Submodular Evidence Selection for Grounded Answer Generation in TREC RAG 2025

Zizhen Li
National University of Singapore

Hai-Tao Yu
University of Tsukuba

Abstract

This paper describes and analyzes four submissions to the TREC 2025 Retrieval-Augmented Generation Answer Generation (AG) task. All runs use the organizer-provided top-100 retrieved segments and differ only in evidence selection, answer construction, and post-hoc refinement. The primary system combines greedy submodular evidence selection, evidence-card compression, and citation-first answer generation, while three companion runs test post-hoc rewriting and a lighter concatenation-style baseline. On the organizer's AG scores, the primary system family achieves the strongest coverage-oriented results, reaching a strict vital score of 0.35 and sub coverage of 0.51, compared with 0.18 and 0.37 for the unrefined concatenation baseline. Adding a refiner to the primary system preserves those two scores while improving weighted precision and weighted recall from 0.578 to 0.690. A refiner applied to the concatenation baseline reaches the highest weighted scores in the released package, but on weaker strict vital and sub coverage. Taken together, the results suggest that upstream evidence selection and grounding decisions matter more than surface-level rewriting for strong AG performance.

1 Introduction

The TREC RAG 2025 track studies systems that combine retrieval and generation for open-domain question answering and analysis over MS MARCO v2.1 [6]. This paper focuses on the Answer Generation (AG) task, where retrieval is fixed in advance: for each narrative topic, the system receives up to 100 candidate segments and must return a concise, sentence-level answer with explicit citations to the supplied evidence [6]. The task is deliberately narrow in one sense and demanding in another. Because retrieval is shared, success depends less on finding documents and more on selecting complementary evidence, compressing it effectively, and turning it into grounded prose under a hard length budget.

That setting makes AG a useful testbed for a simple question: when retrieval is held constant, what matters more for final answer quality, better upstream evidence organization or better downstream rewriting? The four submitted runs were designed to answer that question. Two runs use a structured pipeline built around submodular segment selection and citation-first generation. Two companion runs act as lighter baselines, relying more heavily on concatenation-style construction and less on model-based evidence compression.

The released evaluation package lets us revisit those design choices with concrete scores. The main story is clear. Structured evidence selection is the strongest lever for coverage-focused metrics, while post-hoc refinement is most helpful when it operates on an already well-grounded answer. The rest of the

paper describes the system, reports the organizer scores, and analyzes what the run comparisons imply about answer generation design.

2 Method

For the AG task, retrieval is treated as fixed. The system operates only over the organizer-provided top-100 candidate segments for each narrative query. The primary pipeline has three stages:

1. Select diverse, high-utility evidence with a greedy submodular objective;
2. Compress selected segments into short evidence cards;
3. Generate citation-first claims, then tighten and repair attribution.

2.1 Evidence Selection via a Submodular Objective

Let $\mathcal{P}$ denote the set of up to 100 candidate segments for a topic. The system selects $K{=}24$ segments with a greedy objective that balances *coverage*, *relevance*, and *domain diversity*. This combination encourages the system to gather complementary evidence while still prioritizing segments that are closely related to the narrative query.

For a narrative query q, let $\mathbf{t}(q)$ be its lightly stemmed token list; for a candidate segment d, let $\mathbf{t}(d)$ be its token list. Define unigram, bigram, and trigram sets $U(\cdot)$, $B(\cdot)$, and $T(\cdot)$.

At selection step s, let C_s be the set of already selected segments.

Coverage. Over the candidate pool $\mathcal{P}$, let x denote an n-gram token. We define the pool document frequencies as

$$\mathrm{df}_U(x) = \Big|\{d \in \mathcal{P} : x \in U(d)\}\Big|, \quad \mathrm{df}_B(x) = \Big|\{d \in \mathcal{P} : x \in B(d)\}\Big|, \quad \mathrm{df}_T(x) = \Big|\{d \in \mathcal{P} : x \in T(d)\}\Big|. \tag{1}$$

Pool-IDF weights are then

$$w_U(x) = \log\frac{|\mathcal{P}|+1}{\mathrm{df}_U(x)+1}, \quad w_B(x) = \log\frac{|\mathcal{P}|+1}{\mathrm{df}_B(x)+1}, \quad w_T(x) = \log\frac{|\mathcal{P}|+1}{\mathrm{df}_T(x)+1}. \tag{2}$$

After selecting C_s, define the covered sets

$$\widehat{U}_s = \bigcup_{d\in C_s} U(d), \quad \widehat{B}_s = \bigcup_{d\in C_s} B(d), \quad \widehat{T}_s = \bigcup_{d\in C_s} T(d). \tag{3}$$

The marginal coverage gain from adding a candidate d is

$$g_{\mathrm{cov}}(d \,|\, C_s) = \sum_{x\in U(d)\setminus\widehat{U}_s} w_U(x) + \sum_{x\in B(d)\setminus\widehat{B}_s} w_B(x) + \sum_{x\in T(d)\setminus\widehat{T}_s} w_T(x). \tag{4}$$

This coverage term encourages the selector to prioritize segments that introduce new information rather than repeatedly selecting segments that mention the same facts.

BM25 relevance. Let $\mathcal{D} = \{\mathbf{t}(d) : d \in \mathcal{P}\}$ denote the tokenized segment collection. A BM25 index is built over $\mathcal{D}$ using the standard probabilistic relevance formulation [4]

$$\mathrm{idf}(t) = \log\!\left(\frac{N - \mathrm{df}(t) + 0.5}{\mathrm{df}(t) + 0.5} + 1\right), \tag{5}$$

with K_1=1.2, B=0.75, and average document length $\overline{L}$. For query q and segment d,

$$\mathrm{BM25}(q, d) = \sum_{t \in \mathbf{t}(q)} \mathrm{idf}(t)\, \frac{f(t, d)\,(K_1 + 1)}{f(t, d) + K_1\!\left(1 - B + B\,\frac{L(d)}{\overline{L}}\right)}, \tag{6}$$

where $f(t, d)$ is term frequency and $L(d)$ is segment length. This relevance term ensures that selected segments remain closely aligned with the narrative query.

Domain diversity. To avoid over-sourcing from a single host, let $h(d)$ be the host of the URL associated with d, and let $n_s(h)$ be the number of selected segments from host h in C_s. The novelty bonus is

$$g_{\mathrm{nov}}(d \,|\, C_s) = \frac{1}{1 + n_s\big(h(d)\big)}. \tag{7}$$

This term discourages oversampling from a single host and promotes evidence from multiple sources. The relevance-plus-novelty intuition is similar to classic MMR-style diversification [1], while our implementation applies novelty at the host level inside the greedy selector.

Greedy objective. At each step $s = 0, \ldots, K - 1$, the system selects

$$d^{\star} = \underset{d \in \mathcal{P} \setminus C_s}{\arg\max}\; \underbrace{\alpha\, g_{\mathrm{cov}}(d \,|\, C_s)}_{\text{coverage}} + \underbrace{\beta\ \mathrm{BM25}(q, d)}_{\text{relevance}} + \underbrace{\gamma\, g_{\mathrm{nov}}(d \,|\, C_s)}_{\text{domain novelty}}. \tag{8}$$

In our runs, we set $\alpha = 1.0$, $\beta = 0.4$, and $\gamma = 0.2$. This produces a compact but deliberately non-redundant bundle of evidence.

2.2 Evidence Compression into Cards

Each selected segment is compressed into a one- or two-sentence evidence card of at most 55 words using a small-model pool consisting of gpt-5-nano and gpt-5-mini. The cards are intended to preserve factual content while stripping away local redundancy, boilerplate, and low-value phrasing. This makes it easier for the generator to stay under the AG length limit without losing citation locality. For the No-LLM runs (pipeline variants that remove LLM-based compression and generation), the same card interface is approximated with a rule-based compressor: from each selected segment, the system retains up to two sentences with the highest lexical overlap with the narrative query.

2.3 Citation-First Claim Generation and Tightening

Given the narrative query and the ordered evidence cards, a larger model generates 6-10 sentence-level claims in strict JSON format. Each sentence must cite one or more evidence cards.

A lightweight post-check then rescans the answer, measures lexical support against the evidence cards, repairs weak attributions, adjusts citations when necessary, and trims wording to satisfy the global word budget. The final output is a list of cited sentences together with the underlying segment references. In the No-LLM variant, this stage is replaced by heuristic claim assembly: the system extracts a small set of salient content words from the narrative, matches each one to the best unused evidence card by token overlap, and uses the highest-overlap clause from that card as the draft claim before optional refinement.

3 Submitted Runs

All four runs use the same organizer retrieval input but vary in how evidence is compressed and how much rewriting is allowed.

1. **Primary: Submodular → Cards → Citation-first.** This run implements the full pipeline described in Section 2: submodular evidence selection, LLM-based evidence-card compression, and citation-first claim generation with post-check tightening.
2. **Primary + Refiner.** A structure-preserving GPT-based refiner rewrites the primary output for fluency and concision while keeping sentence boundaries, citations, and references fixed.
3. **No-LLM Concatenation.** A lighter baseline that still uses the same submodular evidence bundle, but replaces most model-based generation steps with deterministic heuristics. In the implementation, each selected segment is converted into an evidence card by a rule-based compressor that chooses up to two sentences with the highest lexical overlap with the narrative query. Claim construction is likewise heuristic: the system extracts a small set of salient content words from the narrative, matches each one to the best unused evidence card by token overlap, and turns the highest-overlap clause from that card into a sentence-level claim. If too few claims are produced, the system pads the answer with the first sentence from additional cards. This baseline therefore preserves the same selected references as the main system, but uses much shallower sentence construction and usually only one citation per claim.
4. **No-LLM + Refiner.** The same concatenation-style baseline followed by the same refiner used in Run 2.

These runs isolate two main questions: whether structured evidence organization helps more than lighter answer construction, and whether post-hoc refinement can compensate for weaker upstream grounding. The No-LLM baseline is intentionally useful here because it shares the same evidence selector as the primary run, so its differences come mainly from rule-based compression and heuristic claim assembly rather than from poorer retrieval or a different candidate pool.

4 Evaluation Setup

The organizer package reports four AG metrics: strict vital score, sub coverage, weighted precision, and weighted recall. Retrieval metrics are not discussed here because retrieval was fixed for the AG task. As noted by the organizers, these AG scores were produced with the Waterloo nuggetizer. [3, 2].

Table 1: Overall AG scores from the organizer "all" rows. For the No-LLM run, weighted precision and weighted recall were not reported in the released organizer file.

Run	Strict vital	Sub coverage	W. precision	W. recall
Primary	**0.35**	**0.51**	0.578	0.578
Primary + Refiner	**0.35**	**0.51**	0.690	0.690
No-LLM	0.18	0.37	–	–
No-LLM + Refiner	0.17	0.34	**0.843**	**0.843**

These metrics capture two different aspects of answer quality. The first pair, strict vital score and sub coverage, are nugget-coverage metrics. In the nuggetizer framework, answers are matched against atomic facts, or nuggets, and the strict vital score gives credit only when a *vital* nugget is fully supported; partial support does not help on that strict metric [3, 2]. The released sub coverage score is the softer companion signal in the organizer package: it reflects broader topic coverage than the strict-vital metric and is therefore useful for distinguishing answers that mention many relevant facts from answers that capture only a small number of core nuggets.

The second pair, weighted precision and weighted recall, are citation-support metrics. In the TREC RAG support-evaluation protocol, each answer sentence is judged against cited evidence using full support, partial support, or no support, typically mapped to weights of 1.0, 0.5, and 0.0 [5]. Weighted precision asks how often cited sentences are genuinely supported by their citations, while weighted recall asks how much of the whole answer is supported. The support-evaluation study also notes that, under the sparse annotation protocol, only the first cited passage per sentence is judged; if essentially every sentence has at least one citation, weighted precision and weighted recall can become numerically identical [5]. That is exactly the pattern observed in the released files for our scored runs.

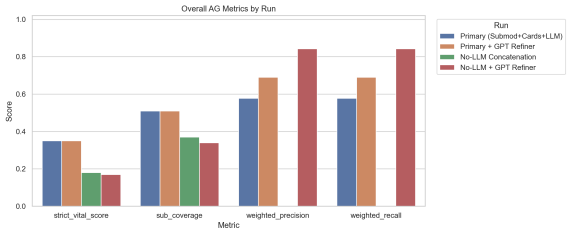
The released files also come with an important caveat. Per-topic coverage is not uniform across metrics: strict vital score and sub coverage are reported for 20 topics for each run, while weighted precision and weighted recall are reported for 14 topics for the two primary runs, 17 topics for *No-LLM + Refiner*, and not reported for the unrefined No-LLM run. In addition, weighted precision and weighted recall are numerically identical in the supplied files. For that reason, the cleanest four-way comparison is on strict vital score and sub coverage, while the weighted metrics are best used as secondary evidence.

5 Results and Analysis

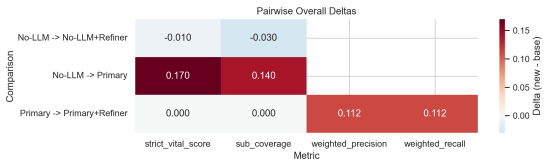
5.1 Overall Comparison

Table 1 and Figure 1 summarize the main results. The strongest pattern is that the primary architecture dominates the two coverage-oriented metrics. Both primary runs reach a strict vital score of 0.35 and sub coverage of 0.51, substantially ahead of the unrefined No-LLM baseline at 0.18 and 0.37.

A second pattern is that refinement has asymmetric value. On top of the primary pipeline, the refiner preserves strict vital score and sub coverage while raising weighted precision and weighted recall from 0.578 to 0.690. On top of the No-LLM baseline, however, refinement slightly reduces strict vital score and sub coverage. The released files assign the highest weighted scores to *No-LLM + Refiner* (0.843), but those values are computed on a different topic subset and therefore should be interpreted with caution rather than as a direct four-way win.



(a) Overall metric values for each run.



(b) Pairwise overall deltas.

Figure 1: Aggregate organizer results. The heatmap makes the main ablation story explicit: the primary pipeline gives the largest gains on strict vital score and sub coverage, while refinement helps weighted metrics most when the underlying answer is already strong.

Table 2: Mean output statistics over all 105 topics, computed from the submitted JSONL files.

Run	Words	Sentences	Cites / sent.	Ref. util.
Primary	264.3	8.61	1.92	0.55
Primary + Refiner	212.8	8.61	1.92	0.55
No-LLM	332.6	7.99	1.00	0.33
No-LLM + Refiner	202.0	7.99	1.00	0.33

The pairwise comparisons sharpen that picture. Relative to No-LLM, the primary system improves strict vital score by 0.170 and sub coverage by 0.140 on the organizer "all" rows. Relative to the unrefined primary run, the refiner adds 0.112 on both weighted metrics with no aggregate loss on the two coverage metrics. By contrast, the same refiner applied to No-LLM changes strict vital score by -0.010 and sub coverage by -0.030.

5.2 How the Outputs Changed

The score differences are consistent with clear differences in answer structure. Table 2 shows mean output statistics over all 105 topics. The primary family writes slightly more sentences, cites almost twice as many pieces of evidence per sentence, and uses a much larger fraction of the selected references. The refiner mostly shortens the wording while preserving citation behavior inside each run family. Figure 2 illustrates the distribution of answer length and citation density across the four runs.

This difference matters because the AG task rewards answers that cover multiple nuggets while remaining tightly supported. The primary pipeline appears to achieve that by spreading support across more evidence cards and more distinct references. The No-LLM variants are shorter on citation structure: they generally assign one citation per sentence and use only about one third of the selected references. That lighter evidence usage lines up with their lower strict vital score and sub coverage.

5.3 Topic-Level Behavior

The topic-level heatmaps in Figure 3 show that the primary pipeline is not just better on average; it wins on most of the evaluated topics. Relative to No-LLM, the primary run improves strict vital score on 14 of 20 topics and sub coverage on 12 of 20 topics. The largest strict-vital gains appear on topics 499 (+0.58), 897 (+0.48), and 213 (+0.39). The largest losses appear on topics 224 (-0.30), 58 (-0.25), and 219 (-0.20), which is a reminder that stronger aggregate performance still leaves room for topic-specific failure modes.

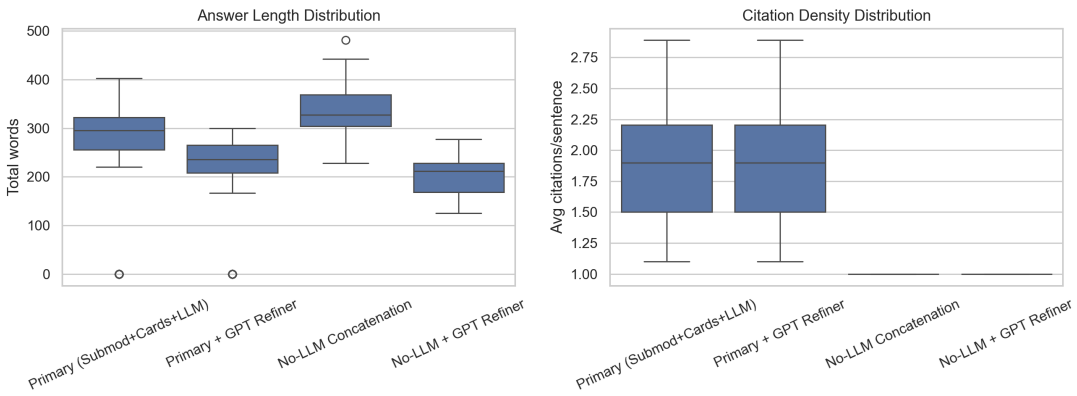
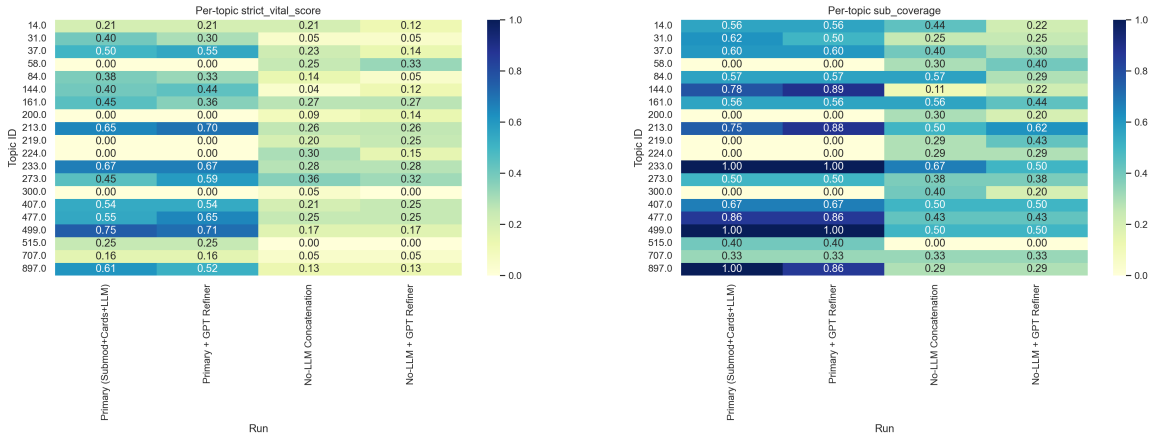


Figure 2: Distribution of answer length and citation density across the four runs. The refiners mostly compress wording, while the gap in citation behavior is driven by the underlying generation strategy rather than the refiner itself.



(a) Per-topic strict vital score.

(b) Per-topic sub coverage.

Figure 3: Topic-level coverage metrics. The main pattern is stable: the primary system family occupies the strongest band on most evaluated topics, while the No-LLM variants are more uneven.

Within the primary family, the refiner has a narrower but still meaningful effect. Figure 4 shows the per-topic score deltas between the primary system and its refined variant. It improves weighted precision on 7 of the 14 topics for which those values are reported and never hurts it on that shared set. The largest gains are on topics 407 (+0.35), 273 (+0.30), and 93 (+0.292). On strict vital score and sub coverage, however, the topic-level changes are mostly zero or small, which is exactly what one would hope from a refiner that is supposed to clean up wording without changing the evidence base.

The organizer median file gives a second way to understand topic difficulty. Topic 515 is the hardest by median strict vital score across all submitted runs in the package, while topic 499 is the easiest. For weighted precision, topic 224 is the hardest and topic 200 the easiest. These difficulty signals are useful because they show that some of the observed failures are not isolated to one system, but reflect genuinely challenging topics.

Figure 4: Per-topic deltas for the primary refiner. Weighted precision improves on half of the reported topics and never declines, while strict vital score and sub coverage mostly stay flat.

6 Discussion

The results support a simple interpretation. Better upstream evidence organization is more valuable than post-hoc rewriting alone. The primary pipeline does three things that the lighter baseline does not do as aggressively: it chooses evidence for complementarity rather than local relevance alone, it compresses evidence before generation, and it generates claims in a citation-first format. Those choices appear to pay off most clearly on the metrics that reward nugget coverage under strict grounding constraints.

The role of the refiner is more nuanced. When the underlying answer is already built from a broad, well-cited evidence set, rewriting helps. In our results, the primary refiner shortens the answer from 264.3 to 212.8 words on average without changing sentence count or citation density, and this is accompanied by a clear gain on the weighted metrics. When the underlying answer is built from thinner evidence usage, however, the same rewriting step cannot invent missing support. That likely explains why the No-LLM refiner compresses the answers successfully but does not recover the losses on strict vital score or sub coverage.

There are also limitations to the present analysis. First, the released AG files do not cover the same topic set for every metric, especially for weighted precision and weighted recall. Second, the package is based on nuggetizer scoring rather than final completed nugget assessments. Third, this paper evaluates only answer generation because retrieval was fixed by the task setup. Even with those caveats, the main conclusion is robust: for AG, better evidence curation and grounding matter more than polishing the final prose in isolation.

7 Conclusion

This paper presented four TREC RAG 2025 AG runs built around a common retrieval input and analyzed what the released organizer scores reveal about their design choices. The full submodular-and-citation-first pipeline was the strongest family on strict vital score and sub coverage, reaching 0.35 and 0.51, while the lighter No-LLM baseline reached 0.18 and 0.37. A refiner applied to the primary system improved weighted precision and weighted recall from 0.578 to 0.690 without hurting the coverage-oriented metrics, whereas the same refiner could not rescue the weaker baseline on those primary AG measures.

The broader lesson is straightforward. In this setting, answer quality is driven mainly by how evidence is selected, organized, and attached to claims. Once those decisions are solid, rewriting can help. Without them, rewriting mostly changes surface form. Future work should revisit these same system choices once final nugget assessments are available and test whether the same pattern holds for end-to-end RAG settings where retrieval is no longer fixed.

References

[1] Jaime Carbonell and Jade Goldstein. "The use of MMR, diversity-based reranking for reordering documents and producing summaries". In: *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 1998, pp. 335–336.

[2] Castorini. *Nuggetizer*. Scoring code and metric descriptions. Feb. 2025. URL: https://github.com/castorini/nuggetizer.

[3] Ronak Pradeep et al. *Initial Nugget Evaluation Results for the TREC 2024 RAG Track with the AutoNuggetizer Framework*. 2024. arXiv: 2411.09607 [cs.IR]. URL: https://arxiv.org/abs/2411.09607.

[4] Stephen Robertson and Hugo Zaragoza. *The probabilistic relevance framework: BM25 and beyond*. Vol. 4. Now Publishers Inc, 2009.

[5] Nandan Thakur et al. *Support Evaluation for the TREC 2024 RAG Track: Comparing Human versus LLM Judges*. 2025. arXiv: 2504.15205 [cs.CL]. URL: https://arxiv.org/abs/2504.15205.

[6] TREC. *What is TREC RAG? — trec-rag.github.io*. https://trec-rag.github.io/. [Accessed 05-03-2026]. 2025.