Final Summary Report: Naive and Enhanced RAG Systems

Executive Summary

This project developed and evaluated a Retrieval-Augmented Generation (RAG) system built on a naive requery baseline and an enhanced variant. The naive system directly queried a vector database using 384-dimensional embeddings, returning the top-k results without additional refinement. The enhanced system extended this design with iterative requery, instructional prompting, and structured retrieval adjustments.

Across 918 test queries, quantitative evaluation showed mixed outcomes. The naive system achieved **34.64% Exact Match (EM)** and **42.13% F1**, while the enhanced version slightly underperformed with **28.98% EM** and **36.41% F1**. However, ARES evaluation revealed a different story. Enhanced RAG improved Answer Relevance **(34.39% vs 33.93%)** and Answer Faithfulness **(34.95% vs 34.60%)**, while sacrificing Context Recall **(23.36% vs 26.92%)** and Context Precision **(21.09% vs 21.52%)**.

These results underscore a precision-recall trade-off. Naive requery surfaces more context, improving overlap-based metrics, while enhanced requery strategies yield more grounded, faithful answers at the cost of breadth. The system is not yet deployment-ready for high-accuracy use cases, but findings highlight critical lessons for balancing retrieval scope, prompting discipline, and evaluation methodology.

System Architecture

The naive system architecture centered on a straightforward requery workflow:

- **Query Embedding:** Input questions were embedded with Sentence-Transformers (all-MiniLM-L6-v2) into 384-dimensional vectors.
- **Requery Retrieval:** Each query was issued directly to Milvus Lite, returning the top-k nearest passages based on cosine similarity.
- **Answer Generation:** Retrieved passages were concatenated with the user query and passed to the OpenAI LLM (gpt-3.5-turbo) for synthesis.

This design offered simplicity and consistent latency. However, it lacked mechanisms to filter noise or rerank passages, leaving the LLM to infer missing details.

The enhanced system introduced three extensions:

- **Iterative Requerying:** Queries were reformulated with synonyms and paraphrases to improve recall, producing multiple candidate retrievals.
- **Instructional Prompting:** LLM prompts constrained generation to “only use provided context,” discouraging hallucinations.
- **Lightweight Metadata Integration:** Where available, metadata-guided requerying toward higher-quality candidates.

This architecture aimed to increase faithfulness and relevance but carried trade-offs: iterative requery added latency, instructional prompts reduced answer fluency in some cases, and stricter retrieval often narrowed the available evidence pool.

The design demonstrates a conscious balance: the naive pipeline is lean and reproducible, while the enhanced version prioritizes factual reliability.

Experimental Results

Naive System Performance

The naive requery system achieved:

- **Exact Match (EM): 34.64%**
- **F1 Score: 42.13%**
- **ARES Context Relevance: 66.75%**
- **ARES Answer Relevance: 33.93%**
- **ARES Answer Faithfulness: 34.60%**
- **ARES Context Precision: 21.52%**
- **ARES Context Recall: 26.92%**

These results show that the naive system provided reasonable lexical overlap with ground truth answers. However, error analysis revealed frequent hallucinations when retrieved documents were off-topic, as well as instability across similar queries due to the lack of query reformulation.

Enhanced System Performance

The enhanced system achieved:

- **Exact Match (EM): 28.98%**
- **F1 Score: 36.41%**
- **ARES Context Relevance: 66.26%**
- **ARES Answer Relevance: 34.39%**
- **ARES Answer Faithfulness: 34.95%**
- **ARES Context Precision: 21.09%**
- **ARES Context Recall: 23.36%**

Here, surface metrics (EM/F1) decreased, reflecting fewer overlapping tokens with ground truth. However, ARES evaluation demonstrated slight gains in answer helpfulness (+1.4%) and faithfulness (+0.99%). These improvements confirm that enhanced requery made the model less prone to hallucination, albeit at the cost of discarding potentially useful context.

Interpretation

The data reflects a classic trade-off:

- Naive requery creates broader recall, better EM/F1, but weaker grounding.
- Enhanced requery creates narrower recall, lower lexical match, but slightly more faithful and relevant answers.

This illustrates why multi-dimensional evaluation is essential: surface overlap metrics alone would undervalue the improvements in groundedness.

Enhancement Analysis

The three main enhancements were:

- **Iterative Requery Expansion:** Helped diversify retrieval, but reduced stability when reformulations yielded divergent passages → Improved faithfulness slightly (+0.35) but hurt recall (–3.6).

- **Instructional Prompting:** Directly targeted hallucinations. Faithfulness gains (~+0.99) stemmed largely from clearer constraints on model behavior → sometimes over-constrained the LLM, leading to terse “insufficient context” answers.
- **Metadata-Guided Requery:** Provided precision improvements in certain cases (e.g., topical focus), but reduced recall overall → Results showed negligible precision gains (+ 0.2%) but measurable recall losses (- 3.6%).

Overall, these enhancements did not boost token-overlap scores but slightly increased groundedness and helpfulness, aligning with the project’s goal of reducing hallucinations. The reduced recall and lexical match highlight that improvements in reliability often come at the expense of coverage.

Production Considerations

For deployment, several key factors emerge:

- **Scalability:** Naive requery is computationally efficient but too unreliable for production. Enhanced requery increases retrieval costs and complexity, requiring caching and batching strategies.
- **Accuracy vs Latency:** Enhanced methods improve reliability slightly but reduce recall, and multiple requery calls raise response times by ~25 - 30%.

Deployment Recommendations:

- Use hybrid retrieval (dense + sparse/BM25) to balance recall and precision.
- Cache requery expansions to minimize latency.
- Integrate reranking selectively for high-stakes queries.
- **Limitations:** The Current system is tuned to a small dataset; open-domain performance remains uncertain. Metadata availability is uneven, reducing robustness.

In short, the enhanced pipeline moves closer to production readiness in restricted domains, but trade-offs in latency, recall, and complexity mean broader deployment will require hybrid approaches and continuous monitoring.

Appendices

- Technical Specs: Sentence-Transformers (all-MiniLM-L6-v2, 384-dim embeddings), Milvus Lite vector DB, OpenAI GPT-3.5-Turbo.
- Reproducibility: Scripts include generating embeddings and the rag pipeline.