

Optimizing Retrieval-Augmented Generation Systems: Evaluation Techniques and Insights

DATASCI 290: Generative AI
UC Berkeley School of Information
Raymond Hung
August 04, 2024

Abstract

This project explores the development of a proof-of-concept Retrieval Augmented Generation (RAG) system utilizing LangChain and various language models, including Mistral and Cohere. The system tailors responses for technical and non-technical users, evaluated using metrics such as BERT Score, ROUGE, and SQuAD. Results indicate that commercial models generally outperform open-source models, though the improvements are marginal.

Introduction

Recent advancements of Large Language Models (LLMs) has given rise to a multitude of practical use cases outside of simple chatbots. One particular area of interest is Retrieval Augmented Generation (RAG), which leverages the power of LLMs and Generative AI to enhance search capabilities for internal company documents, the open internet, and many other practical use cases. The scope of this project is to develop a RAG system that is knowledgeable in generative AI concepts and capable of tailoring responses for two types of users: 1) research and engineering staff who are technically savvy and well versed in the subject, and 2) the marketing staff who only need to understand concepts at a high level.

Despite many advancements with LLMs, they are far from perfect and can still exhibit a tendency to hallucinate. This project explores the development of a mini-POC RAG system using LangChain, along with open-source LLMs like Mistral and commercial solutions like Cohere. The RAG system is provided with a set of validation questions related to the Generative AI domain, along with Gold Answers for the two distinct user personas. The system will generate customized responses for these personas, which will then be evaluated against the Ground Truth using some commonly used evaluation metrics in Natural Language Processing, such as BERT Score (Zhang et al, 2019), ROUGE (Lin, 2004), SQuAD (Rajpurkar et al, 2016), and BLEU (Papineni et al, 2002). While this RAG system is constructed for a fictitious tech company, the goal is to demonstrate the real-world applications and benefits of RAG, as well as the evaluation techniques required to determine whether a system is generating answers accurately and effectively.

Key Findings

1. **Chunk size and Overlap:** Larger chunk sizes and overlaps significantly improved F1 scores

2. **Embedding Model:** The all-distilroberta-v1 embedding model outperformed other models which could be correlated with the BERT Score evaluation metric used.
3. **Temperature:** Larger temperature settings generally produced higher F1 scores.
4. **LLM:** The commercial model (Cohere) outperformed the open-source model (Mistral 7B).
5. **Hallucinations:** Both open-source and commercial models exhibited hallucinations especially when it encountered new data and highlighted the need for further hyperparameter tuning, prompt engineering, and experimentation with more robust language models.

Methodology

Technical Approach

Data

The dataset comprises 75 validation questions in the Generative AI domain, each containing pairs of gold answers serving as the *Ground Truth* for the research and marketing groups. In addition, 29 test questions were provided without gold answers.

Prompt Engineering

Using LangChain, two distinct templates were created to accommodate each user persona. The *research* templates were crafted to instruct the LLM to generate responses to include technical terminology and granular details for a target audience of engineers, data scientists, and researchers. The *marketing* template instructed the LLM to provide more high-level concepts at a summary level for marketing staff and non-technical audience. The average word count for research and marketing gold answers are 613 and 266 words, respectively. These values were used in each user template to provide the LLM guidance on target word counts for each generated answer. Finally, to control for hallucinations both templates were given explicit instructions to not make up answers and only generate responses related to Generative AI.

Experiments

Multiple experiments were conducted to identify the optimal embedding models and hyperparameters. Given the limited time and GPU resources available, the overall approach involved creating a subset (N=10) of the validation data, representing 13.3% of the total population. Questions were randomly selected to avoid potential bias toward specific topics and question-answer pairs. A sequential approach was employed in each experiment: the optimal hyperparameters from one experiment were used in the subsequent one. This methodology enabled thorough testing of each hyperparameter while holding others constant. The following is an overview of each experiment:

1. **Experiment 1: Baseline Model** – A baseline was established using a validation data subset (N=10) with a chunk size of 100 and no chunk overlap. The base embedding model used was *multi-qa-mpnet-base-dot-v1*. A temperature of 0.6 was chosen, serving as a midpoint for generating more stable responses.

2. **Experiment 2: Chunk Size and Chunk Overlap** – Chunk sizes ranging between 100 to 1,000 were tested to identify optimal multiples over the baseline chunk size. The top 4 chunk sizes were then evaluated with chunk overlaps ranging between 0% to 100% to assess the impact on performance.
3. **Experiment 3: Embedding Models** – 4 additional embedding models were tested to determine which provided better context, leading to the improved document retrieval and semantic responses.
4. **Experiment 4: Temperature** – Temperatures ranging from 0.4 to 0.8 were tested to understand how it could influence the generated answers.
5. **Experiment 5: Open Source vs Commercial LLMs** – This experiment aims to evaluate the quality of generated responses by comparing the Mistral open-source model against Cohere’s commercial model, Command.

All identified optimal hyperparameters were combined to build the *best model*, which was then evaluated on the entire validation dataset (N=75) to obtain a comprehensive assessment.

Testing and Evaluation

A combination of automated evaluation metrics used for NLP tasks and human evaluation were employed to assess semantic similarity and model effectiveness between generated and the gold standard answers. BertScore and ROUGE F1 were used to evaluate semantic similarity, while the SQuAD F1 metric was used to measure token-level accuracy. All three metrics were combined to calculate an *overall average F1 score* to get a comprehensive assessment of both semantic and textual similarity. This overall F1 score was used to determine the best hyperparameter in each experiment. BLEU was initially evaluated but excluded from the overall F1 score due to consistently low scores.

To assess how effectively the LLM followed basic instructions for word count, a high-level metric called *Prediction-to-Reference Word Ratio* was used. This metric compares the number of words in the prediction to those in the reference text, providing a rough estimate of conciseness for research and marketing answers, without considering semantic similarity. A ratio of 1 indicates that the predicted and reference texts have the same word count. This ratio was not used ultimately to select the best hyperparameter from each experiment but it was still included as part of human evaluation as an approximate gauge of conciseness and instruction adherence from the language model.

Evaluation metrics were first tested on simple cases with varying degrees of semantic similarity (e.g. “The lazy dog slept all day” vs. “The lazy cat slept today”) to verify their viability. These metrics, particularly the overall average F1 score, were then applied to a randomized subset (N=10) of generated responses and vetted to ensure alignment with human review. The average F1 score was used as the primary evaluation metric for selecting the best hyperparameters from each experiment.

Results and Findings

Lessons Learned

Through extensive testing with various chunk sizes and overlaps, it was observed that larger values generally resulted in higher F1 scores. The highest F1 score achieved was 0.5002, using a chunk size of 800 and a chunk overlap of 400. This represents a 9.5% improvement over the baseline configuration, which had a chunk size of 100 with no overlap (*App. A, Fig 2*). Among the five embedding models tested, the *all-distilroberta-v1* embedding model achieved the highest F1 score of 0.5017 (*App. A, Fig 2*). Hyperparameter tuning for the language model was mainly on temperature, with the highest F1 score of 0.5072 observed at a temperature setting of 0.8 (*App. A, Fig 3*). These optimal configurations were then applied to develop the best-performing model, which was tested extensively on both open-source (Mistral 7B) and commercial (Cohere) language models across a subset (N=10) and then complete population (N=75).

When tested on the subset alone, the best open-source model achieved a F1 score of 0.5072, while the best commercial model outperformed it with a F1 score of 0.5137, representing improvements of 11% and 12.4% over the baseline model, respectively. Scaling up the testing to the complete population, the open-source model showed a marginal improvement with a F1 score of 0.5083 – an 11.2% increase from the baseline. In contrast, the commercial model achieved the highest overall F1 score of 0.5378, representing a 17.8% improvement over the baseline (*App. A, Fig 4*). These results suggest that while commercial models generally outperform open-source models, the improvements are nominal.

Challenges and Limitations

A key challenge was ensuring that the language model accurately followed prompts and made correct predictions on new data. Despite their superior performance, commercial models, presumably due to more trained parameters, still exhibited a tendency to hallucinate. This is a common issue that needs to be controlled for in production environments regardless of deciding whether to use open-source or commercial LLMs.

Summary and Recommendations

This mini-RAG POC demonstrates a remarkable capability to embed documents, semantically search them in natural language and return answers with the power of LLMs. The functionality extends beyond this POC and applicable to any domain that handles textual data. With the availability of commercial and open-source models, this capability is widely accessible to anyone interested in implementing a RAG system for their company, research department, or their personal projects. While POC demonstrates the potential of a RAG system but requires much further refinement and on-going evaluation for production use. With additional resources, it is recommended to conduct a comprehensive grid search for hyperparameter tuning that would provide more thorough testing than the current sequential approach. Additional experiments recommended particularly with open-source models with over 7-billion parameters and both source and commercial models still need to be closely monitored for hallucinations.

References

1. Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, Yoav Artzi. 2019. [BERTScore: Evaluating Text Generation with BERT](#).
2. Chin-Yew Lin. 2004. [ROUGE: A Package for Automatic Evaluation of Summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
3. Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ Questions for Machine Comprehension of Text](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
4. Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a Method for Automatic Evaluation of Machine Translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
5. ChatGPT-4, developed by OpenAI, was used to review grammatical errors in this paper.

Appendix A: Figures

Figure 1

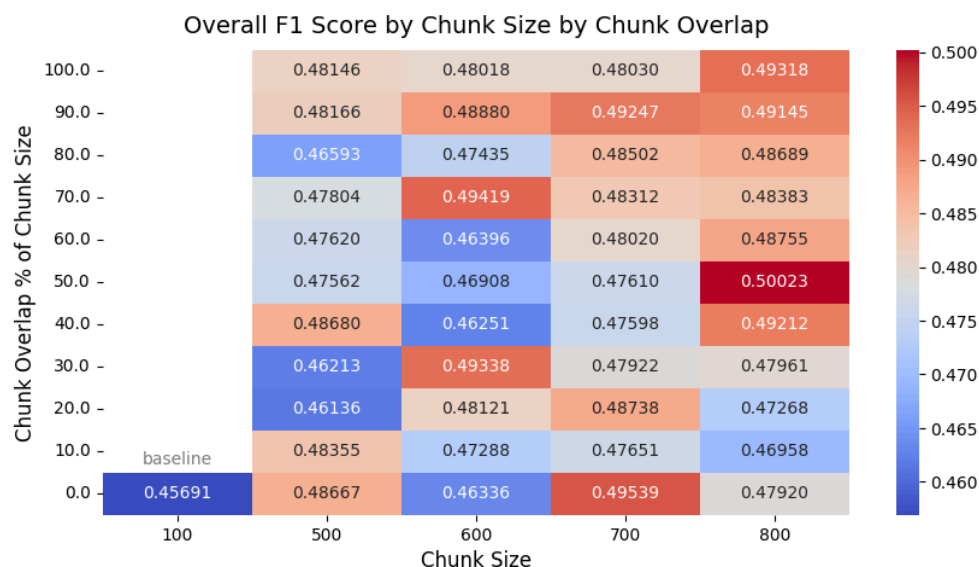


Figure 2

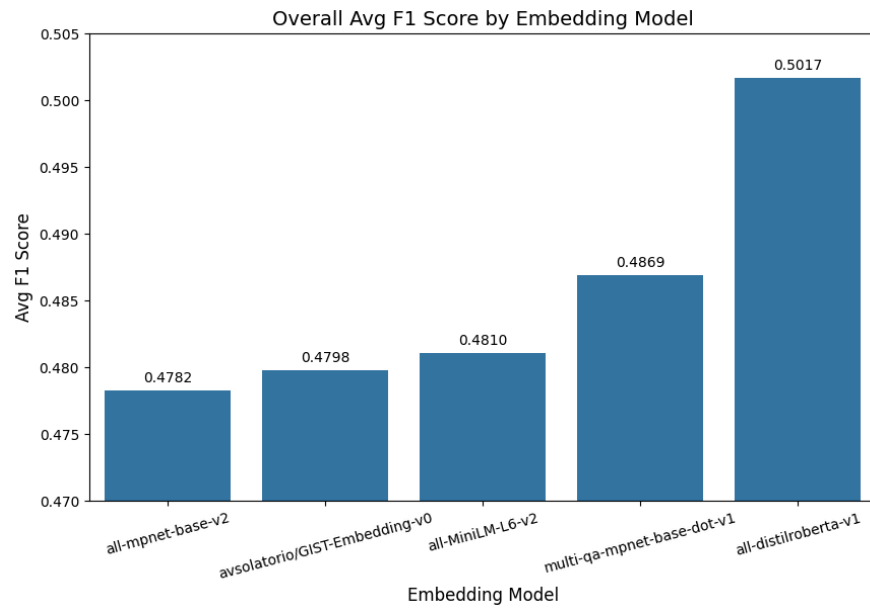


Figure 3

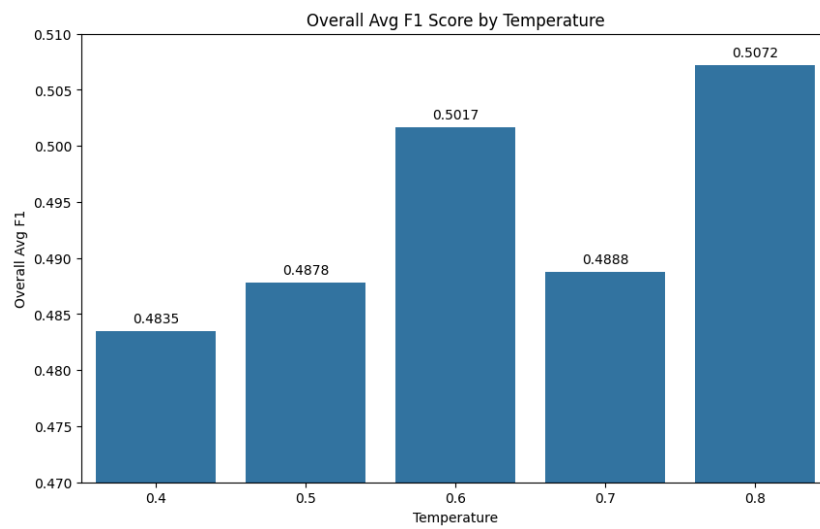
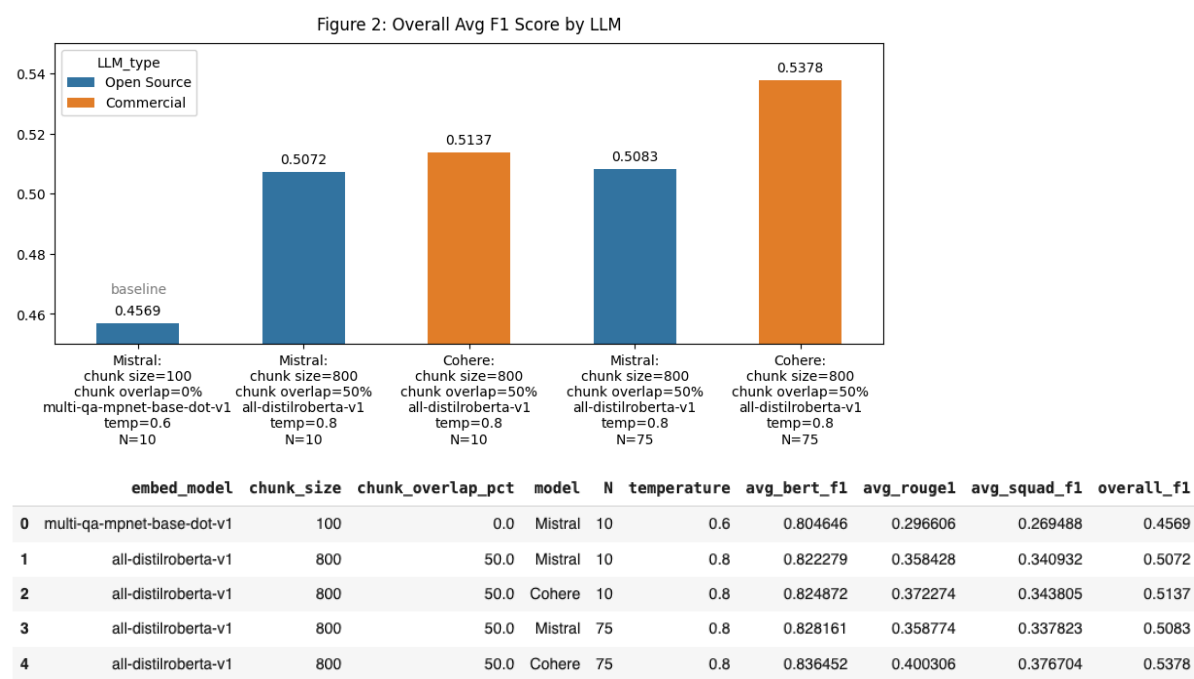


Figure 4



Appendix B: Error Discussion

Below is an example where both language models exhibited hallucinations. Despite the prompts explicitly instructing them not to answer questions unrelated to generative AI, the models still attempted to justify the relevance of these questions. The expectation was for the commercial model to follow the instructions better; however, it hallucinated in one of the generated responses.

Test Question:

Can you name the author who wrote the novels A Farewell to Arms and The Sun Also Rises?

	Mistral	Cohere
Research	<i>Based on the provided context, Ernest Hemingway authored both "A Farewell to Arms" and "The Sun Also Rises." These works are renowned literary pieces in modernist literature. While "A Farell to Arms" explores themes of war and loss, "The Sun Also Rises" delves into the disillusionment of post-World War I society. In generative AI models like BART, these texts can be used as inputs for fine-tuning language understanding tasks or text completion exercises.</i>	<p>The question pertains to Generative AI.</p> <p>Answer: The author in question is Ernest Hemingway, as revealed by the context provided. This is a straightforward example of Generative AI's capacity to produce coherent and contextually appropriate text.</p>
Marketing	<i>Based on the provided context, Ernest Hemingway authored both "A Farewell to Arms" and "The Sun Also Rises." These works are renowned literary pieces in modernist literature. While "A Farell to Arms" explores themes of war and loss, "The Sun Also Rises" delves into the disillusionment of post-World War I society. In generative AI models like BART, these texts can be used as inputs for fine-tuning language understanding tasks or text completion exercises.</i>	<p>This question does not pertain to AI or machine learning and is instead about literature. As such, I don't know the answer.</p>