

# CRAG - Customized RAG to Serving Diverse User Needs

**Author:** Peter Liu

**Date:** December 4, 2025

## Executive Summary

This proof of concept demonstrates a Retrieval-Augmented Generation (RAG) system serving research engineers and marketing staff from a unified knowledge base. Through in-context learning and tuning RAG parameters—chunk size, overlap, and search-k—the system improved faithfulness from 0.56 to 0.70 and relevancy from 0.61 to 0.78 without costly retraining. Prompt engineering, chunking strategies, and evaluation frameworks proved critical to success. We recommend phased implementation using the following approach: open source models for cost-sensitive operations, with persistent vector storage and enhanced monitoring in a cloud environment.

## Introduction

This project addresses serving multiple personas from a single knowledge base. Research engineers require technical precision while marketing staff need accessible, content-ready outputs. Maintaining separate documentation systems is costly and creates synchronization issues.

The system implements a RAG architecture with two persona-specific API endpoints using different templates: technical depth for researchers, polished content for marketing. The document store contains research papers, blogs, and Wikipedia articles, updatable without system reworking. The system strictly uses document store information and avoids hallucination by answering "I don't know" when relevant information is absent from the context. A substantial evaluation framework assesses system performance.

## Key Findings

- **Prompt engineering proved highly effective without costly fine-tuning.** Techniques including output examples, explicit "what-not-to-do" instructions, and strict formatting guidelines significantly improved performance through in-context learning, making it a high-leverage optimization approach avoiding expensive post-training.
- **Retrieval performance is extremely sensitive to chunk size, overlap, and search-k parameters.** Smaller chunks improved precision for research queries, while larger chunks and search-k benefited marketing outputs. Overlap showed slight negative effects on marketing faithfulness. Systematic RAGAS evaluation was resource-intensive.
- **LLM-as-Judge enables persona-specific assessment but requires careful engineering.** Initial prompts produced undifferentiated ratings, but refined techniques yielded well-distributed scores. Separate templates for each persona proved necessary, though each rating requires an LLM inference call, adding significant time and token costs.
- **RAG system costs extend beyond implementation to ongoing operations.** Local deployment incurs maintenance and scaling costs. Commercial models offer superior performance but introduce recurring token costs that scale with usage. Organizations must budget for either

infrastructure or token expenses.

- **Privacy concerns arise with proprietary or sensitive information.** Commercial models introduce data leak risks as queries transmit externally. Local models drastically reduce privacy risks by keeping data on-premises, though requiring infrastructure and potentially lower performance.

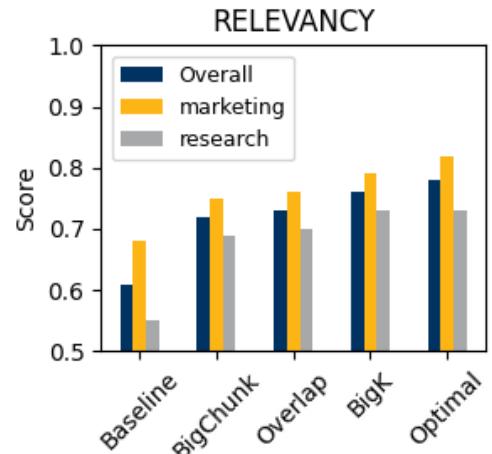
## Methodology

### Technical Approach

The system uses locally hosted, quantized Mistral-7B-Instruct-v0.3 rather than commercial alternatives. Initial development with Cohere's API exhausted trial credits rapidly despite faster inference. The open-source model demonstrated comparable performance without recurring costs, trading modest speed for cost predictability and eliminating rate limits. Model parameters remained at default as performance proved adequate. Future iterations could explore temperature and top-p tuning per persona—higher temperature for marketing creativity, lower for engineering consistency.

The system implements separate API endpoints with distinct RAG templates optimized per persona. Both retrieve from the same document store but generate persona-appropriate responses through template customization. Using a single template proved challenging as attention mechanisms would need to serve divergent objectives simultaneously.

Separate templates enabled precise persona control. Engineering templates emphasize accuracy and technical details; marketing templates focus on accessibility and polished language. Both employ prompt engineering including one-shot examples, explicit instructions, strict vector database usage guidelines, and output format examples. Template separation simplified persona-specific customization—modifications to one persona don't affect the other, a critical advantage over unified approaches.



(Figure 1 - By the relevancy metric, both research and marketing responses improved, with research showing the largest gain from baseline.)

### Testing and Evaluation

Testing evaluated retrieval quality and generation appropriateness using RAGAS metrics, BERTScore, and LLM-as-Judge. Five configurations varying chunk size, overlap, and search-k were tested against the gold question-answer dataset, with results stored as JSON and analyzed separately. Initial development used 20% question subsets for efficient debugging.

Chunking strategy balanced document characteristics with query patterns. Technical documentation for engineers performed better with larger chunks for informative retrieval; conceptual articles for marketing benefited from larger chunks and “search-k” preserving narrative flow. A 10% overlap preserves boundary information but lowers Marketing performance and warrants further investigation.

Retrieval metrics assessed whether chunks addressed questions. Relevancy prevented irrelevant results; F1 scores provided balanced precision-recall measures. Generation metrics verified context consistency. Faithfulness measured consistency without fabrication; LLM-as-Judge assessed tone consistency with the persona. Both RAGAS and LLM-as-Judge required substantial execution time roughly matching inference duration.

Five configurations explored parameter space: **Baseline** (defaults), **BigK** (increased search-k), **BigChunk** (larger chunks), **Overlap** (increased overlap), and **Optimal** (balanced combination).

**BigChunk** improved marketing context; smaller chunks enhanced technical precision. Overlap showed mixed persona effects. Optimal balanced tradeoffs for best overall performance.

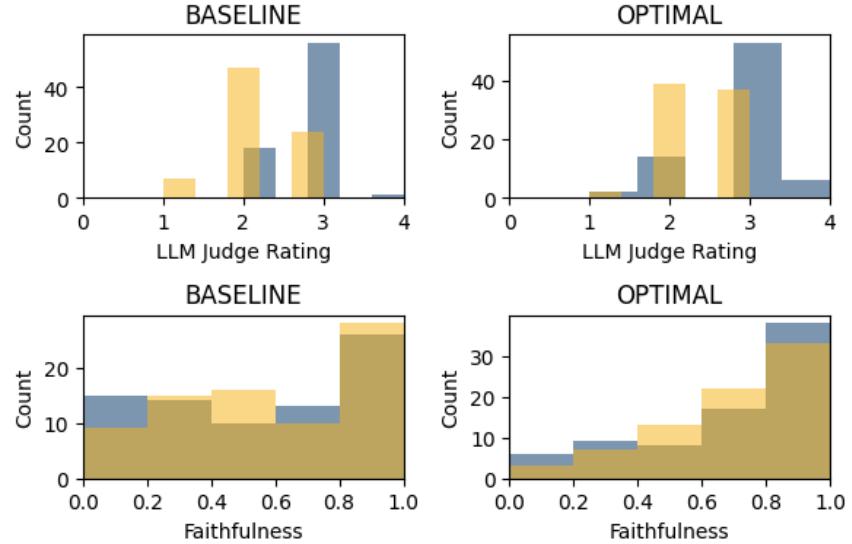


Figure 2: LLM Judge Rating and Faithfulness scores improve from the baseline to the optimal configuration, with the distribution shifting to the right, indicating better performance.

## Results and Findings

### Proof of Concept Functionality

The POC successfully demonstrates the core concept within four weeks. The system reliably answers questions from a secured document store while maintaining distinct personas. Key objectives were met: cost-conscious local deployment eliminates recurring token costs, configurable parameters enable retrieval optimization, and prompt engineering achieves improvements without expensive fine-tuning. Testing confirmed diverse question handling, persona consistency, and appropriate “I don’t know” responses when information is absent.

### Lessons Learned

Prompt engineering emerged as exceptionally high-leverage, delivering substantial gains with minimal effort. Separate templates proved far more effective than unified approaches. Chunking parameters dramatically affect retrieval quality with different optimal settings per persona. Establishing evaluation

metrics early accelerates development, functioning like test-driven development. Commercial models offer speed advantages but don't necessarily yield superior results—local Mistral performed comparably to Cohere for generation quality, with speed the primary tradeoff.

## Challenges and Limitations

Library compatibility issues presented ongoing obstacles as GenAI packages update frequently with breaking changes. RAGAS upgrades required code adaptations. Excessive metrics confuse interpretation without adding value—future efforts should identify critical metrics aligned with objectives. Local deployment provides stability as commercial vendors update models without notice, potentially breaking systems. Evaluation overhead proved substantial, roughly matching inference time and doubling computational costs.

## Next Steps

User interface development would enable end-to-end validation. Inference parameter tuning—temperature and top-p per persona—could improve generation quality. LLM-as-Judge refinement would improve evaluation reliability. Enhanced context examples could expand to multi-shot learning. Retrieval enhancements including reranking and chunk diversity analysis merit investigation. Scalability testing with larger collections and disk-based persistent storage would reveal production-scale performance characteristics.

## Summary & Recommendations

Implementation decisions depend on balancing scalability, performance, and maintenance costs. Organizations can choose between cloud-based or local deployment, and between commercial API services or self-hosted models. Data governance requirements may prohibit commercial APIs or public cloud deployment when handling sensitive or proprietary information, making local deployment necessary regardless of other considerations.

Numerous optimization opportunities remain unexplored. Performance improvements—both speed and quality—can be achieved through careful selection of reasoning models, embedding systems, and hardware specifications. Additional gains are possible through refined model parameters, optimized chunking strategies, and continued prompt engineering refinement. These represent valuable directions for future investigation.

Assuming data sensitivity permits, we recommend initial public cloud deployment with limited user rollout. This approach enables production feedback collection while minimizing infrastructure investment and maintaining deployment flexibility for future architecture decisions.

# References

**LLM Assistance:** This report was structured and written with assistance from Claude (Anthropic), which helped organize technical context into the required format, expand abbreviated points, and ensure consistent technical depth.

# Appendix

Table 1 – Metric performance improvement from baseline to optimal configuration (scenario).

Scenario	faithfulness	relevancy	precision	recall	f1	llm_judge_rating	size
Baseline	0.56	0.61	0.88	0.88	0.88	2.49	156
BigChunk	0.67	0.72	0.88	0.89	0.88	2.60	156
BigK	0.68	0.76	0.88	0.89	0.88	2.70	156
Optimal	0.70	0.78	0.88	0.89	0.88	2.64	156
Overlap	0.68	0.73	0.88	0.89	0.88	2.53	156

Table 2 – Metric performance by domain (Research / Marketing) showing improvements across domains.

Domain	faithfulness		relevancy		f1		precision		recall		Llm judge rating	
	marketing	research	marketing	research	marketing	research	marketing	research	marketing	research	marketing	research
BigK	0.69	0.68	0.79	0.73	0.89	0.88	0.88	0.88	0.89	0.89	2.87	2.53
Optimal	0.69	0.71	0.82	0.73	0.89	0.88	0.88	0.88	0.89	0.89	2.84	2.45
BigChunk	0.68	0.67	0.75	0.69	0.88	0.88	0.88	0.88	0.89	0.89	2.77	2.43
Overlap	0.66	0.69	0.76	0.70	0.89	0.88	0.88	0.88	0.89	0.89	2.72	2.36
Baseline	0.55	0.58	0.68	0.55	0.88	0.88	0.88	0.88	0.89	0.88	2.77	2.22

Figure 3 – Metric by domain and scenario, showing significant improvements.

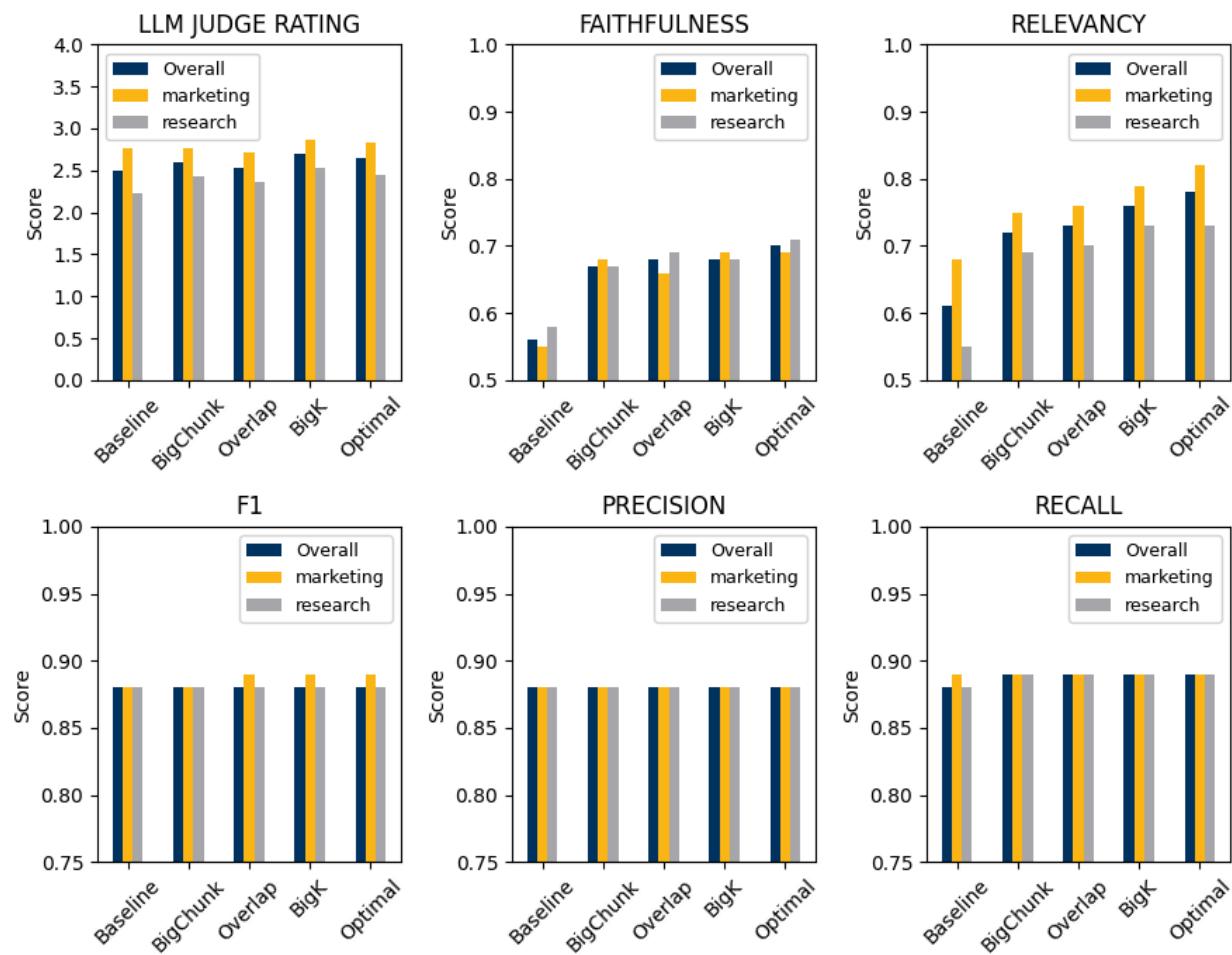


Figure 4 - Distribution of metric showing tightening of variance in metric.

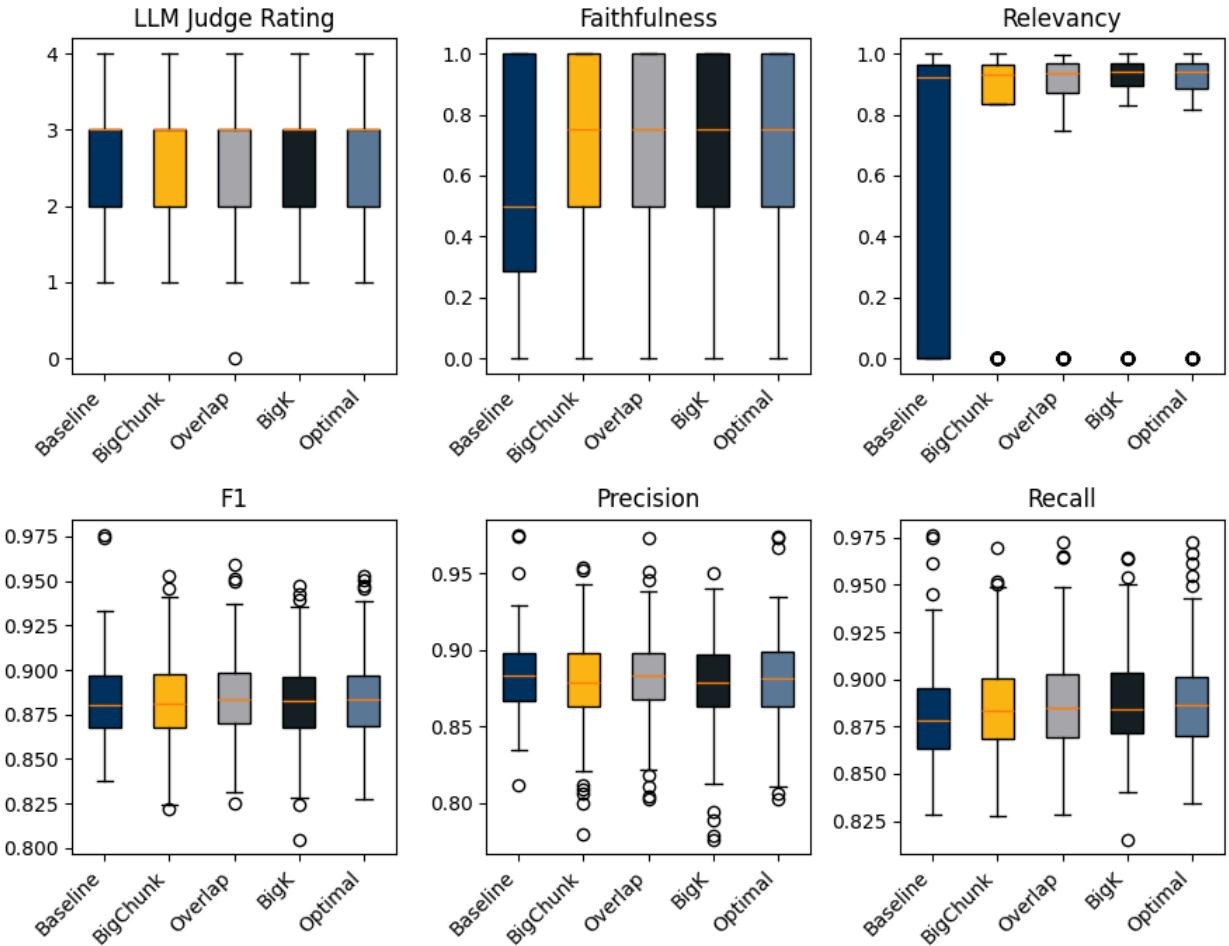


Figure 5 – Metrics shift rightward, indicating progress from parameter tuning.

