Benchmarking a RAG Pipeline

Carson Crockett

I. Introduction

This document is to brainstorm possible approaches to use in the future to benchmark our RAG pipeline. These approaches are meant to be used on a pipeline built to deconstruct and analyze COBOL code.

II. A Second Model

The first idea is to have the RAG pipeline generate answers to a set of queries. We would experiment with the volume of these queries. We take these answers and provide them to another model and instruct it to grade the answers on their relevance to their relative questions. This approach is concerning because we would then need to validate the benchmark for its accuracy in determining relevance.

III. Regular Intelligence

Our next idea is similar to the first but instead of having the model evaluate answers based on relevance to the question, we instead generate our own set of answers. We then feed both the question, generated answer, and actual answer to another model. We then instruct the model to compare the two answers to each other and grade the generated answer based on similarity to the answer we give it.

IV. RAGAS

RAGAS is a framework developed to test the faithfulness of a RAG system's responses to the context it was provided. "RAGAS uses LLM's to generate statements from a question answer pair and compute a faithfulness score based on how many statements are supported by the given context." The problem with using other LLM's as mentioned above is that their scope is too wide and they cannot accurately detect hallucinations and deviations from the context material.

a. Lynx

Lynx is a fine tuned model using LLaMa-3-70B to focus on detecting hallucinations. It uses example question answer pairs with answers purposely inserted that are correct but deviate from the provided context. This allows Lynx to recognize hallucinations closed-source LLM's don't see.

https://arxiv.org/abs/2407.08488v1
https://slides.com/sasatrivic/ragas#/5/0/0

V. NVIDIA Open-Source tools

We have been in contact with the Capstone team working with NVIDIA. They are also exploring RAG with the NVIDIA API. We don't have access to this API but they were able to direct us to some useful open source tools we can use to benchmark our model. The referenced us to the following github links for NeMo SDG;

- https://github.com/NVIDIA/GenerativeAIExamples/tree/main/nemo/retriever-synthetic-data-generation
- https://github.com/NVIDIA/NeMo-Curator/blob/main/tutorials/synthetic-retrieval-evaluation/SDG-Retriever_Eval_Tutorial.ipynb .

NeMo SDG is a tool to create a dataset of question answer pairs we can “quiz” the model on to evaluate its performance. NeMo SDG can “Quickly generate complex QA datasets from existing text documents for retriever model evaluation”.

VI. Summary

To test a RAG model the best approach is to query the model with a set of question and answer pairs with the context it needs to answer them. The model should be graded on the correctness of its answer as well as the faithfulness of the answer to the context provided and whether or not the answer was the result of a hallucination. We can use NeMo SDG to generate a QA dataset to use against the model. We can also use another LLM such as Lynx to detect hallucinations within its responses as well as manually grading the model. Once the model is ready to be tested we can experiment with these ideas to see what we can get running and how useful the evaluation is. At the moment this is just a starting point for the concepts we will need to put in place, when necessary we will develop an implementation plan for benchmarking.