Crucible @ Rag4Reports: Generating Nuggets for Report Generation and Evaluation

Laura Dietz
University of New Hampshire
USA
dietz@cs.unh.edu

Eugene Yang
Johns Hopkins University
USA
eugene.yang@jhu.edu

Abstract

We submit to both tracks of the RAG4Reports challenge with two complementary components: PREFNUGGET, which derives concise nugget banks from pairwise preference judgments between system responses, and CRUCIBLE, a nugget-first pipeline that uses such banks to assemble reports on a given topic. The shared nugget-level representation unifies our approach to report evaluation (Task A) and report generation (Task B).

1 Introduction

Our recent work explored the interplay between LLM-as-a-Judge and Retrieval-augmented Generation systems (RAG) (Dietz et al., 2026c).In particular, we explore different paradigms of automatically generated nuggets (Voorhees, 2003; Nenkova and Passonneau, 2004) in the form of open-ended questions to be used for two purposes:

1. using nuggets in the evaluation of reports via LLM-as-a-Judge,
2. using nuggets to drive the report generation inside a RAG system.

A nugget is a piece of information that should be contained in a good system response. There are different ways of phrasing nuggets, such as via concrete facts or claims that should be mentioned or questions that should be answered. We focus on question-nuggets as prior work found question-nuggets to be better at distinguishing better from best systems than fact nuggets (Farzi and Dietz, 2024a). Question nuggets can either be fact-oriented questions with "known good" gold answers or open-ended questions for which any reasonable answer would count.

The truth. In the experimental setup of Rag4Reports, the truth for both tasks is determined semi-automatically by AUTO-ARGUE (Walden et al., 2025). The evaluation uses manually identified questions with gold answers as nuggets, curated by NIST assessors. Which nuggets are successfully addressed is verified with a Yes/No prompt. Along with verification of faithful citations, this determines the "official" nugget coverage score. A prompt to verify whether a citation supports a sentence of the summary provides the basis for a sentence support score. AUTO-ARGUE's F1 measure is the harmonic mean between the nugget coverage and the sentence support score.

Dimension	Ours	AUTO-ARGUE
Nugget source	LLM-generated	manually curated
Nugget questions	open-ended	gold answers
Grading	0–5 scale	binary Yes–No
LLM	`gpt-oss-120b`	`llama-3.3-70b`
Grounding	preferences	cited snippets

Table 1: Comparison of our approaches to the AUTO-ARGUE "truth" evaluation pipeline.

For the report evaluation task (Task A), the official leaderboard of systems previously submitted to TREC 2025 is used as a ground truth.

For the report generation task (Task B), the manual evaluation artifacts collected when assessing the systems submitted to TREC 2025 are applied to evaluate submitted reports.

Our report evaluation system: PREFNUGGET. The LLM-as-a-judge system we submitted to Task A[1] also uses nuggets, but with critical differences as detailed in Table 1. The most differentiating factor is that nugget banks are automatically derived, then automatically checked against responses—as a result our system runs fully automatically, with no human intervention.[2]

[1]Code for PREFNUGGET evaluation: https://github.com/laura-dietz/prefnugget-starterkit/commits/rag4reports-submission

[2]Despite our submitted approach, the authors believe that evaluation systems must involve human assessors to be reliable (Dietz et al., 2025b).

Proceedings of the First Workshop on Multilingual Report Generation via Retrieval Augmented Generation (RAG4Reports), pages 77–82

```
You are a highly experienced and accurate assessor for TREC.
Select the passage that answers the query better. Just answer 1
or 2, without any explanation or extra verbiage.
If both passages are similar, answer with 0.
```

Field	Description
In: query	Query title, background, narrative
In: passage_1	Passage 1
In: passage_2	Passage 2
Out: better	1, 2, or 0 (tie)

(a) Phase 1: Preference judging. Prompt inspired by Arabzadeh and Clarke (2025); Yu et al. (2026).

```
Compare Winner vs Loser RAG responses for a query. Focus on
relevance, correctness, completeness. From given_exam_questions,
identify or generate questions the Winner addresses much better
than the Loser. Reuse questions where possible. New
differentiating_questions must be brief, atomic questions about
information the Winner handles much better. Avoid generic
quality questions. Make questions self-contained (e.g.,
"Capital of France?" not "The capital?").
```

Field	Description
In: query	Query title, background, narrative
In: winner_passage	Winning response
In: loser_passage	Losing response
In: given_exam_questions	Prior questions
Out: diff_questions	JSON array of new questions

(b) Phase 2: Iterative contrastive nugget extraction. Runs iteratively; previously extracted questions are provided to avoid redundancy. Stops after 20 unique questions.

```
Grade how well a passage answers a specific question. Can the
question be answered based on the available context? Choose one:
- 5: The answer is highly relevant, complete, and accurate.
- 4: The answer is mostly relevant and complete but may have
  minor gaps or inaccuracies.
- 3: The answer is partially relevant and complete, with
  noticeable gaps or inaccuracies.
- 2: The answer has limited relevance and completeness, with
  significant gaps or inaccuracies.
- 1: The answer is minimally relevant or complete, with
  substantial shortcomings.
- 0: The answer is not relevant or complete at all.
```

Field	Description
In: question	Nugget question
In: passage	Text to evaluate
Out: grade	Grade 0–5

(c) Phase 3: Grading prompt from RUBRIC (Farzi and Dietz, 2024a; Dietz, 2024).

```
Given a question on the given topic/background/problem
statement, find the sections in the provided source document
that support and validate the answer to the question. Provide
the supporting section with complete sentences directly from
the document, with enough surrounding context to justify why
the answer is correct. Respond with the extracted text segment
only. Then condense the extracted text into one concise
sentence that clearly demonstrates how the question is
answered, without referring to the source document.
```

Field	Description
In: nugget_text	Question
In: source_document	Source document
In: title_query	Query topic
In: background	Query background
In: problem_statement	Problem statement
Out: <extracted>	Supporting text from document
Out: <condensed>	One-sentence condensation

(d) Crucible Step 3: Report sentence generation.

Figure 1: LLM prompts and their input/output.

Details of the PREFNUGGET approach (Dietz et al., 2026a) are given below.

Our report generation system: CRUCIBLE. As the Retrieval-augmented Generation (RAG) system submitted to Task B,[3]

we build on our CRUCIBLE system, abbreviated "cru", (Dietz et al., 2026b), which uses an automatically generated nugget-bank to drive the generation of long-form responses.

While the CRUCIBLE RAG system originally used automatic nugget banks generated with DOGMATIQ (Li et al., 2026), for this submission we are using it with the nugget banks generated by PREFNUGGET (our Task A submission). We also submit one variant, called "rubric", that uses a generated nugget bank from LLM's internal knowledge without any grounding.

We do not perform an internal validation to selection submissions, but draw on experience with related datasets: TREC NeuCLIR 2024(Lawrie et al., 2025a) for CRUCIBLE and TREC AutoJudge Pilot Dataset v0.2 (Dietz et al., 2025a) for PREFNUGGET. We do not perform internal validation to select submissions, but draw on experience with related datasets: TREC NeuCLIR 2024 (Lawrie et al., 2025b) for CRUCIBLE and TREC AutoJudge Pilot Dataset v0.2 (Dietz et al., 2025a) for PREFNUGGET.

2 Approach Task A: Automatic Report Evaluation

The goal is to derive compact, discriminative nugget banks from pairwise preference signals and the query. Winner-loser comparisons reveal which information distinguishes strong responses from weak ones; questions targeting these differences naturally capture evaluation-relevant criteria.

2.1 Phase 1: Pairwise Preferences

For each topic, we collect pairwise preference judgments over system responses that are to be evaluated. Each response is compared against four others, with pairs selected via stratified sampling (avoiding periodicity). An LLM judge determines which response better answers the query, resulting in winner-loser pairs. We follow the preference judgment paradigm of Arabzadeh and Clarke (2025), but for this submission, we permitted the

[3]Code for CRUCIBLE report generation: https://github.com/laura-dietz/scale25-crucible/commit/rag4reports-submission

system to indicate ties, based on findings from Yu et al. (2026). After the submission we realized that permitting ties actually reduces the performance slightly.

2.2 Phase 2: PREFNUGGET Banks

For each winner-loser pair where response A is better than response B, we use the prompt in Figure 1b to extract at most two factual questions that capture how response A is better. To avoid redundant nuggets (which are costly to reduce later (Li et al., 2024)), we provide previously extracted questions as candidates for selection, and ask the LLM to generate novel questions only when necessary. Extraction terminates for a topic when either the maximum budget of unique questions has been reached (here: 20) or a maximum number of pairs (here: 100) have been processed.

2.3 Phase 3: Response Grading

The final phase evaluates all RAG responses against the derived nugget bank. Each nugget question is scored on a six-point scale (0–5) indicating how well the provided RAG response answers it. When grading response texts, the nugget-grade is obtained by scoring the nugget against the response text, yielding one grade per nugget.

We predict an evaluation score for each response by one of these measures:

maxgrade: The nugget for which the RAG response obtained the highest grade is used as an evaluation score. This approach was proposed in Farzi and Dietz (2024b).
avggrade: The average grade the response obtained across all nuggets.
cover4: The fraction of nuggets for which the response obtained a grade of 4 or 5.

3 Approach Task B: Multilingual Report Generation

CRUCIBLE is a RAG pipeline which starts by generating question-nuggets and uses them to guide retrieval, extraction, and assembly as follows.

Step 1: Nugget ideation. We begin by generating a bank of open-ended question nuggets. Here these are obtained from PREFNUGGET or generated from the LLM's internal knowledge (`rubric`).

Step 2: Retrieval. While nuggets can be used to retrieve relevant documents (as in CRUXX, Ju et al. (2025)), here we use a one-shot retrieval stage using the multi-lingual retrieval model MILCO (Nguyen et al., 2025) based on a search query that concatenates problem statement, title, and background.

Step 3: Scanning and generation. Using the nugget bank from Step 1, we scan retrieved documents[4] for passages that directly answer each nugget. Using the prompt in Figure 1d, we (1) locate a supporting passage, (2) generate a concise self-contained sentence, and (3) record the LLM's token-likelihood as the extraction confidence.

The original paper makes reference to a verification step which is prone to creating a circularity with AUTO-ARGUE, which we did not use here.

Step 4: Sentence selection. We rank the remaining candidates for each nugget by the extraction confidence and choose the top k sentences, with $k = 1$ for short reports and $k = 5$ for long reports. This ensures that each sentence is tied to exactly one citation.

Step 5: Assembly. Selected sentences are concatenated into a report. Repeated sentences (same stopped/stemmed text) are omitted. Because every sentence is self-contained and atomic, the order of sentences does not affect the readability. Every sentence cites exactly one document.

3.1 Variations

We submit three variations of CRUCIBLE to Rag4Reports.

cru-prefnu: Uses CRUCIBLE on a nugget bank of 20 open-ended questions from PREFNUGGET (as submitted to Task A).
cru-prefnu-extract: As previous, but instead of using abstractive summarization, the supporting passage is used as-is.
cru-rubric: Uses CRUCIBLE on a nugget bank of 20 open-ended questions that is generated only from the LLM's internal knowledge without any form of grounding. (The approach is similar to the RUBRIC as introduced by Farzi and Dietz (2024b), albeit with an adjusted prompt and a more recent LLM.)

We remark that while GINGER (Lajewska and Balog, 2025) also uses nuggets, they define nuggets to be clusters of sentences extracted from source documents. In contrast, CRUCIBLE operates by

[4] Segmented into 1000 character chunks, split at sentence boundary.

Team / Run	τ_{gap}	Kendall	Topic
coordinators / autoargue-f1	*0.470*	*0.636*	*0.607*
unh / citation accuracy	0.414	0.559	0.168
unh / retrieval quality	0.226	0.469	0.415
unh / final	0.221	0.430	0.386
crucible / maxgrade	**0.177**	**0.390**	**0.356**
unh / attribution	0.173	0.291	0.306
crucible / cover4	**0.128**	**0.360**	**0.190**
tiet / f1 (weighted)	0.127	0.334	0.573
unh / negation-judge	0.109	0.284	0.260
crucible / avggrade	**0.107**	**0.355**	**0.201**
ju-nlp-ug / rank-argue-v4	0.099	0.375	0.546
unh / checklist	0.096	0.300	0.343
rgipt / Qwen2-NRB	0.077	0.133	0.221
tiet / nugget coverage	0.075	0.289	0.599
ju-nlp-pg / cite-first	0.026	0.123	0.161
tiet / sentence support	−0.056	0.183	0.247
unh / naive-length	−0.084	0.222	0.229
rgipt / qwen0-5	−0.125	0.154	0.251
rgipt / semiauto-run1	−0.125	0.154	0.251
rgipt / Qwen2-0.5B-NRB	−0.269	−0.297	0.185
unh / naive-random	−0.277	−0.143	0.010

Table 2: Evaluation metrics by team and run, sorted by τ_{gap}. Our team's entries shown in bold.

Team / Run	Sent.	Nugget	F1
crucible / cru-prefnu-extract	**0.965**	**0.470**	**0.586**
genaius / cluster-gpt4	0.814	0.443	0.546
genaius / trial	0.794	0.439	0.538
crucible / cru-rubric	**0.783**	**0.447**	**0.522**
genaius / cluster-gpt5	0.662	0.449	0.501
crucible / cru-prefnu	**0.684**	**0.425**	**0.480**
amu / bge	0.828	0.340	0.435
amu / qwen4b	0.833	0.287	0.383
amu / qwen8b	0.789	0.276	0.350
ju-nlp-pg / ver3-taskb	0.214	0.378	0.231
sgd / subtask2-local-bm25	0.934	0.149	0.228
sgd / subtask2-test-2	0.934	0.149	0.228
ug-tiet-nlp / efsg-t5	0.612	0.126	0.182
ug-tiet-nlp / efsg-t4	0.327	0.065	0.097
ju-nlp-pg / ver1-submission	0.036	0.469	0.046
ju-nlp-pg / ver2-taskb	0.036	0.469	0.046

Table 3: Sentence support, nugget coverage, and F1 by team and run, sorted by F1. Our team marked in bold.

first identifying question-nuggets, then extracting sentences from source documents.

4 Results

All results are generated with `gpt-oss-120b` as the underlying LLM. Results are provided by Rag4Report coordinators via the TIRA leaderboard.[5] Overall, we find that our approaches are working reasonably well. Using PREFNUGGET as Auto-Judge (Task A), we find that the maxgrade approach demonstrates medium-well agreement with manual "truth" leaderboards in Kendall's tau on the F1 measure of 0.39, and τ_{GAP} of 0.177. This is a good result, given that the truth measures emphasize citation accuracy verification, which is not captured in our auto judge approach.

We find that other variants of PREFNUGGET work less well than maxgrade, which is a finding that is in line with other experimental studies (Dietz et al., 2026a; Farzi and Dietz, 2024a).

When PREFNUGGET is used along with our RAG system CRUCIBLE to generate long-form reports (Task B), we obtain the best results with the extractive summarization approach `cru-prefnu-extract`. As sentences are taken verbatim from the cited document, it obtains a near-perfect sentence support score of 0.965. With 0.470 the nugget coverage is slightly above the abstractive approach `cru-prefnu` (0.425) even though the same underlying nugget bank is used to drive the generation. The explanation is that the generated nuggets are not a perfect match with the manually curated nuggets that form the ground truth, but that they match sentences that include additional relevant information. In contrast, the abstractive generation is instructed to only focus on representing the contained answer to the nugget question. So the strength of this approach lies in detecting good sentences with our nugget bank, which also includes other information that was found to be more essential by manual assessors.

We note that among the two abstractive methods, the RUBRIC-style nugget bank, which does not use any form of grounding, performs better than the PREFNUGGET approach. This implies that grounding can miss relevant information, especially when using a strong modern LLM like `gpt-oss-120b`. In order to improve upon strong LLMs, the grounding would need to include knowledge that is not already known by the LLM.

5 Conclusion

We demonstrate that nugget-bank generation methods can yield both high-quality LLM-as-a-Judge approaches as well as retrieval-augmented generation systems. Our results show that modern LLMs can successfully generate nugget banks without grounding, and extractive summarization approaches are still competitive.

[5] archive.tira.io/task-overview/rag4reports/task-a-report-evaluation-20260403-training

Limitations

Our submission is limited by the circularity risk inherent in using LLM-generated nuggets to evaluate or guide LLM-generated reports (Dietz et al., 2026c). This risk is particularly relevant when the same model family, prompts, or intermediate representations are used across generation and evaluation. While our methods are fully automatic and therefore easy to reproduce, they should not be understood as replacing human assessment for reliable evaluation.

Only three submissions were accepted by the challenge. In hindsight, adding a rubric-based experiment to the Task A submission would have led to more conclusive findings on whether grounding is necessary. Our current results therefore compare preference-grounded nuggets against rubric-style nuggets only in Task B, not in Task A.

Ethical considerations

Datasets used to develop and test methods originated from a public competition at TREC 2025. Participating teams submitted methods; as their identities are immaterial, their run names are anonymized (both in Rag4Reports and in TREC AutoJudge). Manual annotations were created by NIST under rules of research with human subjects.

Because of the circularity risk, bad actors can misuse these findings to obtain inflated evaluation results in other data challenges, especially when insider knowledge of the evaluation approach is available. This concern is not specific to our system, but applies broadly to automatic evaluation settings where participants can infer or approximate the evaluation procedure. We therefore recommend that benchmark organizers keep human oversight, use multiple independent evaluation signals, and avoid relying on a single automatically generated nugget bank as the sole source of truth.

References

Negar Arabzadeh and Charles L. A. Clarke. 2025. A human-ai comparative analysis of prompt sensitivity in llm-based relevance judgment. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25)*. ACM.

Laura Dietz. 2024. A workbench for autograding retrieve/generate systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1963–1972.

Laura Dietz, Naghmeh Farzi, Eugene Yang, and Dawn Lawrie. 2026a. Too many questions: Deriving concise and effective nugget banks. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26)*, SIGIR '26, Melbourne, VIC, Australia. ACM.

Laura Dietz, Bryan Li, Gabrielle Liu, Jia-Huei Ju, Eugene Yang, Dawn Lawrie, William Walden, and James Mayfield. 2026b. Incorporating Q&A nuggets into retrieval-augmented generation. In *Proceedings of the 48th European Conference on Information Retrieval (ECIR 2026)*.

Laura Dietz, Bryan Li, Eugene Yang, Dawn Lawrie, William Walden, and James Mayfield. 2026c. Insider knowledge: How much can rag systems gain from evaluation secrets? In *Proceedings of the 48th European Conference on Information Retrieval (ECIR 2026)*.

Laura Dietz, Ian Soboroff, and Coordinators of various TREC tracks. 2025a. Trec autojudge pilot data. Licensed under CC BY-SA 3.0. Based on content from TREC DRAGUN, TREC RAG, TREC RAGTIME, their license applies.

Laura Dietz, Oleg Zendel, Peter Bailey, Charles LA Clarke, Ellese Cotterill, Jeff Dalton, Faegheh Hasibi, Mark Sanderson, and Nick Craswell. 2025b. Principles and guidelines for the use of llm judges. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*, pages 218–229.

Naghmeh Farzi and Laura Dietz. 2024a. Exam++: Llm-based answerability metrics for ir evaluation. In *Proceedings of LLM4Eval: The First Workshop on Large Language Models for Evaluation in Information Retrieval*.

Naghmeh Farzi and Laura Dietz. 2024b. Pencils down! automatic rubric-based evaluation of retrieve/generate systems. In *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 175–184.

Jia-Huei Ju, Suzan Verberne, Maarten de Rijke, and Andrew Yates. 2025. Controlled retrieval-augmented context evaluation for long-form rag. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 21102–21121.

Weronika Lajewska and Krisztian Balog. 2025. Ginger: Grounded information nugget-based generation of responses. In *Proceedings of the 48th International ACM SIGIR Conference (SIGIR '25)*. SIGIR 2025 paper.

Dawn Lawrie, Sean MacAvaney, James Mayfield, Paul McNamee, Douglas W Oard, Luca Soldaini, and Eugene Yang. 2025a. Overview of the TREC 2024 NeuCLIR track. In *Proceedings of TREC*.

Dawn Lawrie, Sean MacAvaney, James Mayfield, Paul McNamee, Douglas W Oard, Luca Soldaini, and Eugene Yang. 2025b. Overview of the TREC 2024 neuclir track.

Bryan Li, William Walden, Yu Hou, Gabrielle Kaili-May Liu, Dawn Lawrie, Jame Mayfield, Eugene Yang, Chris Callison-Burch, and Laura Dietz. 2026. Dogmatiq: Automated generation of question-and-answer nuggets for report evaluation. *Preprint*, arXiv:2605.04458.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*.

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, Boston, Massachusetts. Association for Computational Linguistics.

Thong Nguyen, Yibin Lei, Jia-Huei Ju, Eugene Yang, and Andrew Yates. 2025. Milco: Learned sparse retrieval across languages via a multilingual connector. *arXiv [cs.IR]*.

Ellen M. Voorhees. 2003. Overview of the TREC 2003 question answering track. In *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003)*, Gaithersburg, Maryland. NIST.

William Walden, Orion Weller, Laura Dietz, Bryan Li, Gabrielle Kaili-May Liu, Yu Hou, and Eugene Yang. 2025. Auto-ARGUE: LLM-based report generation evaluation. *arXiv preprint arXiv:2509.26184*.

Chuting Yu, Hang Li, Guido Zuccon, Joel Mackenzie, and Teerapong Leelanupab. 2026. When llm judges inflate scores: Exploring overrating in relevance assessment. *arXiv preprint arXiv:2602.17170*.