LangGraph Chatbot Documentation

Model Used: meta-llama/Meta-Llama-3-8B-Instruct (HuggingFace)

1. Introduction

This project demonstrates the development of a stateful AI chatbot using LangGraph integrated with the HuggingFace model 'meta-llama/Meta-Llama-3-8B-Instruct'. The chatbot processes user input, maintains conversation history, and generates intelligent responses.

2. System Architecture

- User Input is received.
- State object stores conversation history.
- LangGraph routes input to LLM node.
- Llama 3 model generates response.
- Updated state is returned to user.

3. Installation Requirements

Required Libraries: - langgraph - langchain - transformers - torch - huggingface_hub

4. Implementation Overview

Step 1: Define Chat State using TypedDict or Pydantic. Step 2: Load HuggingFace Llama 3 model using transformers pipeline. Step 3: Create LangGraph nodes for response generation. Step 4: Define edges and compile StateGraph. Step 5: Execute graph using invoke() or stream().

5. Core Code Structure Explanation

The chatbot node sends conversation messages to the Llama 3 model. The model generates text using instruction tuning. LangGraph updates the shared state and returns the response.

6. Features

- Stateful multi-turn conversation
- HuggingFace Llama 3 integration
- Scalable graph-based workflow
- Supports tool integration
- Production-ready architecture

7. Future Enhancements

- Add Retrieval-Augmented Generation (RAG) - Integrate Vector Databases (FAISS/Chroma) - Add Persistent Memory (Redis/Database) - Deploy with FastAPI or Streamlit