Using Clustered Nuggets

Goal:
Write a comprehensive, informative answer to the Query using the clustered
    nuggets.
Make sure the response covers all different aspects of the question in a
    coherent way.

Guidelines:
- Use only the clustered nuggets provided.
- While answering a specific aspect of the query, make sure to use all
    relevant clusters.
- Make sure the order of the sentences and the overall response is smooth and
    meaningful.
- Each sentence must address part of the Query and be fully supported by the
    nugget(s) used.
- After each sentence, include the relevant nugget IDs in square brackets, e.g
    ., [1_2, 1_4].
- If multiple nuggets support the same sentence, list all IDs used.
- Do not invent or add any information not present in the nuggets.
- Do not include any nuggets that are irrelevant to the Query.
- The response should be comprehensive and detailed.
- The response length should be less than {length} words.

Output (Strict):
- Return a list of JSON objects.
- Each object must have:
    - 'text': a single sentence.
    - 'citations': a list of reference IDs supporting that sentence.
- Do not include commentary, extra formatting, or metadata outside the JSON
    list.
- Each sentence must be directly supported by the referenced nuggets.
```

Figure 4: Prompt used to synthesize the final report from the cluster-level nuggets

Both approaches relied on the same underlying evidence pool but differed in how extracted knowledge was structured prior to response generation.

The results of the nugget-based response evaluation metrics are summarized in Table 1, where the two submissions exhibited closely aligned performance, with nugget-generation consistently achieving slightly stronger results. Strict vital scores for our runs ranged between 0.38 and 0.41, while sub-narrative coverage fell between 0.62 and 0.66. Nugget-generation occupied the upper bound of both ranges, reflecting marginally stronger recovery of indispensable information units and broader narrative coverage. Cluster-generation followed closely but trailed slightly on both metrics.

This performance gap can be attributed to information compression effects introduced during the clustering stage. While clustering improves thematic organization and reduces redundancy, it also abstracts fine-grained factual units that are explicitly measured in nugget-based evaluation. Because strict vital score requires each indispensable nugget to be fully supported, compressing multiple nuggets into higher-level thematic representations can reduce measurable recall even when semantic fidelity is preserved.

Clustering may also attenuate minority or sparsely represented nuggets that nonetheless correspond to distinct sub-narratives, slightly reducing overall coverage breadth. In contrast, directly conditioning generation on the complete nugget set preserves atomic fact granularity, enabling stronger alignment with nugget-based evaluation criteria and resulting in modestly higher recall and coverage scores.

Run	Nugget PE + AutoAssign		AutoNuggetizer	
	strict vital score	sub-narrative cov.	strict vital score	sub-narrative cov.
nugget-generation	0.3800	0.6600	0.4100	0.6300
cluster-generation	0.3800	0.6400	0.4000	0.6200

Table 1: Automatic Evaluation Scores on RAG

Run	Manual Assessment		LLM Assessment	
	Wght. Precision	Wght. Recall	Wght. Precision	Wght. Recall
nugget-generation	0.7780	0.7780	0.8220	0.8220
cluster-generation	0.7483	0.7483	0.7920	0.7920

Table 2: Support Evaluation Scores

In overall leaderboard positioning, both submissions ranked within the upper-middle tier of participating systems, with nugget-generation placing modestly higher than cluster-generation across response evaluation settings. While these scores trail the top-ranked entries, their interpretation should be contextualized within the structural differences between Augmented Generation and Retrieval-Augmented Generation systems. Leading runs in the response evaluation track predominantly employed RAG pipelines capable of expanding evidence coverage beyond the provided context. Because strict vital score and sub-narrative coverage depend directly on the breadth and diversity of retrieved information, AG systems operate under an inherent evidence ceiling determined by the initial document set. When vital nuggets or narrative perspectives are absent from the supplied passages, AG systems cannot recover them through retrieval, naturally constraining achievable recall and coverage.

To further examine whether these limitations stemmed from evidence quality or evidence availability, we analyze the retrieval assessment over citations. In this evaluation, cited documents were treated as a ranked list and assessed using nDCG@30 to measure how well supporting evidence aligned with response information needs. Our runs achieved an nDCG@30 score of 0.5373, placing GenAIus in the mid-to-upper range of participating systems and above several RAG submissions.

This result indicates that the documents cited in our generated responses were highly relevant and well-prioritized relative to the evaluation queries. Importantly, this demonstrates that lower nugget recall and sub-narrative coverage cannot be attributed to weak evidence selection or grounding. Rather, they reflect the bounded evidence pool inherent to the AG setting. Because our systems could only cite from the provided document set, citation relevance was optimized within a constrained retrieval space, whereas higher-ranked RAG systems benefited from the ability to retrieve and rank evidence from a substantially broader corpus.

Taken together, these findings suggest that GenAIus responses were strongly grounded in relevant evidence but operated under structural coverage limitations imposed by the AG evaluation condition. Thus, performance differences relative to top RAG systems primarily reflect retrieval breadth advantages rather than deficiencies in evidence utilization or response synthesis.

As shown in Table 2, GenAIus submissions in the AG task were further evaluated under the support evaluation framework [2], which measures the extent to which generated answer sentences are factually grounded in their cited documents. This evaluation operates at the sentence level, assigning support judgments—Full, Partial, or No Support—based on how well cited evidence substantiates each statement. Performance is summarized using weighted precision and weighted recall, capturing citation faithfulness across responses.

Under human support evaluation, the nugget-generation pipeline achieved a weighted precision and recall of 0.7780, placing it approximately 4th overall among submitted systems. The cluster-generation pipeline achieved 0.7483, ranking around 6th. These results indicate that a large proportion of sentences produced by both approaches were well-supported by their cited evidence, with nugget-generation demonstrating modestly stronger grounding alignment.

A consistent pattern is observed in the automatic support evaluation results (Table 2), where grounding was assessed using GPT-OSS 120B as the judge. Nugget-generation achieved weighted precision and recall of 0.8220, ranking 5th overall, while cluster-generation achieved 0.7920, placing 7th. The stability of these rankings across human and automated assessment suggests that the observed differences are systematic rather than evaluator-dependent.

Run	agreement_frac	kappa_val
nugget_cnt	0.2800	0.1300
norm_nugget_cnt	0.2600	0.0600
cluster_cnt	0.2800	0.0900
unique_cluster_cnt	0.2800	0.0700
citation_cnt	0.2800	0.0700

Table 3: Alignment Comparison Scores for Relevance Judgment Task

The performance gap between the two pipelines can be attributed to structural differences in how evidence was incorporated during generation. Nugget-generation conditions directly on atomic factual units extracted from supporting passages, encouraging sentence constructions that closely mirror discrete evidence segments. This facilitates clearer citation alignment and increases the likelihood of Full Support judgments. In contrast, cluster-generation introduces an intermediate abstraction layer that groups nuggets into thematic clusters prior to synthesis. While this improves narrative coherence, it can compress fine-grained facts into broader statements, occasionally reducing sentence-level support scores.

Overall, support evaluation results demonstrate that both GenAIus pipelines produce responses with strong factual grounding within cited evidence, with nugget-generation achieving slightly higher citation alignment due to its preservation of atomic evidence granularity during response construction.

3.2 Relevance Judgment Task

For the relevance judgment task, we submitted five runs, described below.

- **nugget_cnt**: Score each passage based on the number of generated nuggets.
- **norm_nugget_cnt**: Score each passage based on the number of generated nuggets normalized by passage length.
- **cluster_cnt**: Score each passage based on how many of its nuggets are included in the clusters.
- **unique_cluster_cnt**: Score each passage by the number of distinct clusters in which its nuggets appear.
- **citation_cnt**: Score each passage according to how many times it is cited in the final response.

In this task, documents are graded on a 0–4 relevance scale, and system labels are compared against human judgments to measure alignment reliability. Evaluation is reported using agreement fraction and Cohen's kappa, capturing both raw consistency and agreement beyond chance.

As summarized in Table 3, the nugget_cnt, cluster_cnt, unique_cluster_cnt, and citation_cnt runs each achieved an agreement fraction of 0.28, while the norm_nugget_cnt run achieved 0.26. This stability suggests that nugget-derived evidence signals, whether measured directly, structurally clustered, or reflected through citation usage, produce comparable relevance labeling behavior when mapped to the narrative query scale.

Overall, the results demonstrate that nugget-centric evidence modeling provides a stable and interpretable proxy for document relevance. Direct nugget frequency emerged as the most reliable indicator of alignment with human judgments, while clustering and citation-based variants offered complementary but slightly less discriminative signals. These findings support the effectiveness of nugget extraction as a foundational representation for automated relevance assessment in narrative-driven retrieval settings.

4 Conclusion

In this paper, we presented GenAIus's participation in the TREC 2025 RAG Track, focusing on the augmented generation and relevance judgment tasks. We introduced two LLM-driven generation

pipelines built on a shared nugget extraction foundation: a nugget-based approach that directly conditions responses on atomic evidence units, and a cluster-based approach that organizes nuggets into thematic structures prior to synthesis.

While both pipelines achieved competitive performance, the nugget-generation approach consistently demonstrated modest advantages in nugget recall, sub-narrative coverage, and citation support, suggesting that preserving fine-grained factual granularity improves measurable grounding and coverage. We further extended the nugget framework to the relevance judgment task, proposing five automated passage ranking strategies derived from nuggets, clusters, and citation signals.

Overall, our work demonstrates that transforming retrieved passages into structured evidence units offers a flexible foundation for both response generation and relevance estimation. These findings underscore the value of intermediate knowledge representations in improving interpretability, grounding, and evaluation alignment in retrieval-augmented generation systems.

References

[1] Ronak Pradeep, Nandan Thakur, Sahel Sharifymoghaddam, Eric Zhang, Ryan Nguyen, Daniel Campos, Nick Craswell, and Jimmy Lin. Ragnarök: A reusable rag framework and baselines for trec 2024 retrieval-augmented generation track. In *European Conference on Information Retrieval*, pages 132–148. Springer, 2025.

[2] Nandan Thakur, Ronak Pradeep, Shivani Upadhyay, Daniel Campos, Nick Craswell, Ian Soboroff, Hoa Trang Dang, and Jimmy Lin. Assessing support for the trec 2024 rag track: A large-scale comparative study of llm and human evaluations. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2759–2763, 2025.

[3] Shivani Upadhyay, Nandan Thakur, Ronak Pradeep, Nick Craswell, Daniel Campos, and Jimmy Lin. Overview of the TREC retrieval-augmented generation (rag) track. In *Proceedings of the 34th Text REtrieval Conference (TREC 2025)*, 2025.