# Technical Report: RAG System Development

## 1. Executive Summary

This project involved developing and evaluating a production-ready RAG system through systematic experimentation and enhancement. Starting from a naive baseline (F1=31.38), strategic improvements were implemented such as including optimized prompting strategies, embedding model selection, and advanced retrieval techniques. The final enhanced system integrates query rewriting and cross-encoder reranking, achieving 56.9% improvement in faithfulness and 75% improvement in context recall over the naive baseline. RAGAs evaluation on 100 questions confirmed substantial gains across all metrics: faithfulness (0.43→0.67), answer relevancy (0.19→0.27), context precision (0.38→0.60), and context recall (0.36→0.63). The system demonstrates production readiness for knowledge-intensive question answering, with identified scalability considerations for deployment at scale.

## 2. System Architecture

### 2.1 Naive RAG Pipeline

The baseline system processes 1,000 passages from the RAG Mini Wikipedia dataset, splitting them into 1,289 chunks of 600 characters each. Both documents and user queries are converted into numerical representations using the all-MiniLM-L6-v2 embedding model (384 dimensions) and stored in a Milvus Lite database with cosine similarity indexing. When a user asks a question, the system retrieves the most similar chunks and combines them (up to 2,000 characters total) to provide context. FLAN-T5-small (60M parameters, temperature=0.2) then generates an answer based only on the retrieved information using an instruction-based prompt.

### 2.2 Enhanced RAG Pipeline

The production system implements two complementary enhancements. First, query rewriting using FLAN-T5-base generates semantic variations of user queries, with aggregated retrieval across variations yielding 30 initial candidates. Second, cross-encoder/ms-marco-MiniLM-L-6-v2 reranks these candidates via joint query-document encoding, selecting top-5 for context assembly. The LLM was upgraded to FLAN-T5-base (220M parameters) for improved instruction-following. Context assembly includes grounded citations with chunk IDs and relevance scores for transparency.

## 3. Experimental Results

### 3.1 Prompting Strategy Evaluation

Three prompting strategies were tested on 100 questions using the top-1 retrieved passage: Chain-of-Thought (CoT), Persona, and Instruction prompting. Persona prompting performed best (F1=31.38%, EM=28.0%), achieving 2.5 times higher F1 score than CoT (12.33%) and 20% better than instruction-based prompting (26.12%). This success likely comes from FLAN-T5's training alignment with role-based prompts, which encourages direct, confident answers. CoT performed poorly (F1=12.33%), possibly because asking the small model (60M parameters) to show its reasoning steps overwhelms its limited capacity, making responses longer but less accurate.

### 3.2 Parameter Experimentation

A systematic evaluation of 10 RAG configurations was conducted using 150 test queries, examining the interaction between embedding dimensionality and retrieval strategies. The experimental matrix compared two embedding models, MiniLM (384 dimensions) and MPNet (768 dimensions), across five retrieval approaches: top-3, top-5, and top-10 passage concatenation, plus top-3 and top-5 Maximum Marginal Relevance (MMR) selection.

The optimal configuration combined MPNet-768D embeddings with top-5 passage concatenation, achieving an F1-score of 42.29% and Exact Match accuracy of 34.67%. This represents a 34.8% performance improvement over the naive baseline system.

**Key Findings:**

**Embedding Impact:** MPNet-768D consistently outperformed MiniLM-384D by an average of 2.8 F1 points across all retrieval strategies, demonstrating that higher-dimensional embeddings capture richer semantic relationships despite increased computational cost.

**Retrieval Volume:** Top-5 emerged as optimal across both embedding models. Top-3 retrieval struggled from insufficient context coverage, while top-10 introduced excessive noise that overwhelmed FLAN-T5's 512-token context limit, degrading answer quality.

**MMR Effectiveness:** MMR showed mixed results. It improved top-3 retrieval by ensuring diversity-based coverage, but degraded top-5 performance by penalizing relevance and removing pertinent passages. The computational overhead (requiring a 3× candidate pool) was not justified by performance gains.

### 3.3 Advanced Evaluation with RAGAs

The naive and enhanced systems were evaluated on 100 questions using the RAGAs framework with Gemini-2.5-Pro as the judge LLM. Evaluation was conducted using ThreadPoolExecutor with 10 concurrent workers to accelerate API calls through parallel processing. A small number of samples (estimated 2-5%) returned NaN values. The remaining 95-98 samples provide statistically meaningful comparisons between systems.

Four metrics assessed different quality dimensions:

**Faithfulness** measures factual consistency between generated answers and retrieved context. Tests whether answers are grounded in evidence without hallucination. Enhanced system achieved 0.670 vs naive's 0.427, representing 56.9% improvement. This validates that cross-encoder reranking surfaces more relevant evidence, enabling FLAN-T5 to generate factually accurate responses.

**Answer Relevancy** evaluates semantic alignment between generated answers and user questions. Tests whether answers directly address what was asked. Scores improved from 0.190 (naive) to 0.271 (enhanced), a 42.6% gain. Query rewriting's semantic variations possibly helped retrieve context better aligned with user intent, enabling more relevant answer generation.

**Context Precision** quantifies what proportion of retrieved passages are actually relevant to answering the question, measuring retrieval quality. The 58.4% improvement (0.380→0.602) directly reflects cross-encoder reranking's superior discriminative power compared to bi-encoder cosine similarity.

**Context Recall** measures whether all information needed to answer the question was retrieved. The dramatic 75% improvement (0.36→0.63) demonstrates query rewriting's effectiveness at casting a wider retrieval net through semantic variations, ensuring critical information isn't missed due to vocabulary mismatch.

Overall**,** the enhanced system showed consistent improvements across all dimensions, with particularly strong gains in context recall (+75%) and precision (+58.4%). This validates the combined effect of query rewriting (improving recall) and cross-encoder reranking (improving precision). The moderate answer relevancy improvement (42.6%) suggests room for further enhancement through advanced prompting or LLM fine-tuning.

## 4. Enhancement Analysis

### 4.1 Query Rewriting

Query rewriting solves the problem of vocabulary mismatch by creating different versions of the user's question using FLAN-T5-base. Each rewritten question is used to search the database separately, and the results are combined by keeping the most relevant passages based on their similarity scores. This approach retrieves 30 initial passages compared to just 5 in the basic RAG system. Test queries typically produced 1-2 rewritten versions, retrieving 30-35 unique passages and providing better coverage of relevant information.

### 4.2 Cross-Encoder Reranking

Cross-encoder reranking uses the ms-marco-MiniLM-L-6-v2 model to re-score the expanded set of retrieved passages. Unlike the initial retrieval system that scores queries and passages separately, the cross-encoder processes each query-passage pair together, allowing it to better understand how well they match. This produces more accurate relevance scores, which ranged from 10.3 (highly relevant) to -2.2 (irrelevant) in testing. The top-5 passages selected after reranking showed strong alignment with correct answers.

The cross-encoder must process each passage individually with the query, requiring 30 separate model runs per query. This was optimized using batch processing (32 pairs at a time), but still created a speed bottleneck.

### 4.3 Combined Benefits

The two enhancements work well together: query rewriting finds more relevant passages (improving recall), while cross-encoder reranking filters out the best ones (improving precision). This two-step approach mirrors real production systems that use fast retrieval first, then slower but more accurate reranking second.

The system also includes citations showing which passages were used and their relevance scores, making it easier to verify answers, debug problems, and understand the system's confidence. Upgrading the language model from FLAN-T5-small to FLAN-T5-base (60M to 220M parameters) improved its ability to follow instructions and handle complex questions.

## 5. Production Considerations

### 5.1 Scalability Challenges

The enhanced RAG system introduces notable latency through its two-stage architecture. Cross-encoder reranking requires processing 30 query-passage pairs per question, while query rewriting adds multiple LLM calls. Single-threaded processing limits query throughput significantly. Scaling beyond the current 1,000-passage test set to production-scale document collections would require approximate nearest neighbor search methods (HNSW, IVF indexing) rather than the current exact search approach. Memory requirements grow linearly with corpus size, as each document chunk requires storage of 768-dimensional embeddings.

### 5.2 Deployment Trade-offs

The naive and enhanced pipelines serve different use cases. The naive system offers faster, more predictable response times with simpler infrastructure requirements, making it suitable for applications requiring consistent latency and high confidence in precision. The enhanced system achieves superior recall and context coverage through query rewriting and reranking, better suited for exploratory search or comprehensive information gathering, though at the cost of increased latency and computational requirements.

### 5.3 Current Limitations

Query rewriting quality varies, some variations become too generic or drift from the original question's intent. FLAN-T5's 512-token limit constrains context window size, potentially excluding relevant information for complex questions. The evaluation dataset (100-150 questions) provides proof-of-concept validation but lacks the scale needed for production confidence intervals.

### 5.4 Recommendations for Deployment

For production deployment, several practical improvements would help. The cross-encoder reranking step is currently the main bottleneck, processing passages one at a time. Batch processing (which worked well in our evaluation with ThreadPoolExecutor) could speed this up significantly. Caching frequently asked questions would avoid repeated processing of common queries. Continuous monitoring should track response time and periodically sample answer quality to detect performance degradation over time.

Future enhancements could include splitting documents more intelligently based on meaning and structure rather than fixed sizes, combining embedding-based search with traditional keyword search (BM25) for better retrieval, and routing different question types to specialized pipelines. Additionally, domain-specific prompts (e.g., "Wikipedia expert" instead of generic "subject matter expert") could improve answer quality.