Implementation and Evaluation of RAG Systems for QA

Executive Summary:

This project focused on developing and testing a Retrieval Augmented Generation (RAG) system to improve our company's question answering and search capabilities in order to accelerate both engineering tasks and the marketing team's production. My proof of concept (POC) explored the capabilities of a RAG system in effectively assisting both the engineering and marketing team in their day to day as well as rolling out new Generative Artificial (Gen AI) Intelligence products. For this POC, I conducted extensive experimentation of various RAG system configurations with different large language models (LLMs), prompts, embedding models, chunk sizes, overlap amounts, and model temperatures to determine the one that generated the best responses for the marketing team and the engineering team. Different configurations yielded optimal results for each team due to their unique requirements; the engineering team required a system that provided more technically detailed responses, while the marketing team required a system that prioritized clarity and more simplified answers. The best performing configuration for the engineering team achieved an average cosine similarity score of 0.80, BLEU score of 0.038, and ROUGE-2 score of 0.09. The best performing configuration for the marketing team achieved an average cosine similarity score of 0.73, BLEU score of 0.037, and ROUGE-2 score of 0.067. These configurations did not yield significant improvements when compared to the baseline; therefore, I recommend further experimentation of configurations, the use of more robust evaluation metrics, the addition of more documents, and model fine-tuning.

Introduction:

With the rapid evolution of artificial intelligence, the ability to efficiently retrieve and answer questions using large amounts of data has not only become possible but also very crucial for technological advancement. This project addresses the challenge of effectively developing question-answering tools. My focus was on developing and evaluating several RAG system configurations that leverage LLMs to improve the relevance of answers generated using documents that discuss Natural Language Processing (NLP) and Gen AI concepts. These responses will help the marketing and engineering teams' everyday tasks and assist them in developing Gen AI products and marketing campaigns.

The RAG system was engineered to handle a broad range of queries from highly technical questions aimed at the engineering team to more simple, less technical queries from the marketing team. The inputs for the RAG system are text queries related to NLP and Gen AI. Since the engineering and marketing teams have differing needs, the systems I developed are customized to work best for each respective team.

Key Findings:

1. The configurations that I developed and evaluated resulted in some improvement, but they were not significant when compared to the baseline. For example, the cosine similarity score for the engineering team's baseline configuration was 0.69, while the optimized configuration was 0.80. This shows that further investigation and enhancement is necessary to optimize the RAG system outputs for our use case. It may be necessary to do more than just modify existing parameters, such as model fine-tuning and/or increasing the diversity of documents included in the vector store.

2. Configurations that processed the text into chunks with overlap consistently outperformed those without, emphasizing the importance of maintaining context for achieving better responses. For example, the cosine similarity score for the marketing team's baseline configuration that contained no overlap was

0.71, while the optimized configuration with an overlap of 25 was 0.73. This shows that modifying the chunking and overlap parameters have an impact on the model's performance.

3. Configurations with prompts that included more detailed, specific instruction consistently outperformed configurations that used more vague prompts. This highlights the value of prompt engineering in helping the model generate more useful outputs.

4. Evaluating the configuration with a variety of metrics is crucial, as each metric offers different insights, providing a more comprehensive assessment of its overall performance. For example, cosine similarity measures the semantic similarity between two pieces of text, while the BLEU score measures the overlap of n-grams between the generated response and the gold answer. Depending on the use case, different metrics will be prioritized.

5. The engineering team's configurations scored consistently higher when compared to the marketing team's configurations. This discrepancy indicates that the document store contains more technically detailed information that is suitable for the engineering team but less suitable for the marketing team. This suggests that a more diverse source of documents would be beneficial, so that the RAG system can be effective for both teams.

Methodology:

Technical Approach

Developing the RAG system required several technical choices. I started by sourcing ArXiv papers, Lilian Weng's blog posts, and Wikipedia articles on relevant NLP and Gen AI concepts to include in my vector store, which is the foundation of building a RAG system. I set the chunk size to 128 and the overlap amount to 25. This also required choosing a specific embedding model for converting the text into vector representations, which is crucial for retrieving the most relevant documents. I chose all-distilroberta-v1 for its robustness in NLP tasks and multi-qa-mpnet-base-dot-v1 as it is optimized for question answering tasks. Then, I designed separate prompts for the engineering and marketing teams catering to their different needs. I decided to use Mistral-7B-Instruct and Cohere's LLM, as they are both designed to understand and generate human-like text.

Testing and Evaluation

To evaluate the performance of the various RAG systems, I used a set of 75 NLP and Gen AI-related questions that contained separate predefined gold answers for the marketing and engineering teams. This allows me to directly compare the generated responses to these gold answers to get an understanding of how well the RAG system performed. I evaluated the baseline configuration on all 75 questions but randomly sampled 25 questions when testing the seven configurations, due to a lack of resources. To measure their performance, I utilized three different metrics: cosine similarity, BLEU score, and ROUGE-2 score. Cosine similarity was used to measure the semantic similarity between the generated responses and the gold answers, providing an idea of how well the model captured the underlying meaning. BLEU score was used to evaluate the grammatical and syntactical precision of the generated responses. ROUGE-2 score was used to capture the overlap of bi-grams to determine how much of the key text from the gold answer was also included in the generated responses. The combination of these three metrics provided a comprehensive evaluation framework, as I was able to capture both semantic and syntactic similarities.

To determine the optimal RAG system configuration, I started by establishing a baseline with multi-qa-mpnet-base-dot-v1 for embedding, a chunk size of 128 without overlap, a model temperature of

0.6, and simple prompts, using Mistral-7B-Instruct-v0.2. From this baseline, I systematically tested seven different setups by modifying key parameters: embedding models, chunk sizes, overlap amounts, model temperatures, and prompt complexities. I also tested two different LLMs: Mistral-7B-Instruct-v0.2 and Cohere's proprietary model. For the embedding models, I experimented with multi-qa-mpnet-base-dot-v1 and all-distilroberta-v1. For chunk sizes and overlap amounts, I tested 3 setups: chunk size of 128 with 0 overlap, chunk size of 128 with 25 overlap, and chunk size of 300 with 50 overlap. This provided me with insight into the impact of having varying amounts of context. For model temperature, I tested 0.2, 0.6, and 0.8 to get an understanding of how different values affect the randomness of the responses. For the prompt, my design philosophy was that there would be two separate prompts that are tailored to the research and marketing teams' differing levels of expertise of complex NLP concepts. I made sure to include very clear instructions so that the provided answers would be useful for each team. The engineering team's prompt was designed to provide more detailed and technical responses. The marketing team's prompt was designed to provide less technically detailed answers without using complex jargon that they might have trouble understanding. The baseline and improved prompts will be included in the Appendix section. To determine how well the configurations performed, I compared their evaluation metrics to the baseline's metrics.

Results and Findings:

Proof of Concept Functionality:

The POC for the engineering and marketing teams' successfully demonstrated the core idea of using RAG systems to enhance question answering capabilities within a tech company. While the metrics of my configuration did not show significant improvement over the baseline, it definitely showed promise for further improvements with more comprehensive experiments and testing. The responses for both teams varied in complexity, with the engineering teams' responses being more technically detailed and the marketing teams' responses being more general, providing a high-level explanation. The objectives were met; however, there is room for improvement.

Lessons Learned:

Throughout my experimentation process, I gained many technical and non-technical insights. Let's begin with discussing the technical insights. First, a smaller chunk size of 128 proved more effective than larger sizes like 300, suggesting that the granularity of information contained in each chunk can have a significant impact on the retrieval accuracy. In addition, having a small amount of overlap between the chunks, such as 25 provides a more relevant response, as more of the context is being preserved. If important pieces of context are missing, there is a higher chance of generating inaccurate responses. Another insight I discovered was that the choice of embedding model plays an important role, with multi-qa-mpnet-base-dot-v1 outperforming all-distilroberta-v1 for the most part due to its optimization for question answering tasks. My experiments also showed the importance of prompt design to customize responses for different use cases. For example, the prompt I built for the engineering team specified that the answers should be longer and contain technical depth. The prompt I built for the marketing team specified that the team lacked a technical background in NLP so the responses should be shorter and not contain complex jargon. The RAG Configuration Results table below contains the performance of the baseline configuration, the two best performing configurations for the engineering team, and the best performing configuration for the marketing team. These four configurations were evaluated on all 75 questions in the validation set. Despite these optimizations, the improvements were not as significant as hoped, emphasizing the need for further experimentation.

On the non-technical side, I learned that using a RAG system for document retrieval and question answering is viable. However, significant improvements are necessary before deploying the RAG system

into production since the engineering and marketing teams will be making important decisions based on the system's outputs.

Challenges and Limitations:

Throughout this project, I encountered several challenges and limitations. The most significant limitation was the limited amount of computational resources, which severely restricted the scope of experimentation and testing. Another limitation was the diversity and quality of the data in the document stores. The data was well suited for the engineering team since most of the data was very technical but not as suited for the marketing team. Another challenge was the varying performance across the different evaluation metrics. This emphasized the difficulty of effectively capturing the performance of response generation. A potential surprise to keep in mind for future iterations is the model's sensitivity to prompt design. The performance of the responses decreased when utilizing a poorly structured prompt, so it is of utmost importance to cater the prompt for the specific use case.

Next Steps

If I had more time, I would conduct experiments that utilized LLMs other than just Mistral-7B-Instruct-v0.2 and Cohere's proprietary model. I would also try increasing the size of the vector store to include a more diverse set of documents that were suited for, both, the engineering and marketing teams. I would also conduct fine-tuning of the LLM with NLP and Gen AI-related text data, which would require more computational resources. I would also conduct a comprehensive grid search that explores additional combinations of hyperparameters to identify the most effective configuration for each user group. Here are some questions that I identified as important to answer through my analysis:

1. Will fine-tuning the LLM significantly improve the quality of the generated responses?
2. Will having a more comprehensive, diverse document store provide better responses for both teams?
3. What is the optimal combination of hyperparameters to provide the best performing responses for both teams?

Summary & Recommendations

The POC demonstrated the potential of utilizing RAG systems to enhance our company's question answering capabilities. While the results of our best performing configurations only showed marginal improvement when compared to the baseline, it is clear that with further experimentation and testing there is potential for significant improvement. Therefore, I recommend further development and testing of the RAG system. Future work should include the following: experimenting with more LLMs, increasing the size and diversity of the vector store, fine-tuning the LLMs with domain-specific data, further prompt engineering, and performing a grid search of additional combinations of hyperparameters. It is important to consider architectural/deployment considerations such as scalability and maintenance. Additionally, it is crucial that the RAG system is able to be integrated with existing company systems. Estimating average and peak loads would also help give further insight into architectural considerations to ensure that the system is ready to handle varying amounts of usage. While the POC has shown great promise for improving our company's question answering capabilities, it is important that the RAG systems are properly and thoroughly evaluated before the engineering and marketing teams utilize it for making decisions.

Appendix:

RAG Configuration Results

Configuration	LLM	Embedding Model	Chunk Size	Chunk Overlap	Prompt	Temperature	Cosine Similarity	BLEU Score	ROUGE-2 Score
Baseline Engineering	Mistral	multi-qa-mpnet-base-dot-v1	128	0	Baseline	0.6	0.6943	0.0483	0.0825
Best Engineering	Cohere	multi-qa-mpnet-base-dot-v1	128	25	Improved	0.6	0.8016	0.0381	0.0897
2nd Best Engineering	Mistral	all-distilroberta-v1	128	25	Improved	0.6	0.7643	0.0349	0.0688
Baseline Marketing	Mistral	multi-qa-mpnet-base-dot-v1	128	0	Baseline	0.6	0.7064	0.0366	0.0767
Best Marketing	Mistral	all-distilroberta-v1	128	25	Improved	0.8	0.7302	0.0370	0.0673

***this table does not contain the results of the experiments but only the baseline and best performing configurations evaluated on all 75 questions**

Prompts:

Baseline Prompt

"""[INST]Please answer the question below only based on the context information provided.\n\nHere is a context:\n{context} \n\nHere is a question: \n{question}.[/INST]"""

Improved Prompts

Marketing Team:

"""[INST]
You are an expert assistant specializing in NLP and generative AI, geared to assist a marketing team at a tech firm planning to introduce a range of GenAI solutions.
This team lacks a technical background in NLP, requiring explanations free from complex jargon.
Please use only the context provided below to answer their questions:

{context}

That concludes the context section.

Now, respond to the following question based on the provided context in no more than 75 words. Aim to deliver your answer in a straightforward manner suitable for non-technical team members, focusing on general insights about GenAI technologies and their implications. Ensure your response is clear and concise, strictly avoiding lists or bullet points.

Here is the question:

{question}
[/INST]

Assistant:"""

Engineering Team:

"""[INST]
You are a dedicated assistant versed in NLP and generative AI technologies. Your audience consists of a group of engineers at a technology company eager to develop innovative GenAI applications.
These engineers possess a strong technical foundation in NLP, so they require answers with sufficient technical depth.
Please rely solely on the context provided below for information:

{context}

That concludes the context section.

Next, please provide an answer to the following question, drawing exclusively from the provided context. Your response should not exceed 125 words and must cater to engineers who expect a nuanced and detailed understanding of NLP concepts. Ensure your answer is precise and detailed, avoiding any lists or bullet points.

Here is the question:

{question}
[/INST]

Assistant:"""