Evaluating RAG Systems for Cross-Functional Supporting Generative AI Topics

Taylor C. Han
UC Berkeley
267 Generative AI: Foundations, Techniques, Challenges, and Opportunities
Section 5
FA 2024

1. Executive Summary

This proof-of-concept examines the viability of implementing a Retrieval-Augmented Generation (RAG) system to support our engineering and marketing teams with Generative AI information. Our testing revealed that while the RAG system improved context precision and recall, there was no significant enhancement in semantic similarity or answer relevancy compared to using Cohere's language models alone. These findings indicate that Cohere's models are sufficiently effective for generating accurate and relevant responses within the Generative AI domain without the added complexity of developing and maintaining a document database. Consequently, we recommend continuing to utilize Cohere's existing models as a more efficient and practical solution for our information retrieval needs in Generative AI.

2. Introduction

To support our company's growing focus on Generative AI product rollouts and the operational needs of engineering and marketing teams, we investigate the effectiveness of utilizing a Retrieval-Augmented Generation (RAG) system for Open-Domain Question Answering (ODQA). Unlike traditional "closed-book" ODQA systems, our RAG system adopts an "open-book" approach, leveraging a document database and advanced Large Language Models (LLMs) to provide accurate, up-to-date responses. By building a context window supplemented with relevant documents, this system minimizes the need for costly retraining while still ensuring precise answers to topics not covered during pre-training.

The RAG system is designed to serve two main functions: providing the engineering team with detailed, accurate information to boost productivity and offering the marketing team clear, accessible answers to aid in content creation and product promotion. This proof-of-concept (POC) evaluates the system's effectiveness by testing and optimizing key components within this complex system such as selection of language models, chunking methods, and embedding strategies. Using a validation set of 75 questions with tailored answers for engineering and marketing, this POC aims to understand the implication for a company-wide implementation of a RAG system while also aligning with the company's quarterly product release schedule.

3. Key Findings

Our proof-of-concept uncovered several key insights into optimizing a Retrieval-Augmented Generation (RAG) system for Generative AI information retrieval. These findings highlight critical trade-offs and opportunities for improvement in document database design, chunk size optimization, evaluation models, prompt engineering, retrieval metrics, and answer quality metrics.

3.1. Key Finding 1 - Developing an Effective Document Database for Generative AI Queries

In our initial tests, we found that high-similarity scores between the user's query and the documents in the RAG database often indicated similarity, but did not always guarantee that the retrieved context added substantive value to the generated answer. For example, when querying, "What is the recommended strategy for training large autoregressive language models with limited compute resources, as contributed by the Chinchilla team?" we observed variations in retrieved context based on chunk size and overlap:

> Chunk Retrieved - 128 Chunk Size, 0 Overlap: Chinchilla contributes to developing an effective training paradigm for large autoregressive language models with limited (Cosine Similarity: 0.7854)
>
> Chunk Retrieved - 300 Chunk Size, 75 Overlap: Chinchilla contributes to developing an effective training paradigm for large autoregressive language models with limited compute resources. The Chinchilla team recommends that the number of training tokens is twice for every model size doubling, meaning that using larger, higher-quality training datasets can lead to (Cosine Similarity: 0.6831)

This example demonstrates that expanding chunk size and adding overlap can yield a more comprehensive context. Such adjustments allow for broader content capture, improving context relevance and relevant information for the query, even if it slightly decreases cosine similarity between the query and the documents.

3.1. Key Finding 2 - Impact of Prompt Complexity on Faithfulness

Our observations suggest that simpler prompts, which provide minimal guidance and focus solely on utilizing the retrieved documents, tend to yield higher faithfulness scores. By placing fewer stylistic or structural demands on the model, these basic instructions allow it to rely more directly on the supplied context. In contrast, more

elaborate prompts encourage additional reasoning and stylistic alignment, which can unintentionally introduce extra details or minor deviations from the retrieved content, thereby reducing faithfulness. We provide both the basic prompts used for benchmarks and the more advanced prompts in the appendix. Although we observed a lower faithfulness score when implementing the advanced prompts, we recommend using them due to their increases in semantic similarity and answer relevancy.

3.2. Key Finding 3 - Enhancing LLM Evaluation through Advanced Models

We initially evaluated responses from our RAG system using the open-source Mistral model within the RAGAS (Retrieval-Augmented Generation Assessment) framework, focusing on metrics like context recall, context precision, faithfulness, answer relevance, and answer similarity. LangChain was integrated with few-shot prompt engineering to align evaluations with our validation set. However, even with few-shot examples and a temperature set to zero, Mistral displayed a tendency for optimistic scoring, indicating limitations in Mistral's ability to evaluate answers in a loop. To address this, we adopted a more robust and consistent evaluation approach by leveraging OpenAI's ChatGPT, which provided significantly improved accuracy in assessing our RAG system's performance.

3.3. Key Finding 4 - Prompt Engineering: Zero-Shot Versus Few-Shot Performance

Few-shot prompting enhances language model performance by providing task-relevant examples, allowing models to adapt more effectively to new contexts (Brown et al., 2020). From the referenced article, we learn that generally, few-shot prompts outperform zero-shot and one-shot prompts and as a result becoming better at task-specific work. Building on this, we integrated a few-shot approach into our RAG system, including two example responses tailored for marketing and research contexts. This adjustment resulted in a ~1% improvement in semantic similarity for marketing responses and a ~0.4% increase for research responses, aligning better with golden answers.

3.4. Key Finding 5 - Impact of Increased Retrieval Metrics on Semantic Similarity

During testing, we noted that optimizing parameters like chunk size, prompting, and overlap improved retrieval metrics—specifically context precision, context recall, and faithfulness—but yielded minimal gains in answer relevancy and semantic similarity. We suspect that expanding chunk size to increase recall and precision can inadvertently introduce extraneous details, reducing alignment with the target content. Given more time, dynamically adjusting the number of retrieved chunks could improve answer relevancy and semantic similarity, which already showed strong baseline scores. Ultimately, we prioritize semantic similarity over strict faithfulness to ensure that the system's responses remain both contextually accurate and stylistically aligned with research and marketing requirements.

4. Methodology

This section of the report outlines the technical approach for constructing a Retrieval-Augmented Generation (RAG) system designed to improve document search for a Q&A chatbot focused on Generative AI. The system's primary goal is to efficiently retrieve relevant documents and generate responses to questions on topics such as foundational concepts, historical developments, recent innovations, and other advancements in the field. The following sections discuss the RAG system architecture, the methodology for testing response quality, and the evaluation benchmarks applied to enhance accuracy and relevance, addressing the specific needs of both engineering and marketing teams.

5. Technical Approach

We developed a RAG system that leverages LangChain's capabilities to streamline document retrieval for our Q&A chatbot. LangChain provides a systematic approach for constructing pipelines that follow a sequence which we have set up covering modular parts such as the document database, retriever, and the LLM. This approach enhances the efficiency and organization within our RAG setup. Using its invoke operator, LangChain enables integration with various data sources and storage solutions, enabling document storage and retrieval capabilities for the system (LangChain, n.d.).

5.1 Prompt Engineering

We leverage LangChain's capabilities to support adaptive prompt engineering, allowing us to tailor responses to different users in our engineering and marketing teams. We customized prompts to adjust the RAG system's output style, ensuring to influence the output of the LLM's generated responses in the RAG system to align closely with the style provided for the research and marketing questions in the validation set. Further, we applied chain-of-thought reasoning, incorporated techniques inspired by Ethan Mollick (Mollick, 2023), and included a few-shot prompt to give the LLM a feel of the research and marketing styled responses. Since style can be challenging to convey with descriptions alone, we utilize few-shot examples to provide a clear expectation between style differences for the LLM. Additionally, we placed constraints on answer length, avoided bullet points to maintain consistency, and emphasized the model's high capability to ensure that the model approached tasks with confidence. The finalized prompts for both the engineering and marketing team are presented in Figure 2, in the appendix.

5.1 Large Language Model Selection

Our RAG system leverages a LLM at the end of the LangChain pipeline. Cohere's capabilities allow it to summarize context and generate responses using both the provided documents and its pretrained knowledge.

We chose Cohere over Mistral after testing semantic similarity on our validation set, finding Cohere slightly better aligned for both marketing and research responses. Additionally, Cohere's enterprise-optimized models offer scalability and efficiency suitable for Q&A chatbots (Cohere, 2024). Although Mistral employs a sliding window attention mechanism for computational efficiency, Cohere provided notably faster inference speeds. Given that our marketing and research teams will frequently query Generative AI topics—especially at quarter's end—faster inference emerged as the most crucial factor, as both models produced similarly high-quality responses.

5.2 Embedding Engine

When choosing an embedding engine, we opted for sentence transformers to vectorize both validation answers and outputs. Sentence embeddings capture entire sentence meanings rather than focusing on individual words, improving semantic alignment and reducing complexity (Yadav, 2024). Among several options, we selected "multi-qa-mpnet-base-dot-v1" for its suitability in semantic search tasks and its large pre-training dataset of 215 million question-answer pairs. We note a 512-token processing limit for this model, which we considered when configuring our RAG system to prevent chunk truncation (Hugging Face, 2023).

5.3 Database

To support high-quality responses in Generative AI related topics, we built a vector similarity search engine with Qdrant and a datastore containing texts from sources like ArXiv for scholarly articles maintained and operated by Cornell University, several blog posts from Lily Weng who currently works on practical AI safety and Alignment at OpenAI, and Wikipedia entries on Generative AI.

6. Testing and Evaluation

We tested our RAG system using two different benchmark: First, we used the average sentence embeddings of all of the golden questions and research, marking answer pairs between the results of the mistral and cohere models to determine the LLM of choice for our RAG system and then utilized the RAGAS framework to test for optimal hyperparameters.

We note that having a higher vector similarity between the generated responses by the LLM and the golden answers would be an effective way to tune the RAG system to capture the stylistic differences between the research and marketing questions. Further, the RAGAS framework was determined to be the best method compared to other evaluation metrics such as BLEU and ROUGE because RAGAS is designed to help provide insights in retrieval metrics and answer generation quality metrics by capturing the nuances that come with evaluating language (Kaarthick, D., 2023).

The categories within the RAGAS framework were answer relevancy, faithfulness, context precision, context recall and semantic similarity**.** We define each metric as defined in (Tiwani, 2023). Further, we also utilize an OpenAI key leveraging ChatGPT's 4o Mini model to evaluate our RAG responses for reasons described in Key

Finding 3[1]. During testing and hyperparameter tuning, we focused on increasing semantic similarity because it closely aligns with the stylistic differences we want to achieve between marketing and research responses. Additionally, higher semantic similarity also serves as a proxy for correctness, as it indicates that the generated answers closely match the standards set by our golden answers.

7. Results

Model	Function	faithfulness	answer_relevancy	context_precision	context_recall	semantic_similarity
Baseline	Research	0.2186	0.9023	0.3267	0.2693	0.9381
Baseline	Marketing	0.2412	0.8392	0.2237	0.3633	0.9165
Final with base prompt	Research	0.5895	0.8786	0.5019	0.3233	0.9351
Final with base prompt	Marketing	0.5372	0.8756	0.4567	0.3698	0.9172
Final	Research	0.3755	0.8943	0.5033	0.3364	0.9399
Final	Marketing	0.3766	0.8905	0.4696	0.3742	0.9264

Figure 3: Results for stylized marketing and research responses compared to the validation set from prompt engineering, chunk size, and overlap tuning. Results are the average across all 75 validation question and answer pairs for each metric in our provided golden set. Both the baseline model and final models are leveraging "multi-qa-mpnet-base-dot-v1" for sentence embeddings and Cohere Chat Model.

Our results from tuning our RAG system to match gold question-answer pairs were compared against a baseline model, which used a generic chain with minimal prompting, a 128-token chunk size, and zero overlap. Our final model utilizes Cohere as the LLM, with 0.2 temperature as suggested in this paper by Hopsworks (Hopsworks, n.d.) for production level models.

Further, an extensive part of our tuning lies in how documents are partitioned in our document database consisting of documents related to Generative Artificial Intelligence topics and prompt engineering our pre-trained large language model, Cohere. For our final chunk size, we utilize a 350 chunk size with a 75 overlap. With a higher overlap, we are able to partition documents in more slices; thereby, increasing the chances of grabbing the most relevant 350 token chunk for each query. For our document database, we utilize recursive character splitting and for the number of documents retrieved with each retrieval, we utilize four documents from our document database to supplement our queries. Lastly, we utilize a score threshold in our retriever of 0.4 after iterative testing, we note that using a score threshold of 0.7 leads us to the highest results. The intuition was to not bring in documents that could potentially be far away from our initial question leading the LLM at higher risk of hallucinations.

Outcome 1: Semantic Similarity to Golden Answers
As discussed in Finding 5, After implementing changes compared to the baseline model, we achieved an increase in semantic similarity to the golden answers of 0.18 percentage points for research responses and 1 percentage points for marketing responses. Although these improvements may seem small, they represent the average across all validation question and answer pairs. Adjustments such as increasing context length can significantly improve responses for some questions but may also risk reducing response quality for other questions.

Notably, we observed that the most substantial improvements were linked to adjustments in chunk size and overlap in our document database, rather than the detailed prompt enhancements made within LangChain.

Outcome 2: Faithfulness
In addition to the improvements in context precision and context recall discussed in Finding 5, we observed a significant increase in faithfulness following our final model adjustments. Specifically, faithfulness increased by 15.69 percentage points for research responses and 13.7 percentage points for marketing responses.

[1] Initial evaluations with Mistral in RAGAS showed optimistic scoring, even with adjustments. We switched to OpenAI's ChatGPT for more reliable and consistent evaluations. (See Finding 3 for a detailed explanation)

These substantial improvements in faithfulness indicate that our final model more accurately utilizes the retrieved context when generating responses, which in turn reduces instances of the LLM leveraging its pretrained knowledge to generate answers. This improvement is critical for both the engineering and marketing teams, as it shows the ability for LLMs to ground their answers based on tailored document databases.

The increased faithfulness suggests that our adjustments to chunk size, overlap, and prompt engineering enhanced the model's ability to generate answers utilizing the retrieved context, further proving the idea behind not needing to retrain an LLM to generate answers based on context that the LLM may not have been pre trained on.

8. Discussion and Recommendations

In conclusion of this proof-of-concept, we present our perspective on implementing a Retrieval-Augmented Generation (RAG) system to assist our marketing and research teams with Generative AI queries. We oppose adopting a RAG system for our company for two primary reasons:

First, our analysis reveals that the RAG system relies heavily on pretrained information. This is demonstrated by the small and subtle improvements in semantic similarity and answer relevancy metrics, despite significant gains in faithfulness and context metrics during our testing. These findings suggest that Cohere would have been able to answer questions in our validation set without the need of an extensive document database covering Generative AI topics. Thus, we conclude that in our use-case, Cohere is primarily leveraging its extensive knowledge in Generative AI gained from its pre-training process rather than treating the documents retrieved from our document databases as a necessity.

Second, we believe that prompt engineering alone allows us to adjust the style of the language model's outputs effectively. This means we can obtain relevant answers using Cohere's pretrained knowledge and modify the response style through prompt engineering without the need to maintain a document database specific to Generative AI.

However, we would like to note that we recognize that a RAG system could be advantageous if our documents are more current and beyond the scope of Cohere's pretrained models. In these cases, or if we aim to stay updated with the latest advancements in the Generative AI industry, we would support the adoption of a RAG system.

9. Challenges and Limitations

Finally, we discuss challenges and limitations, especially in recommending our initial RAG prototype for the research and marketing teams. The Generative AI landscape evolves rapidly, exemplified by OpenAI's o1 model (December 2024), which significantly enhances reasoning (Wong, 2024). Incorporating documentation on such advancements would let models like Cohere accurately address cutting-edge topics. However, our testing with Cohere's 'command-r-08-2024' model revealed a limitation: when asked about the o1 model, it hallucinated and referred to GPT-1, illustrating the need for a robust RAG system to quickly update and maintain current information.

Looking ahead, we recommend using RAG selectively, triggering it only for specialized queries while relying on Cohere's pretrained knowledge for general topics. This strategy reduces maintenance overhead on the document database and focuses on updates where they add the most value. We also acknowledge RAGAS's complexity as an evaluation framework and suggest involving domain experts through methods like DPO or RLHF. Human-in-the-loop feedback can refine both RAG and Cohere's models, leading to more accurate, contextually rich responses.

In conclusion, while Cohere handles most Generative AI queries well, selectively integrating a RAG system can enhance context for specialized, time-sensitive inquiries. Moving forward, we advise a balanced approach that dynamically leverages Cohere's inherent strengths alongside curated documents, guided by continuous human feedback for ongoing improvement.

References

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language models are few-shot learners. *arXiv*. https://arxiv.org/abs/2005.14165

Cohere. (2024). The leading AI platform for enterprise. Retrieved from https://cohere.com/

Hugging Face. (2023). sentence-transformers/multi-qa-mpnet-base-dot-v1. *Hugging Face*. Retrieved from https://huggingface.co/sentence-transformers/multi-qa-mpnet-base-dot-v1

Hugging Face. (n.d.). multi-qa-mpnet-base-dot-v1. Retrieved November 3, 2024, from https://huggingface.co/sentence-transformers/multi-qa-mpnet-base-dot-v1

Kaarthick, D. (2023, October 26). Ragas for RAG in LLMs: A comprehensive guide to evaluation metrics. *Medium*. https://dkaarthick.medium.com/ragas-for-rag-in-llms-a-comprehensive-guide-to-evaluation-metrics-3aca142d6e38

LangChain. (n.d.). Why LangChain? Retrieved November 3, 2024, from https://python.langchain.com/docs/concepts/why_langchain/

Tiwani, N. (2023, September 20). Evaluate RAG with RAGAS. *Medium*. https://namratanwani.medium.com/evaluate-rag-with-ragas-e1ad1aa99c2e

Wong, M. (2024, December 6). The GPT era is already ending. The Atlantic. Retrieved from https://www.theatlantic.com/technology/archive/2024/12/openai-o1-reasoning-models/680906/

Yadav, S. (2024, September 18). Understanding similarity with word and sentence embeddings. *Medium*. https://medium.com/@sagarmi4545/understanding-similarity-with-word-and-sentence-embeddings-9d12773163ce

Appendix

Research RAG System Prompt	Marketing RAG System Prompt
[INST] You are an exceptionally capable AI Question and Answer chatbot specializing in Generative AI concepts for engineers in a technology company. Assume all questions pertain to Generative AI topics, regardless of how general they may seem. Many critical decisions rely on your responses, so accuracy and clarity are essential. Think this through carefully, step by step, and remember: you truly can do this. {context} **Answer Structure:** - Frame your answer as a clear, single-paragraph response without lists or indentations. - Provide depth and clarity in answers, aiming for 2 to 4 sentences that blend core insights with relevant technical details. - Focus on practical applications and implications in Generative AI, ensuring completeness without over-simplification. **Tone and Style:** - Use a precise, confident tone suitable for a technical audience. - Integrate relevant terminology without dense jargon; keep answers clear and to the point. **Examples of Answer Style:** 1. *Question*: “How does retrieval-augmented generation enhance large language models?” *Answer*: "Retrieval-Augmented Generation (RAG) enhances large language models (LLMs) by integrating them with an external retrieval mechanism. This architecture allows the model to access and incorporate pertinent information from external knowledge bases during the generation process. By retrieving and conditioning on relevant documents, RAG mitigates issues like hallucination and outdated knowledge inherent in standalone LLMs, thereby improving accuracy and reliability in tasks requiring up-to-date or domain-specific information." 2. *Question*: “What is the role of self-attention in transformers?” *Answer*: “Self-attention mechanisms in transformer architectures compute the relevance of each token in a sequence relative to others, enabling the model to capture dependencies regardless of their distance. This process involves generating attention scores that determine the influence of each token on the representation of others, facilitating the modeling of complex relationships within the data. Such capability is crucial for tasks like machine translation and text summarization, where understanding contextual nuances across entire sequences is essential.” Remember, engineers depend on your response—think through each answer carefully. **Question:** {question}	[INST] You are a remarkably capable AI chatbot answering questions on Generative AI for marketing professionals in a technology company. Assume all questions relate to Generative AI, even if they appear general. Keep in mind that people are counting on you to explain these concepts well. You’ve got this—take a deep breath, think step by step, and make each answer clear and impactful. {context} **Answer Structure:** - Provide answers in a single-paragraph statement without lists or indentations. - Keep responses concise yet informative, ideally in 1 to 3 sentences, emphasizing the main point. **Tone and Style:** - Use a clear, approachable tone that makes complex concepts understandable. - Highlight practical applications and impacts, focusing on accessibility for a non-technical audience. **Examples of Answer Style:** 1. *Question*: “How does retrieval-augmented generation enhance large language models?” *Answer*: "Retrieval-Augmented Generation (RAG) improves large language models by integrating them with external information sources. This combination allows the model to access up-to-date and specific data during response generation, enhancing accuracy and relevance, especially for tasks requiring current or specialized knowledge." 2. *Question*: “What is the role of self-attention in transformers?” *Answer*: “Self-attention in transformers enables the model to assess the importance of each word in a sentence relative to others. This mechanism helps the model understand context and relationships within the text, leading to more accurate interpretations and outputs in tasks like translation and summarization.” Remember, marketing depends on your response—each answer should be clear, friendly, and insightful. **Question:** {question} [/INST]

Figure 2: Finalized Prompts for marketing and engineering

Research RAG System Prompt	Marketing RAG System Prompt

```
research_template = """[INST]
Please answer the following question using the
context below:
{context}

**Question:** {question}
[/INST]
"""
```

```
marketing_template = """[INST]
Please answer the following question using the
context below:
{context}

**Question:** {question}
[/INST]
"""
```

Figure 3: Baseline Prompts