RAG Evaluation

This report presents the evaluation of a Retrieval-Augmented Generation (RAG) system using an open-source LLM to assess both its generation and retrieval performance.

Core Functionality

The script works by following a clear, step-by-step process for each question it's given:

1. **Gets the Question:** It starts by loading a list of questions and their "ground truth" (ideal) answers from a JSON file.
2. **Retrieves Context:** For each question, it generates a numerical representation (an embedding) and uses it to search a **ChromaDB** vector database for the most relevant pieces of information, called "context chunks."
3. **Generates an Answer:** It then passes the retrieved context and the original question to a large language model (**LLM**), instructing it to formulate an answer based only on the information provided.

The "LLM-as-a-Judge"

The most notable feature is its sophisticated evaluation method. Instead of simple checks, the script uses another LLM to act as an impartial **"judge."** This judge assesses the generated answer against the ground truth based on four key metrics:

- **Faithfulness:** Is the answer true to the provided context?
- **Answer Relevancy:** Does the answer actually address the user's question?
- **Context Precision:** Was the retrieved information concise and on-topic?
- **Context Recall:** Did the retrieved information contain everything needed to give the perfect answer?

Final Output and Reporting

After evaluating all the questions, the script does two things:

1. It prints a **final summary** to the console, showing the average score (out of 5) for each of the four metrics. This gives a quick overview of the system's strengths and weaknesses.
2. It saves a detailed log of the entire process, including the question, retrieved context, generated answer, and the judge's full evaluation, into an `evaluation_results.json` file for in-depth analysis.

-

```
============== Summary ==============
Average Faithfulness: 4.67 / 5  (6/6 evaluated)
Average Answer Relevancy: 4.17 / 5  (6/6 evaluated)
Average Context Precision: 4.00 / 5  (6/6 evaluated)
Average Context Recall: 4.33 / 5  (6/6 evaluated)
```