Ground the Domain: From Naïve RAG to Production Patterns

Phase 7: Final Report

Chongdi Wang

I Executive Summary

In this project, we explored the design, evaluation, and enhancement of a Retrieval-Augmented Generation (RAG) pipeline, beginning with a naive baseline and progressing to advanced feature integration. The naïve system, built on chunked Wikipedia passages, Milvus indexing, and a Flan-T5 generator, established baseline performance using exact match (EM) and F1. Results confirmed basic functionality but exposed gaps in both precision and recall.

Subsequently, we conducted systematic experiments which revealed that retrieval depth (top-k) and reranking strategies heavily influenced performance. Concatenation at k=5 maximized F1, while maximal marginal relevance (MMR) reduced accuracy. These findings motivated us to add two advanced features to the RAG pipeline: query rewriting to handle ambiguous queries and reranking to diversify evidence.

Evaluation with RAGAS metrics provided retrieval-specific insights. The naive pipeline achieved higher precision, whereas the advanced system improved recall but at the cost of precision. These complementary results illustrate a trade-off between factual coverage and answer purity.

Overall, the project demonstrates that RAG pipelines are deployment-ready for applications prioritizing recall or precision, but further hybridization is recommended for balanced performance. With proper chunking, indexing, and evaluation design, the system offers a strong foundation for scalable production deployments and academic benchmarking.

II System Architecture

The RAG system comprised three main stages: data preprocessing, retrieval, and generation.

Preprocessing and Chunking. Exploratory data analysis of the *rag-mini-wikipedia* dataset showed passage lengths averaging ~390 characters (~81 tokens). To align with embedding model limits while retaining semantic completeness, passages were chunked into 600-character windows. This design balanced retrieval efficiency against context richness.

Vector Indexing. In the construction of the naïve RAG pipeline, Milvus served as the primary vector database, using all-MiniLM-L6-v2 embeddings (384 dimensions).

Hierarchical Navigable Small World (HNSW) indexing was tested but replaced by FLAT indexing for reproducibility. In the advanced evaluation phase, as we lacked access to the most recent version of RAGAS, FAISS acted as a fallback backend for simplified local evaluation, confirming pipeline modularity.

Retrieval. For naive retrieval, queries were embedded and matched against the vector store using inner product similarity. The top-k passages were selected and concatenated into prompts. This simple concatenation proved reliable but also introduced redundancy.

Generation. The chosen generator was Flan-T5-base, a seq2seq model with 512-token input constraints. Generated answers often reflected the quality of retrieved evidence, making retrieval precision and recall critical.

Design Trade-offs. Milvus offered scalability but introduced infrastructure complexity; FAISS was lightweight but lacked distributed deployment features. Larger chunk sizes improved coverage but risked truncation during generation. Retrieval depth increased recall but diluted relevance. These trade-offs framed the rationale for enhancements such as query rewriting and reranking, aiming to mitigate weaknesses in the naive setup.

III Experimental Results

The system was evaluated in two phases: early EM/F1 testing and RAGAS-based retrieval evaluation.

Phase 1: EM/F1 Benchmarks. On 20 sampled questions, the naive pipeline achieved:

- Instruction prompt: EM = 35.0, F1 = 44.6
- Persona prompt: EM = 35.0, F1 = 44.6
- Chain-of-Thought prompt: EM = 0.0, F1 = 8.6

These results confirmed baseline functionality but highlighted prompt sensitivity. Chain-of-Thought underperformed due to excessive complexity relative to dataset size. On the other hand, persona prompt consistently yields the highest performance across metrics, and is thus selected as the baseline for all experiments in the next phases.

Phase 2: Parameter Sensitivity. Experiments varied top-k (3, 5, 10) and compared reranking strategies (concatenation vs. MMR).

- For MiniLM6v2, concatenation at k=5 yielded the best F1 (53.3). Increasing to k=10 reduced F1 (50.1) due to noise.

- MMR consistently underperformed: MiniLM6v2 at k=5 with MMR dropped to F1 = 37.0.
- Mpnetv2 embeddings were more stable but trailed MiniLM in peak performance.

Statistical Support. These trends confirmed the bias–variance trade-off: higher k improved recall but reduced precision, while reranking promoted diversity at the expense of accuracy.

Phase 3: RAGAS Evaluation. To isolate retrieval quality, RAGAS was applied on 50 samples for each system—100 samples in total. Due to OpenAI key restrictions, ID-based context metrics were used.

The results across two pipelines are displayed in the *./results/step6_comparison_analysis_ragas.csv* table in the GitHub project repository.

The naive system excelled in precision, reflecting its selective retrieval. The advanced system substantially improved recall, capturing more relevant documents but at the expense of irrelevant inclusions. This confirmed that query rewriting and reranking enhanced coverage but weakened selectivity.

Interpretation. For fact-critical applications (e.g., compliance, legal), naive retrieval may be preferable. For exploratory tasks (e.g., research support), advanced retrieval ensures broader coverage.

IV Enhancement Analysis

Two enhancements were introduced in Step 5: query rewriting and reranking.

a) Query Rewriting.

Motivation: Many failures stemmed from ambiguous queries not matching document phrasing. A rewriting module rephrased user input into semantically equivalent but context-aligned alternatives. For example, “Lincoln president” could be expanded into “Was Abraham Lincoln the sixteenth President of the United States?”.
Effectiveness: Rewriting improved recall, as evidenced in the advanced pipeline’s RAGAS recall score. However, excessive expansion occasionally retrieved tangential passages, diluting answer quality.

b) Reranking.

Motivation: To address redundancy in top-k retrieval, MMR was reintroduced in the advanced pipeline. While prior experiments showed MMR alone underperformed, combining

it with rewritten queries allowed for coverage of distinct facets of the corpus.
Effectiveness: Reranking prevented near-duplicate passages from dominating retrieval. Still, RAGAS precision dropped sharply, demonstrating that diversification without strong filters risks overwhelming the generator with noise.

Several challenges have occurred during the evaluation of the advanced pipeline:

- Token Length Constraints: Flan-T5's 512-token limit required truncation logic, complicating evaluation.
- API Limitations: RAGAS LLM-based metrics were unavailable without OpenAI keys, forcing fallback to ID-based measures.
- Stability: Query rewriting and reranking modules introduced runtime overhead and additional hyperparameter tuning.

In summary, enhancements produced measurable gains in recall but did not yield an overall performance uplift across all metrics. They demonstrated proof-of-concept value but require hybridization—such as confidence-weighted reranking or selective rewriting—to achieve balanced improvements.

V Production Considerations and Limitations

Deploying this RAG pipeline in production raises several considerations:

a) Scalability.

- *Indexing*: Milvus offers horizontal scalability for large corpora, while FAISS provides lightweight local deployment. A hybrid approach could balance scalability and simplicity.
- *Query Load*: Advanced features increase query latency due to rewriting and reranking. Production systems may need caching, batched retrieval, or approximate nearest neighbor optimizations.

b) Deployment Readiness.

- *Naive Pipeline*: With higher precision and predictable runtime, the naive system is better suited for environments requiring consistent, high-confidence answers.
- *Advanced Pipeline*: With higher recall, the advanced system suits exploratory use cases but requires safeguards against irrelevant retrieval.

Limitations.

- *Evaluation Gaps*: Lack of LLM-based RAGAS metrics limited fine-grained assessment of answer faithfulness.
- *Token Limits*: The 512-token ceiling of Flan-T5 constrains scaling to larger contexts.
- *Trade-offs*: Precision–recall imbalance highlights the need for adaptive systems that tune retrieval depth dynamically.

VI Final Recommendations

For deployment, a dual-pipeline strategy is advised: defaulting to naive retrieval for precision-sensitive domains and enabling advanced retrieval for recall-sensitive tasks. Incorporating monitoring (e.g., precision–recall drift detection) and progressive enhancement (e.g., hybrid reranking, local evaluators) will ensure robustness as workloads scale.