End-to-End Evaluation of Generative AI Applications

Capstone Assignment III

The learning objective of this assignment is to build your knowledge on end-to-end evaluation of a generative AI applications. While, the focus of this assignment is on a Retrieval Augmented Generation solution, the insights that you develop, and processes that you follow will be largely applicable to other generative AI solutions that you will encounter in the future.

Overview

There are three application elements that you will need to install and run for assignment. Each of which is run on the command line of a terminal/shell:

Capstone RAG: This is a Python application that closely follows the RAG implementation of Capstone Assignment II.

RAG Test Driver: This is Python application that makes it easier to experiment with model and prompt template options for the RAG application.

Promptfoo: An Open Source Generative AI test framework.

Task 1

Capstone RAG & RAG Test Driver: Install and system test these applications. Do this by following the instructions in the document ReadMe_to_install_capstone_rag.md.

Task 2

Promptfoo: Install and system test promptfoo. Do this by following the instructions in the document ReadMe_to_install_promptfoo.md.

Task 3

End-to-end System Test: This will test the three components working together. Complete task 1 and 2 before doing this. The instructions for doing this are in the document ReadMe_to_install_promptfoo.md.

Task 4

Update and add to the end-to-end tests: There are 3 example tests provided in the promptfoo_config.yaml file that drives the automated testing. Add to 7 or more tests to this file and update the 3 initial tests as you desire. Each test will run with the three different LLMs that are configured, and their corresponding prompt templates. The document corpus is three large PDFs on US taxes. Take a look at those to get inspiration for the questions that you will use in your test cases, and the evaluation of the responses. All the changes that you make during this task will be to the promptfoo_config.yaml file.

Complete this task and write-up your findings. Submit your updated promptfoo_config.yaml, and your summary to complete the assignment .

Task 5 (Optional)

Change the set of LLMs used during testing: The mapping of the different LLM options (one, two, three) to Amazon Bedrock model identifiers used by the capstone_rag application occurs in the rag_test_driver.py file. Update the set of model-ids to match the LLMs that you are most interested in and enable those models in the Amazon Bedrock Console. Then re-run your tests.

To complete this task, write-up your findings and submit this summary along with your summary from task 4.

Task 6 (Optional)

Change the set of RAG prompt templates used during testing: The mapping of the different RAG prompt template options (one, two, three) to actual templates used by the RAG application occurs in the rag_test_driver.py file. Update the set of prompt templates to ones that better suited to the models that will use them. Then re-run your tests and tune the prompts some more.

To complete this task, write-up your findings and submit this summary along with your summary from task 4.

Task 7 (Optional)

Update the dataset that is used by capstone_rag to be you dataset from Capstone II: This requires updating capstone_rag.py to use your dataset. After

updating the datasources to be loaded, delete the cached embedding to get the application to process your new data. You may also choose to update capstone_rag at this time to reflect your learnings from Capstone II.

Once the updates to capstone_rag.py and system testing is completed, create a set of 5 or more Promptfoo tests to run with your dataset.

To complete this task, write-up your findings and submit this summary along with your summary from task 4.