Evaluation Assistant: Helping AI Students Master LLM Evaluation

Introduction

You've heard it before, all RAG applications rely on putting the right stuff in context. This one's personal.

Background and Context

When I started learning AI engineering, one topic gave me more trouble than anything else: evaluation. I could build cool stuff with RAG, LangChain, and embedding models, but I didn't know how to tell if my work was actually good.

What is faithfulness, and how is it different from factual correctness? When should I even evaluate a RAG pipeline? I would get lost in a sea of metrics and blog posts, and nothing stuck.

So I decided to fix the problem the best way I know how, by building a tool that could help me and others like me. Introducing Evaluation Assistant, a smart helper that explains evaluation concepts clearly and helps AI students figure out what they are doing.

Task 1: Defining My Problem and Audience

I built an Evaluation Assistant to solve my own problem, but it can't be just me. There must be other students that have wrestled with evaluation in one way or another.

We are handed terms like context recall and hallucination rate, but we are rarely shown how to connect them to real workflows.

The audience for this tool is students like me who are building RAG applications and want to do it right, but need help figuring out how and when to evaluate their models effectively.

Task 2: My Solution

The solution is a chatbot I built called Evaluation Assistant. It acts like a tutor and a study buddy rolled into one. You can ask it a question like 'What is faithfulness?' or 'Should I run evals before or after fine-tuning?' and it gives you helpful, grounded answers.

It also pulls from trusted sources like arXiv and research blogs, so I know the information is solid. It is not just a homework shortcut. It is a tool to actually learn how to think about evaluation.

Here is the stack I used to build Evaluation Assistant:
- LLM: gpt-4o-mini
- Embeddings: text-embedding-3-small and snowflake-arctic-embed-l
- Orchestration: LangGraph
- Vector Store: I ended up using the AIMakerspace helpers here
- Monitoring: LangSmith
- Evaluation: RAGAS
- UI: Chainlit
- Inference and Hosting: Hugging Face

This stack gave me flexibility to iterate quickly, while also showing that open-source tools can deliver production-level performance.

Task 3: Getting the Data

I gathered 3 papers on the topic:
- Re-evaluating Automatic LLM System Ranking for Alignment with Human Preference

- LLM Evaluation: Metrics, Methodologies, Best Practices | DataCamp

-[2310.19736] Evaluating Large Language Models: A Comprehensive Survey

Then I hooked up the ArXiv tool and Tavily Search so I could pull in more sources when needed. Chunking is just standard recursive splitting for now, but I will refine it as I see where the Assistant struggles.

Task 4: Building the Prototype

I used LangGraph to wire everything together, AIMakerspace helpers for the vector store, and Chainlit for the UI. It is all hosted on Hugging Face so I can show it off easily. I am pretty proud of how fast I got the prototype up and running. It is already helping me debug my own thinking, and that was the goal.

Task 5: Golden Test Data Set

I made a set of evaluation questions.

https://app.ragas.io/dashboard/alignment/testset/b0009f7a-3106-42fa-862b-4897d56d5cea

I used those to baseline the first version of the Assistant with RAGAS. Here is how it performed:

- Context Recall: 0.7000
- Faithfulness: 0.8305

- Factual Correctness: 0.3540
- Answer Relevancy: 0.4684
- Context Entity Recall: 0.0875
- Noise Sensitivity Relevant: 0.2366

Task 6: Fine-Tuning the Embeddings

I fine-tuned an open-source embedding model using synthetic question and answer pairs based on my golden test set. I published the model to Hugging Face for transparency and reuse.

Task 7: Evaluating the Results

After fine-tuning, the results improved across the board:

- Context Recall: 0.9833
- Faithfulness: 0.8269
- Factual Correctness: 0.4408
- Answer Relevancy: 0.6212
- Context Entity Recall: 0.1927
- Noise Sensitivity Relevant: 0.2293

The boost in faithfulness and factual correctness was especially exciting. It means the Assistant is doing a better job of staying grounded in real, reliable information.

Final Submission

1. GitHub Repo: coming soon
2. App: chelleboyer/cert-challenge at main
3. Fine-tuned Embeddings: chelleboyer/llm-evals-2-79b954ef-4798-4994-be72-a88d46b8ecca · Hugging Face
4. Time Invested: 40 hours