Do Nugget-Based Evaluation Patterns Generalize to List-QA?

MohammadJavad Ardestani
University of Alberta
ardestan@ualberta.ca

Ehsan Kamalloo
ServiceNow AI Research
ehsan@servicenow.com

Davood Rafiei
University of Alberta
drafiei@ualberta.ca

Abstract

Evaluating long-form answers from retrieval-augmented generation (RAG) systems remains challenging: human evaluation is expensive, while automatic metrics must reliably capture answer completeness. The AutoNuggetizer framework addresses this by decomposing evaluation into atomic facts (nuggets) and using LLMs for both nugget creation and assignment. The original study validated this approach on open-ended TREC RAG queries; however, it remains unclear whether the same cost–quality tradeoffs hold for structurally different tasks. We reproduce AutoNuggetizer on seven RAG systems over the QAMPARI list-QA benchmark, where answers consist of discrete entities and omissions are more directly measurable. Our results directionally reproduce the original findings: fully automatic evaluation preserves run-level rankings, assignment-only automation yields stronger agreement than end-to-end automation, and LLM-based assignment is highly concordant with human labels while being modestly stricter. These findings support the use of AutoNuggetizer for comparative evaluation beyond open-ended RAG, while also identifying systematic biases in automatic nugget creation and assignment.

1 Introduction

Evaluating long-form answers from retrieval-augmented generation (RAG) systems is difficult for two main reasons. First, equivalent facts may appear with different wording, paraphrase, or sentence structure, so entailment can occur without lexical overlap. Prior work has shown that lexical matching can substantially underestimate QA performance because generated answers may be semantically correct while differing from the reference surface form, a problem that becomes more pronounced with longer LLM-generated answers (Kamalloo et al., 2023). Our study is aligned with this broader motivation, but focuses on whether nugget-based evaluation behavior generalizes to list-QA, where correctness depends on recovering multiple answer entities and their associated support. Second, human evaluation is expensive, slow, and hard to scale, yet automatic metrics must still capture whether an answer contains the information a user needs. This challenge is particularly acute for *recall*, i.e., measuring completeness rather than just correctness, because omissions are harder to detect than hallucinations.

The difficulty of recall evaluation depends on the answer structure. For open-ended queries (e.g., "How did African rulers contribute to the triangle trade?"), good answers may cover overlapping aspects, and semantically related content can partially satisfy multiple information needs. In contrast, *list-QA* tasks require the system to enumerate all entities satisfying a given criterion (e.g., "Which films were directed by X?"). Each answer item is independently verifiable and none is substitutable for another, making list-QA a stringent testbed for recall evaluation: missed entities are more directly observable as recall errors. List-QA benchmarks like QAMPARI have accordingly been used to study recall completeness in long-form generation (Ardestani et al., 2025), motivating their use for validating evaluation frameworks.

The AutoNuggetizer framework. Pradeep et al. (2025) address the evaluation challenge by adapting *nugget-based evaluation*—a methodology from TREC Question Answering circa 2003—to modern RAG systems. The key idea is to decompose what constitutes a good answer into atomic facts called *nuggets*, then check which nuggets appear in each system's response. AutoNuggetizer automates this process using LLMs in three stages: (1) **nugget creation** extracts atomic facts from gold context; (2) **importance scoring** labels each nugget as *vital* (must be present) or *okay* (good to have); and (3) **assignment** classifies each nugget against a

Proceedings of the Fifth Workshop on Generation, Evaluation and Metrics (GEM), pages 1071–1081

system answer as `support`, `partial_support`, or `not_support`. The framework's value proposition is a spectrum of cost-quality tradeoffs: fully automatic evaluation enables cheap system comparison at scale, while hybrid approaches (e.g., human nuggets with automatic assignment) offer stronger alignment at moderate cost.

Research questions. Following the original paper, our reproduction evaluates their three research questions:

RQ1 For RAG systems, how well does the fully automatic nugget evaluation framework correlate with different manual nugget evaluation strategies?

RQ2 Does automating only nugget assignment result in stronger agreement with manual evaluations compared to fully automating the entire evaluation?

RQ3 Are there any noticeable differences between nugget assignments by humans versus LLMs?

These RQs define the framework's practical value proposition across cost-quality settings and structure the analysis throughout this paper.

Why reproduce on list-QA? The original study establishes that AutoNuggetizer can approximate human-based evaluation for open-ended TREC RAG queries, especially at the run level. However, open-ended queries allow multiple valid answer structures, and the original paper reports substantially weaker topic-level agreement—i.e., agreement when comparing automatic vs. manual scores on individual questions rather than aggregated across all questions—limiting the framework's usefulness for fine-grained debugging. List-QA provides a complementary test: each answer item is a discrete entity that either satisfies the question constraints or does not. By reproducing the same comparison structure on QAMPARI, we test whether AutoNuggetizer's cost–quality tradeoffs hold when recall errors are more directly enumerable and when the systems are built outside the TREC shared-task ecosystem.

Contribution. We evaluate the original three research questions (RQ1–RQ3) on seven RAG systems we built across three model families (Claude, Gemini, GPT-4o) and multiple retrieval configurations. Our findings confirm all three directionally: fully automatic evaluation preserves system rankings ($\tau \approx 0.71$–0.81), assignment-only automation yields stronger agreement than end-to-end automation, and LLM assignment shows high concordance with humans while being modestly stricter. To support full reproducibility, we release a complete reproduction package at https://github.com/MoJa-Ardestani/ReproNLP-2026, including the code, data, prompts, configurations, and scripts required to reproduce our results.

2 Related Work and Positioning

Nugget- and content-unit evaluation. The nugget-based evaluation methodology used by AutoNuggetizer has roots in TREC complex QA (Lin and Demner-Fushman, 2006b,a) and is closely related to the Pyramid method in summarization, which represents content using weighted summary content units (Nenkova and Passonneau, 2004). Later work explored lightweight and automated pyramid construction (Shapira et al., 2019; Gao et al., 2019). This methodological lineage—trading off annotation cost against evaluation fidelity—motivates our interest in whether AutoNuggetizer's agreement patterns generalize across task formats.

Long-form QA and list-QA benchmarks. Recent long-form QA work emphasizes that answer quality depends not only on factual support but also on coverage of multiple relevant facts. ASQA frames long-form answers as summaries that resolve ambiguous factoid questions (Stelmakh et al., 2022), while ALCE places QAMPARI alongside ASQA and ELI5 in a benchmark for citation-grounded long-form generation (Gao et al., 2023). Our choice of QAMPARI for this reproduction is motivated by its distinctive answer structure: the set-valued format makes recall errors directly enumerable, providing a stringent test of whether nugget-based evaluation patterns generalize beyond open-ended QA.

Automatic evaluation of RAG and long-form generation. Automatic RAG evaluation frameworks such as RAGAS and ARES emphasize context relevance, answer relevance, and faithfulness (Es et al., 2024; Saad-Falcon et al., 2024). In parallel, FActScore decomposes long-form generations into atomic facts to measure factual precision against external knowledge (Min et al., 2023). Our focus is different: rather than evaluating factual precision or component quality in isolation, we test

whether a nugget-based evaluator preserves human agreement patterns when transferred across answer formats, with particular attention to recall and completeness in list-QA.

LLM judges and reproducibility. This work also relates to the broader LLM-as-judge literature. G-Eval showed that strong LLM judges can better align with human judgments than earlier automatic metrics, while later work documented substantial judge biases and showed that diverse judge panels can reduce intra-model bias (Liu et al., 2023; Chen et al., 2024; Verga et al., 2024).

3 Scope of Reproduction

3.1 Original Experimental Setup

The original evaluation is grounded in the TREC 2024 RAG Track, comprising 301 topics (i.e., queries or questions) with community-submitted runs over the MS MARCO passage collection. The primary evaluation metrics are V_{strict}—the fraction of *vital* nuggets receiving a full `support` label—and A_{strict}—the fraction of *all* nuggets fully supported. V_{strict} serves as the primary metric, as vital nuggets represent facts that must be present in a good response; `partial_support` counts as no credit under both metrics. Due to resource constraints, NIST annotators evaluated only the two highest-priority runs per group, resulting in 56 of the 301 topics being fully annotated. This annotated subset enables comparison across conditions that vary the nugget source (automatic, post-edited, or manual) and the assignment method (automatic vs. manual).

3.2 Adaptations of the Original Setup

We matched the original framework as closely as possible given available documentation. A full side-by-side summary of setting differences is provided in Appendix Table 5.

Task domain. The original study validated AutoNuggetizer solely on open-ended TREC queries, leaving open whether its cost-quality tradeoffs hold for structurally different task types. We deliberately chose a different domain to probe this generalizability. List-QA is a natural stress-test: answer items are discrete, independently verifiable entities, making recall assessment more demanding than in open-ended QA. QAMPARI (Amouyal et al., 2023) is an established benchmark for this format and is well-suited to evaluate whether the framework's *relative* findings transfer beyond the original TREC setting.

Codebase. We implemented the three-stage framework following the paper's prompts and algorithmic descriptions, using the public OPEN-RAG-EVAL codebase (Vectara, 2025) as a code reference. Stage 1 processes context in batches of 10 segments per LLM call with iterative nugget list updates, matching Figure 2. Stages 2 and 3 process nuggets in batches of 10 with the same label sets (vital/okay; support/partial/not_support). Parameter settings follow the paper and are summarized in Table 5. Temperature was set to 0.0, and a fixed random seed of 42 was used.

Annotation. All human nugget creation and assignment were performed by the authors following the original guidelines; no external annotators were recruited and no compensation was involved. Full details are in Appendix B.

4 Experimental Setup

4.1 Dataset

QAMPARI is a list-QA benchmark where each question requires exhaustive enumeration of entities satisfying a criterion. Questions fall into three types of increasing reasoning complexity: *simple* questions involve a single entity–relation lookup (e.g., "What song was created by Vishal Chandrasekhar?"); *intersection* questions require the answer set to satisfy two independent relations simultaneously (e.g., "Which TV programme had both X and Y as screenwriters?"); and *composition* questions chain two relations, requiring multi-hop evidence (e.g., "What is the height of buildings located in Dubai?"). From the 1,000 test samples, we apply preprocessing to remove superficial variations such as duplicate passages.

We filter for substantive answers (300–900 words in the gold reference) requiring multi-entity responses with supporting evidence, yielding approximately 450 questions distributed across question types: simple (40%), intersection (26%), and composition (34%). We used the 300–900 word range as a pre-specified filter to exclude two extremes: very short references that often contain too little evidence for meaningful nugget extraction, and very long references that can make the task closer to document summarization than list-QA recall evaluation. The range was selected be-

Code	Condition definition
A/A	AUTONUGGETS/AUTOASSIGN
E/M	AUTONUGGETS+EDITS/MANUALASSIGN
M/M	MANUALNUGGETS/MANUALASSIGN
E/A	AUTONUGGETS+EDITS/AUTOASSIGN
M/A	MANUALNUGGETS/AUTOASSIGN

Table 1: Condition notation. The first letter denotes the nugget source; the second denotes the assignment method (A = Auto, E = Edited, M = Manual).

fore manual annotation and was not tuned based on agreement results. This produced a sufficiently large candidate pool while retaining questions with enough evidence to support multi-entity nugget creation. From this pool, we draw 100 questions via stratified sampling, randomly selecting from each question-type split to preserve the original distribution.

For manual evaluation, we selected 50 topics from the 100-topic evaluation set. To stress-test agreement on harder cases, we intentionally undersampled simple questions: 20% of topics were simple, and the remaining were sampled from the intersection and composition categories. Within each category, topics were selected uniformly at random using the same fixed seed used for dataset construction.

4.2 RAG Systems

We evaluate seven RAG systems using a shared retrieve-then-generate pipeline. For each question, Wikipedia passages are retrieved via the Wikipedia API and truncated to approximately 200 words each; the retrieved passages are then injected into a structured prompt that instructs the model to exhaustively enumerate all relevant entities in the answer. The three model backends are GPT-4o (OpenAI, 2024), Gemini 2.0 Flash (Google DeepMind, 2024), and Claude Sonnet 4 (Anthropic, 2025). We run each backend at top-4 and top-8 passage retrieval budgets, yielding six systems, plus one additional top-12 using only GPT-4o, which forms the seven systems in total. This setup spans three model families and two retrieval depths, providing sufficient coverage to test whether the original paper's conclusions transfer across backbone LLMs and context sizes. The full prompts, retrieval parameters, model identifiers, and generated outputs are released in our reproduction repository.

4.3 Descriptive Statistics and Task-specific Patterns

Table 2 presents descriptive statistics for nugget creation. Our statistics differ from the original paper's Table 4 in three ways, which reflects the structural difference between QAMPARI's list-QA format and the open-ended QA in the TREC setting:

Nugget Type	#Top.	Avg Nug.	Avg Len (words)	%V	%O
Auto	50	6.6	7.4	63.5	36.5
Auto + Edited	50	7.2	9.1	80.8	19.2
Manual	50	7.2	10.9	80.1	19.9
Auto	100	7.6	7.2	71.6	28.4

Table 2: Descriptive statistics for nugget creation. The first three rows compare Auto, Auto + Edited, and Manual nuggets on the same 50 topics; the last row reports Auto nuggets on the full 100-topic set. %V = percentage of vital nuggets (must be present in a good answer); %O = percentage of okay nuggets (relevant but non-essential detail).

Nugget count convergence. All three nugget sources cluster at 6.6–7.2 nuggets per topic, compared to the original paper's roughly 14–19 nuggets per topic. This convergence reflects QAMPARI's list-QA structure: each question has a finite set of correct entity answers, so nugget count is relatively bounded by list length rather than answer verbosity.

Higher vital rate. Human-curated conditions (edited and manual) show ∼80% vital nuggets, compared with 59–72% in the original paper. In our scheme, nuggets must substantively engage with the question, so tangential context is excluded. A nugget is *vital* if it satisfies a required answer constraint, and *okay* if it contributes relevant but non-essential detail. Editing shifted the distribution toward vital: weak auto-generated okay nuggets were removed, and missing answer-critical nuggets—mostly vital—were added from the gold context. We analyze the specific failure patterns driving these edits in Section 6.1.

Longer human-curated nuggets. Within QAMPARI, edited and manual nuggets are longer than auto nuggets (9.1 and 10.9 words vs. 7.4 words). Across datasets, QAMPARI's human-curated nuggets are also longer than TREC's (9.1–10.9 vs. 7.0–8.5 words in the original paper), while auto nuggets are comparable (7.4 vs. 6.8 words). We attribute this to QAMPARI's complex question types (intersection and composition), which

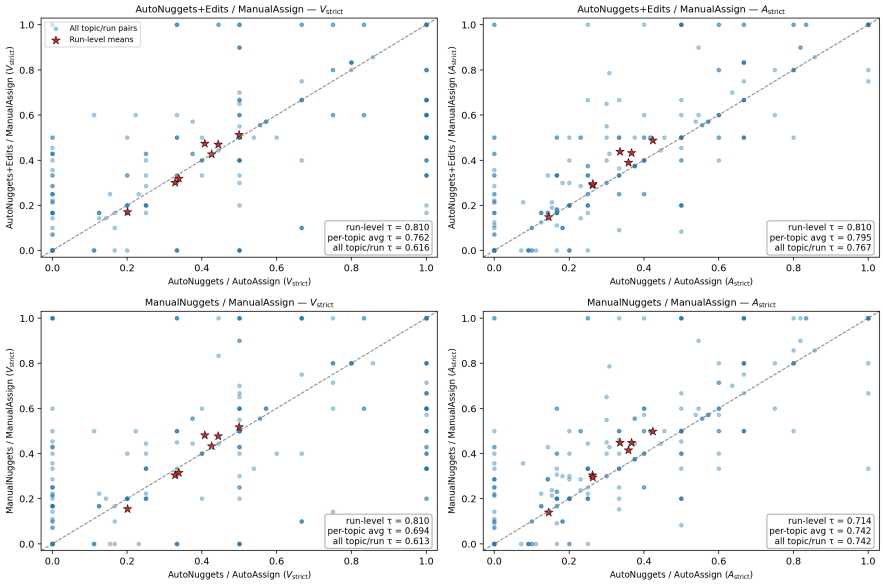

Figure 1: RQ1 analysis: **A/A** vs. manual variants. Top: comparison against **E/M**. Bottom: comparison against **M/M**. Left: V_{strict}; Right: A_{strict}.

require nuggets to encode multiple constraints that human annotators capture more completely than the automatic method.

4.4 Evaluation Protocol

We analyze five conditions corresponding to the original paper's evaluation strategies, using the notation in Table 1, and report both V_{strict} and A_{strict}. Following the original paper, we use Kendall's τ to measure ranking agreement between evaluation conditions (RQ1, RQ2) and confusion matrices for label-level behavior (RQ3). For RQ1, we compare fully automatic evaluation (**A/A**) against manual baselines (**E/M**, **M/M**); for RQ2, we compare assignment-only automation (**E/A** vs. **E/M**, **M/A** vs. **M/M**) to isolate the effect of automatic assignment. Kendall's τ is computed at three granularities: **run-level**, where per-topic scores are averaged before ranking systems; **per-topic average**, where correlations are computed per topic then averaged; and **all topic/run**, where correlation is computed over all pooled observations. For RQ3, we align AUTOASSIGN and MANUALASSIGN labels on identical nugget sets, reporting diagonal agreement and marginal label rates. All LLM components were run with temperature 0.0, and fixed seeds were used where applicable; each configuration was executed once.

Because this reproduction changes both the evaluation object (TREC open-ended QA vs. QAMPARI list-QA) and the system pool, we do not report CV* scores and instead treat the comparison as a generalization-oriented reproduction (Belz, 2025), reporting directional agreement over the matched RQ structure rather than claiming scale-matched reproducibility of absolute scores.

5 Results

5.1 RQ1: Fully Automatic Evaluation

Figure 1 compares fully automatic evaluation (**A/A**) against manual variants (**E/M** and **M/M**), testing whether automatic scoring can serve as a reliable surrogate for system ranking.

Finding. The results confirm RQ1. Following the original paper, run-level Kendall's τ is the primary evidence for this claim: high run-level agreement indicates that fully automatic evaluation can serve as a surrogate for system ranking. We observe run-level τ of 0.810 for both V_{strict} and A_{strict} against **E/M**, and 0.810 (V_{strict}) / 0.714 (A_{strict}) against **M/M**—comparable to the original paper's 0.887/0.901 and 0.727/0.758, respectively.

Per-topic average correlations ($\sim$0.69–0.80) and all-topic/run correlations ($\sim$0.61–0.77) corroborate the run-level finding at finer granularities. All τ values remain in a range considered sufficient to validate an evaluation metric in IR meta-evaluations.

5.2 RQ2: Assignment-Only Automation

RQ2 tests whether automating only the assignment step, while holding nuggets fixed, yields stronger

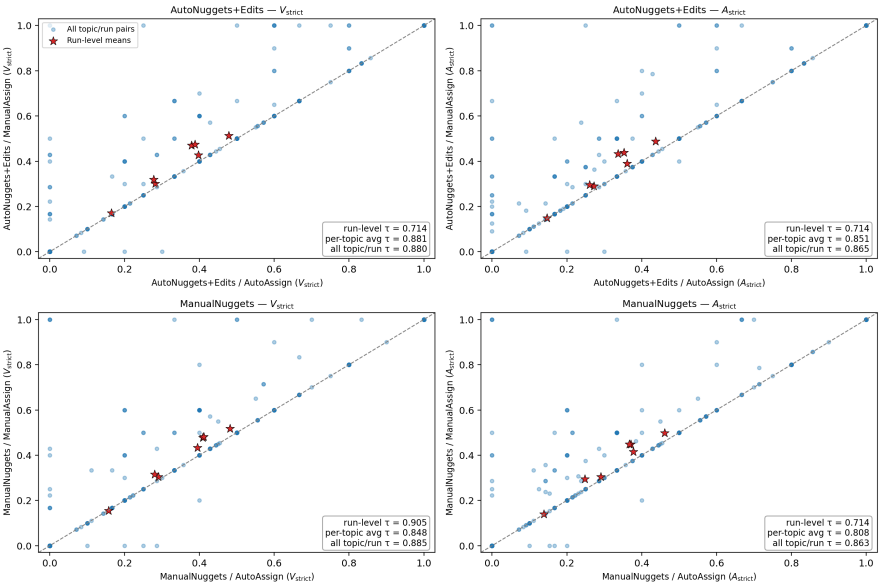

Figure 2: RQ2 analysis: isolating assignment automation. Top: **E/A** vs. **E/M**. Bottom: **M/A** vs. **M/M**. Left: V_{strict}; Right: A_{strict}.

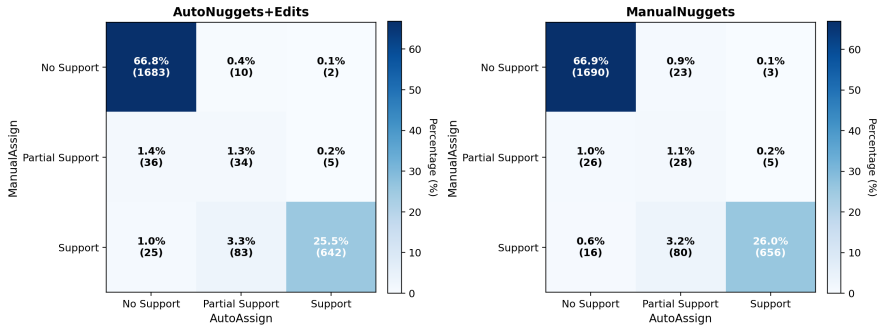

Figure 3: RQ3 analysis: confusion matrices comparing AUTOASSIGN (rows) against MANUALASSIGN (columns).

agreement with manual evaluation than end-to-end automation. Figure 2 isolates the effect of assignment automation.

Finding. The results confirm the hypothesis in RQ2. Following the original paper, per-topic average and all-topic/run τ are the primary evidence for this claim, as they capture agreement across individual queries rather than at the aggregate run level. When nuggets are fixed (edited or manual), replacing MANUALASSIGN with AUTOASSIGN yields substantially stronger topic-level agreement than end-to-end automation. For V_{strict}, per-topic average τ reaches 0.848–0.881 under assignment-only automation, compared with 0.694–0.762 under full automation (RQ1). All-topic/run τ shows the same pattern: 0.880–0.885 vs. 0.613–0.616 for V_{strict}. The improvement holds across both metrics and both nugget-source conditions; for A_{strict}, per-topic average τ reaches 0.808–0.851 (vs. 0.742–0.795 under RQ1).

This mirrors the original paper: for V_{strict}, per-topic average τ rose from 0.36–0.49 under full automation to 0.65–0.66 with assignment-only automation, with similar all-topic/run gains. Run-level τ stayed comparable to RQ1 (0.714–0.905 vs. 0.714–0.810), reflecting the paper's finding that improvements were modest at run level but stronger at finer-grained levels.

5.3 RQ3: Assignment Behavior

RQ3 examines behavioral patterns rather than ranking agreement. The original paper argued that AUTOASSIGN agrees substantially with MANUALASSIGN on `support` and `not_support`, but uses `partial_support` more frequently and is slightly stricter overall. Table 3 and Figure 3 present our reproduction.

The results reproduce the original behavioral pattern. Diagonal agreement exceeds 93% in both

Statistic	Edited	Manual
Label pairs	2520	2527
Diagonal agreement	93.6%	94%
Auto partial_support rate	5%	5.2%
Human partial_support rate	2.9%	2.3%
Ordinal strictness	−0.060	−0.041

Table 3: RQ3 aggregate behavior.

Claim	Status
RQ1: Fully automatic evaluation preserves run-level rankings.	directionally verified
RQ2: Assignment-only automation yields stronger agreement than end-to-end automation.	directionally verified
RQ3: AUTOASSIGN shows high concordance with MANUALASSIGN but is systematically stricter and uses `partial_support` more frequently.	directionally verified

Table 4: Claims verification summary. All claims are confirmed directionally; absolute values differ due to task domain.

conditions, indicating strong concordance. The marginal distributions reveal the same asymmetry: AUTOASSIGN uses `partial_support` more frequently than human annotators (5% vs. 2.9% for edited nuggets; 5.2% vs. 2.3% for manual). We compute ordinal strictness as the mean difference between automatic and human labels after mapping `not_support`=0, `partial_support`=1, and `support`=2; negative values indicate stricter automatic assignment. The ordinal strictness statistic is negative in both cases, confirming slightly stricter automatic labeling. The two nugget-source conditions produce confusion matrices of similar shape, supporting the original claim that LLM-generated draft nuggets do not radically alter assignment-level patterns. The edited and manual conditions show the same qualitative behavior, differing only in magnitude.

Summary. Taken together with RQ1 and RQ2, these results confirm all three research questions directionally (Table 4): fully automatic evaluation preserves run-level rankings, assignment-only automation yields stronger agreement than end-to-end automation, and LLM-based assignment shows high concordance with human labels while being modestly stricter.

6 Error Analysis

6.1 AUTONUGGETS Failures

We examined all 50 manually annotated topics (the subset with both automatic and human-curated nuggets; see Appendix B) for post-auto nugget changes. Because a single topic can require multiple edit types simultaneously (e.g., both adding missing nuggets and removing irrelevant ones), we report per-topic incidence rates that sum to more than 100%: `no_edit` 32%, `add_nugget` 34%, `remove_nugget` 34%, `major_edit` 28%, `minor_edit` 16%. We use `major_edit` for substantive revision of existing nuggets—rewording for clarity, merging or splitting to fix granularity, or inserting answer-critical information that the draft nuggets omitted—rather than simple surface tweaks. Three failure modes account for the majority of substantive edits.

Incomplete entity enumeration. The most common failure is under-extraction: AUTONUGGETS generates nuggets for some but not all entities satisfying the question's constraints. This is particularly pronounced for intersection questions, where multiple constraints must be jointly satisfied. For example, given *"What film directed by M. Krishnan Nair had its music composed by V. Dakshinamoorthy?"*, auto-nuggetization produced a single nugget *"Music composed by V. Dakshinamoorthy"* rather than enumerating the specific films. This pattern—extracting the relation rather than the entities—accounts for the largest vital-count increases during editing.

Relation fragmentation. For complex questions, AUTONUGGETS sometimes splits a single answer into multiple nuggets that each capture only part of the required relation. On *"What film has V. Ravichandran as director and Hamsalekha as composer?"*, auto-nuggetization produced two separate nuggets: *"Nattukku Oru Nallavan directed by V. Ravichandran"* and *"Nattukku Oru Nallavan music composed by Hamsalekha."* Neither nugget alone answers the intersection question; human editors merged them into a single nugget encoding both constraints. This fragmentation inflates nugget counts without improving coverage.

Non-answer background extraction. AUTONUGGETS occasionally extracts contextual facts that do not directly answer the question. These nuggets are typically labeled *okay* (contributing

relevant but non-essential detail) but dilute the evaluation signal. For instance, on a question about Continental Navy members, auto-nuggetization extracted biographical facts about naval officers that do not establish membership. Human editors removed these tangential nuggets, simultaneously reducing the okay share and tightening the evaluation focus.

Implications. These failures explain the statistical patterns in Table 2: the +17 percentage-point vital rate increase from auto to edited nuggets reflects systematic under-extraction of answer-critical entities (corrected by addition) and over-extraction of tangential context (corrected by removal). The convergence of edited and manual nugget counts (7.2 each) suggests that human curation—whether post-editing or from-scratch—arrives at similar granularity for list-QA, while the auto method's lower count (6.6) reflects incomplete enumeration rather than appropriate compression. For practitioners, these patterns indicate that AUTONUGGETS may *inflate* absolute recall scores—fewer nuggets means a lower coverage bar—though relative system rankings remain valid if the bias affects all systems similarly.

6.2 AUTOASSIGN Failures

We examined the ∼45 harsh conflicts (`support` vs. `not_support`) across all systems. Roughly 85% are *false negatives* (human marks support, framework rejects); 15% are *false positives* (framework marks support, human rejects).

Missing secondary details (∼50% of False Negatives). The passage supports the core answer but omits a secondary detail in the nugget. For example, on *"Who was a member of the Continental Navy?"*, a generated answer may name John Paul Jones as a Continental Navy officer without stating that he joined in 1775. The nugget "John Paul Jones joined the Continental Navy in 1775" is marked `not_support` because the year is absent, while humans mark `support`—membership is what the question asks; the date is not a core part of the nugget for this question, yet the pipeline treats it as a conjunctive requirement.

Unstated constraints (∼25% of False Negatives). The passage supports a relation at a coarse level; the nugget adds a tighter qualifier. For example, on *"Who directed TV programs written by John Swartzwelder?"*, a generated answer may state that David Silverman directed numerous *The Simpsons* episodes written by Swartzwelder, without specifying that any was the *first*. The nugget "David Silverman directed the first episode written by John Swartzwelder" is marked `not_support` because the ordinal is absent, while humans accept the documented director–writer link as sufficient.

Name variants (∼15% of False Negatives). The pipeline struggles with name variants. A passage stating "Geoffrey Bruce was a climbing member of the 1924 expedition" is marked `not_support` for the nugget "John Geoffrey Bruce was part of the 1924 Everest expedition"—the full-name vs. short-name mismatch causes rejection despite clear semantic equivalence.

False positives. AUTOASSIGN can over-credit answers by drawing on parametric knowledge—facts the LLM learned during pretraining rather than evidence in the passage—instead of strict passage–nugget entailment, or by matching surface form while ignoring binding constraints. For example, a passage supporting a *2006* Pilot Pen Tennis victory was marked `support` for a nugget claiming the *2005* title—same players and event, but the year is an essential identifier of the specific tournament edition, so this is a real mismatch.

Implications. AUTOASSIGN operates with a conservative bias: it under-credits answers requiring alias resolution or unstated detail, but occasionally over-credits when parametric associations fill gaps. For comparative evaluation, this conservative bias may be less damaging than systematic over-crediting, because it avoids rewarding unsupported answers. However, it can still underestimate absolute recall, especially for answers involving aliases or compressed phrasings.

7 Discussion

Why is agreement stronger in list-QA? Per-topic agreement in our study consistently exceeds the original paper's values (e.g., per-topic average τ of 0.85 vs. 0.65 for assignment-only automation). We attribute this to QAMPARI's bounded answer structure: each question has a finite set of correct entities, reducing within-topic variance compared to open-ended queries where answers may cover variable subsets of relevant information. This suggests AutoNuggetizer may be particularly effective for evaluation tasks with clear, enumerable answer

sets—a promising signal for domains like fact verification or structured knowledge extraction.

Implications for practitioners. The hybrid strategy—human-curated nuggets with automatic assignment—emerges as an attractive middle ground because it automates the most labor-intensive annotation stage while preserving high agreement with fully manual evaluation (>93%). The error analysis (Section 6) reveals that the two automation stages have opposing biases: AUTONUGGETS under-extracts answer-critical entities, which inflates recall by lowering the coverage bar; AUTOASSIGN under-credits answers requiring alias resolution, which deflates recall but preserves validity. These opposing tendencies may help explain why fully automatic evaluation still preserves relative system rankings, but they also caution against interpreting absolute recall scores without calibration.

8 Limitations

This study is a generalization-oriented reproduction rather than a scale-matched replication. Our evidence comes from seven RAG systems on QAMPARI rather than a community-wide shared-task pool; we do not claim scale-matched reproduction of absolute values. QAMPARI's list-QA structure differs from open-ended QA in the TREC setting, affecting descriptive statistics (nugget counts, vital rates) while leaving core agreement patterns intact.

9 Conclusion

We reproduced the AutoNuggetizer framework on seven RAG systems over the QAMPARI list-QA benchmark, confirming all three research questions (RQ1–RQ3) in a structurally different domain. Notably, per-topic agreement is often *stronger* than in the original study, likely because bounded answer structures reduce within-topic variance. Our error analysis identifies opposing biases in the two automation stages that partially offset each other during fully automatic evaluation, motivating the hybrid strategy—human-curated nuggets with automatic assignment—as a practical middle ground that balances annotation cost against evaluation fidelity.

Ethics Statement

This reproducibility study uses only author annotations and does not involve additional human subjects. We report limitations transparently. This work was conducted as part of the ReproNLP 2026 shared task (Belz et al., 2026). A completed Human Evaluation Datasheet (HEDS) (Shimorina and Belz, 2022; Belz and Thomson, 2025) was submitted separately to the organizers and will be archived in the ReproNLP HEDS repository.[1]

Acknowledgment

This research was supported by the Natural Sciences and Engineering Research Council of Canada.

References

Samuel Joseph Amouyal, Tomer Barak, Akari Asai, Hiroaki Ohashi, Tomer Wolfson, Jonathan Berant, and Amir Globerson. 2023. QAMPARI: An open-domain question answering benchmark for questions with many answers from multiple paragraphs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4693–4703.

Anthropic. 2025. Claude sonnet 4. Model ID: claude-sonnet-4-20250514, Accessed via Anthropic API.

MohammadJavad Ardestani, Ehsan Kamalloo, and Davood Rafiei. 2025. LongRecall: A structured approach for robust recall evaluation in long-form text. *arXiv preprint arXiv:2508.15085*.

Anya Belz. 2025. QRA++: Quantified reproducibility assessment for common types of results in natural language processing. *arXiv preprint arXiv:2505.17043*.

Anya Belz and Craig Thomson. 2025. The human evaluation datasheet 3.0: Formal data documentation for human evaluation of natural language generation systems. In *Proceedings of the Fifth Workshop on Generation, Evaluation and Metrics (GEM)*.

Anya Belz, Craig Thomson, and Javier González Corbelle. 2026. The shared task on reproducibility of evaluations in NLP (ReproNLP) 2026: Overview and results. In *Proceedings of the Fifth Workshop on Generation, Evaluation and Metrics (GEM²)*, San Diego, USA. Association for Computational Linguistics.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024. Humans or LLMs as the judge? a study on judgement biases. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4634–4650. Association for Computational Linguistics.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2024. RAGAS: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–163. Association for Computational Linguistics.

[1] https://github.com/nlp-heds/repronlp2026

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488. Association for Computational Linguistics.

Yanjun Gao, Simeng Sun, and Steffen Eger. 2019. Automated pyramid summarization evaluation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 404–418. Association for Computational Linguistics.

Google DeepMind. 2024. Gemini 2.0 flash. Accessed via Google Gemini API.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606, Toronto, Canada. Association for Computational Linguistics.

Jimmy Lin and Dina Demner-Fushman. 2006a. Overview of the TREC 2006 question answering track. In *Proceedings of the Fifteenth Text REtrieval Conference (TREC 2006)*.

Jimmy Lin and Dina Demner-Fushman. 2006b. Will pyramids built of nuggets topple over? In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 383–390. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100. Association for Computational Linguistics.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152. Association for Computational Linguistics.

OpenAI. 2024. GPT-4o: Omni-modal flagship model. Accessed via OpenAI API.

Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, and Jimmy Lin. 2025. The great nugget recall: Automating fact extraction and RAG evaluation with large language models. *arXiv preprint arXiv:2504.15068*.

Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2024. ARES: An automated evaluation framework for retrieval-augmented generation systems. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 6605–6627. Association for Computational Linguistics.

Ori Shapira, David Davidov, and Ido Dagan. 2019. Crowdsourcing lightweight pyramids for manual summary evaluation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 682–687. Association for Computational Linguistics.

Anastasia Shimorina and Anya Belz. 2022. The human evaluation datasheet: A template for recording details of human evaluation experiments in NLP. In *Proceedings of the 2nd Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 54–75.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288. Association for Computational Linguistics.

Vectara. 2025. Open-RAG-Eval: RAG evaluation without the need for golden answers. `https://github.com/vectara/open-rag-eval`. GitHub repository.

Pat Verga, Sebastian Hofstatter, Fabio Petroni, Sebastian Riedel, and Patrick Lewis. 2024. Replacing judges with juries: Evaluating LLM generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*.

A Original vs. Reproduction Deviations

Table 5 provides a side-by-side comparison of the original study and our reproduction across two categories of settings. The upper rows capture intentional differences in experimental scope: we evaluate on QAMPARI (100 topics, list-QA) rather than the TREC 2024 RAG Track (301 topics, open-ended QA), build seven RAG systems rather than relying on community-submitted runs, and annotate 50 topics manually compared to the original's 56. These differences reflect the deliberate choice to test generalizability on a structurally different task domain, and are the primary focus of our analysis. The lower rows show that all core framework parameters—batch sizes, nugget cap, temperature, and processing order—were matched exactly to the original paper's specifications.

Aspect	Original	Reproduction
Dataset	TREC 2024 RAG (301 topics)	QAMPARI (100 topics)
Task type	Open-ended QA	List-QA
Systems	45 community runs	7 RAG systems
Manual topics	56 topics	50 topics
Stage 1 batch	10 segments/turn	10 segments/turn
Nugget cap	30 → top 20	30 → top 20
Stage 2/3 batch	10 nuggets/call	10 nuggets/call
Temperature	0.0	0.0

Table 5: Key differences between the original study and our reproduction. Framework parameters (bottom rows) match the paper's specifications.

B Annotation Details

All human annotations in this study—nugget creation, nugget editing, and nugget assignment—were performed by the paper authors. No external annotators were recruited, and no compensation was involved. Annotators studied the original AutoNuggetizer paper and its guidelines before beginning work. Calibration was done informally: annotators discussed boundary cases on the first few topics before proceeding independently. To ensure consistency, we prepared a dedicated instruction document for each annotation stage, specifying label definitions and decision rules; these documents are released in our reproduction repository at https://github.com/MoJa-Ardestani/ReproNLP-2026.

Annotation tasks. Annotation proceeded in three stages across 50 topics:

- **Nugget creation (MANUALNUGGETS):** Given a question and its gold reference segments, the annotator extracted atomic facts as nuggets and labeled each as *vital* or *okay*.
- **Nugget editing (AUTONUGGETS+EDITS):** Given a question, its gold segments, and the auto-generated nugget set, the annotator corrected the draft—adding missing nuggets, removing irrelevant ones, and revising poorly formed entries.
- **Assignment (MANUALASSIGN):** Given a question, its nugget set, and a system answer, the annotator assigned `support`, `partial_support`, or `not_support` to each nugget. This stage was applied to both the edited and manual nugget sets against all seven RAG system answers.

Scope. Manual annotation covered 50 topics out of the 100-topic evaluation set. These 50 topics were annotated completely across all stages. For the remaining 50 topics, only automatic nugget creation and assignment were used (reported as the 100-topic Auto condition in Table 2).

Effort and time. Annotation effort varied substantially across the three stages.

- **Nugget creation (50 topics):** Each topic required reading a question and its gold reference segments and composing nuggets from scratch. Estimated time: **10–15 minutes per topic** (∼8–12 person-hours total).
- **Nugget editing (50 topics):** Each topic required reviewing the question, gold segments, and the auto-generated draft nugget set, then applying corrections. Estimated time: **8–10 minutes per topic** (∼7–9 person-hours total).
- **Assignment (50 topics × 2 nugget sets × 7 systems):** Each topic required assigning support labels to every nugget in both the edited and manual nugget sets against all seven RAG system answers—yielding 700 (topic × nugget-type × system) annotation instances, each involving reading a system answer and labeling ∼7 nuggets. This was the most labor-intensive stage. Estimated time: **20–30 minutes per topic** (∼17–25 person-hours total).