Smart Python Assistant using LangGraph, RAG, and FastAPI

Doaa Said AbdElazim

Project Overview

- Built an intelligent Python assistant.
- Detects user intent: **generate code** or **explain code**.
- Uses LangGraph state machine for clean workflow.
- Integrated Retrieval-Augmented Generation (RAG).
- Deployed using FastAPI.

Project Objectives

- Build a modular AI assistant.
- Use sentence embeddings for code retrieval.
- Implement LangGraph for multi-step logic.
- Connect to HuggingFace LLM.
- Create a deployable API endpoint.

System Architecture

- Input $\rightarrow$ Intent Classifier
- RAG Retrieval (Chroma + MiniLM)
- LLM Response Generator
- LangGraph State Machine
- FastAPI service endpoint

LangGraph State Machine

- Nodes:
  - chat_state $\rightarrow$ classify intent
  - llm_node $\rightarrow$ generate/explain
  - final_node $\rightarrow$ return output
- Edges define clean conditional flow.

Intent Classification

- Keyword-based:
  - “explain”, “what does” $\rightarrow$ explain mode
  - Otherwise $\rightarrow$ generate mode
- Output stored as: `state["intent"]`.

Dataset for Retrieval

- Used HumanEval Dataset.
- Stored in Chroma vector DB.
- Each example:
  - Problem description
  - Canonical solution

RAG Retrieval Pipeline

- Embeddings: MiniLM L6-v2
- Vector DB: Chroma
- Retrieve the closest example:

```
vector_store.similarity_search(user_query)
```

LLM Interaction

- Custom HuggingFace LLM class.
- Two prompt templates:
  - Code generation
  - Code explanation
- Uses local Zephyr-7B model.

Prompt Design

- Generate Code:
  - Uses retrieved example as guidance.
- Explain Code:
  - Breakdown in simple steps.

FastAPI Deployment

- API endpoint:

  `/run`

- Sends input $\rightarrow$ runs LangGraph $\rightarrow$ returns output.
- Runs as background server thread in notebook.

Results

- Assistant correctly intent.
- Generates helpful Python code.
- Retrieves relevant examples for RAG.
- Answers explanation questions clearly.

Conclusion

- Successfully built an end-to-end intelligent assistant.
- Combines LangGraph, RAG, HF LLM, and FastAPI.
- Modular, scalable and deployable.