An Iterative Preference Learning Framework for Retrieval-Augmented Generation

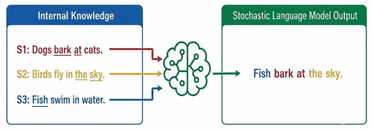
Aaryan Patel
Phillips Exeter Academy • apatel@exeter.edu

1 LLM Hallucination Mitigation via RAG

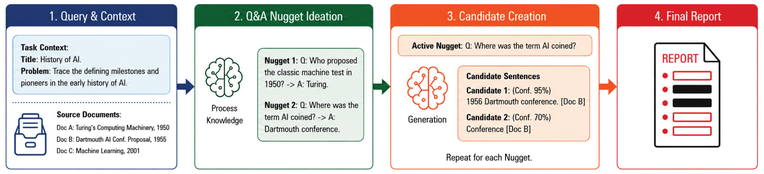
- **Large Language Models** (LLMs) frequently hallucinate and spread factual inaccuracies.
- **Root Cause:** LLMs primarily use next-token prediction instead of querying knowledge bases, making them prone to hallucination.
- The RAG Framework: **Retrieval-Augmented Generation** mitigates this by creating language-model reports grounded in citable evidence from verified knowledge bases.
- **Research Objective:** We investigate whether a RAG system can iteratively "learn" and improve its outputs through a self-feedback loop.

2 Generation & Evaluation using Nuggets

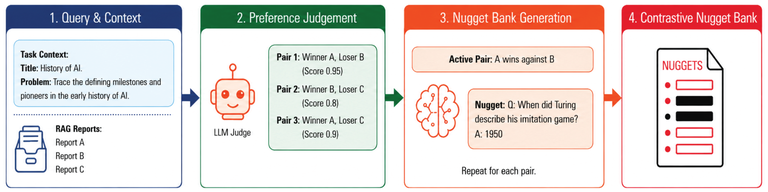
- **Stochastic Language Models**, the most basic form of LLMs, rely on probabilistic token prediction, leaving them highly prone to conflating facts and hallucinating false outputs.



- **CRUCIBLE** is a RAG pipeline that constructs a bank of Q&A nuggets, specifying the information the final report should contain.
- **Nuggets** are atomic facts or questions representing a specific piece of information that an ideal RAG report would include based on the user's prompt.
- Each nugget guides the system to find supporting evidence in the retrieved documents and keep track of citations.



- **PrefNugget** compares a winner and loser response to construct a discriminative nugget bank of atomic questions regarding information the winning report addresses better.

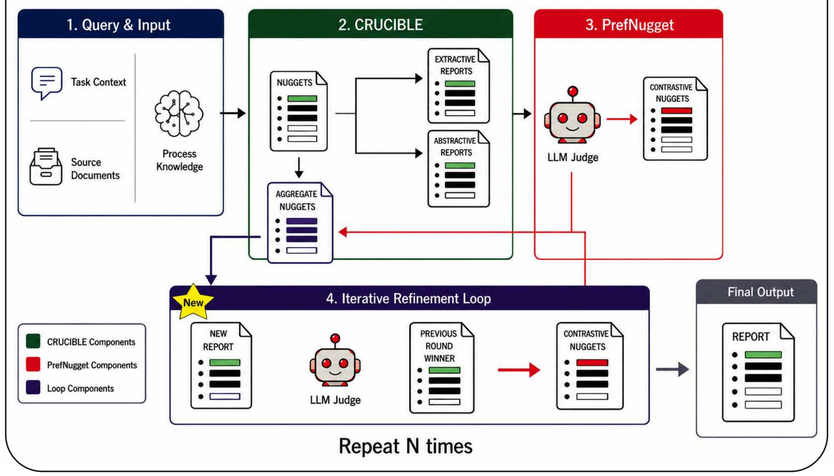


3 Limits of Static Retrieval

- **Limitation of Standard RAG Pipelines:** Initial outputs can miss key information, and single-pass systems lack the ability to identify and correct those gaps.
- **Actionable Feedback over Numerical Scores:** Pairwise preference judgments offer actionable, attribute-specific feedback instead of uninformative scalar scores.
- **Hypothesis:** Iteratively feeding preference-derived information nuggets back into the RAG pipeline guides subsequent iterations to fill missing information gaps.

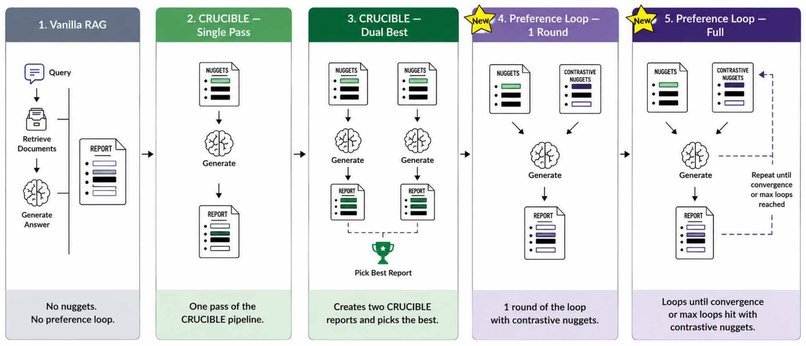
4 Contrastive Nugget Feedback Loops

- We utilize the CRUCIBLE and PrefNugget frameworks to create a feedback loop between generation and evaluation.
- **Step 1 CRUCIBLE Extraction:** A CRUCIBLE-inspired system is used to process the user's task context, extract initial nuggets, and generate an extractive and abstractive report.
- **Step 2 PrefNugget Judgement:** A PrefNugget-inspired system uses an LLM judge to compare the two outputs and produce a contrastive nugget bank.
- **Step 3 Iterative Refinement:** This pipeline repeats, iteratively adding the new contrastive nugget bank to the existing one and generating a new report. This report is then compared against the previous round's winner.
- **Termination:** The loop repeats a predetermined number of times or stops when the contrastive nuggets converge and no new information is generated.



5 Benchmarking on TREC RAGTIME Corpus

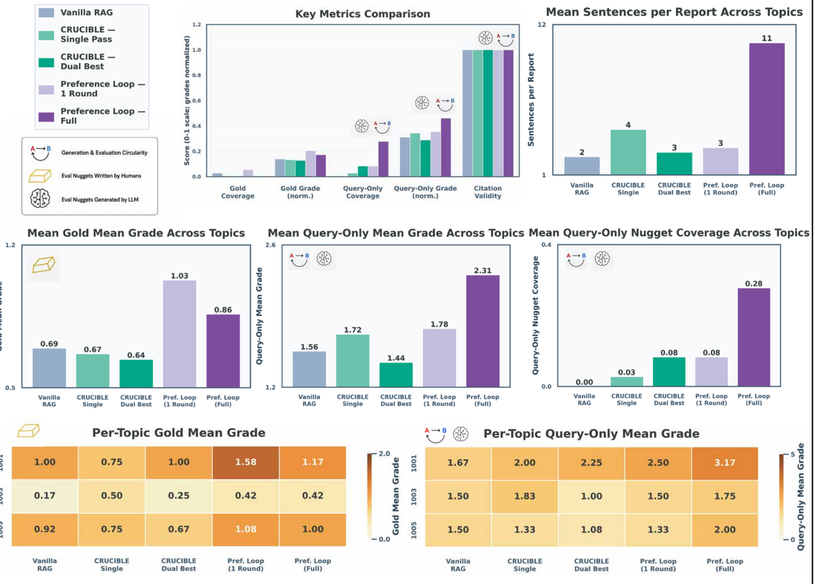
- **Dataset:** The RAG reports were generated using The Text Retrieval Conference RAGTIME dataset which offers numerous multilingual documents and corresponding topics.
- **Control:** Each RAG report was generated using the same query, context, and knowledge base to ensure fair comparison.
- **Baselines:** 5 total RAG reports were generated for each topic.



- **Evaluation:** Each report was scored on 5 metrics: Gold Grade & Coverage (human written nuggets), Query-Only Grade & Coverage (LLM generated nuggets), and Citation Validity.

6 Comparative Performance across Pipelines

- Results were attained using three topics from the dataset.



- **Baseline Outperformance:** Preference loop pipelines beat 80% of baseline systems.
- **Performance Ceiling:** Gold coverage is a shared ceiling across pipelines, likely due to the retrieval method, LLM, and other controlled variables.
- **Report Length:** Multiple loops significantly increased report size.
- Multi-loop preference was favored on query-only metrics while single-loop preference was favored on Gold metrics.

7 System Limitations & Future Work

- **Model & Evaluation Scope:** Evaluation was only done on 3 topics. Low-level LLMs and retrieval methods were used, likely leading to the low gold coverage results and overall low performance across all pipelines.
- **Bias:** The LLM used for evaluation was also used for pairwise comparison and contrastive nugget generation. This bias could explain stronger performance on the query-only metrics.
- **Termination**: The current pipeline runs for the maximum number of iterations (n=4). Smarter stopping mechanisms will mitigate overly long reports and generate stronger overall reports.

The closed loop optimizes LLM-assessed report quality, yet gold-nugget performance remains unchanged; bridging judge-aligned and gold-aligned improvement remains an open work.

8 References & Acknowledgments

Research Advisor: Prof. Laura Dietz (Department of Computer Science, University of New Hampshire)

L. Dietz et al. Incorporating Q&A Nuggets into Retrieval-Augmented Generation. ECIR 2026.
L. Dietz et al. Crucible @ Rag4Reports: Generating Nuggets for Report Generation and Evaluation. TREC RAGTIME / Rag4Reports, 2025.
L. Dietz et al. Too Many Questions: Deriving Concise and Effective Nugget Banks. SIGIR 2026.