

Deep Research AI Agentic System

Overview

This system implements a dual-agent research pipeline using Tavily, LangChain, and LangGraph. The Research Agent gathers online data, and the Answer Drafter Agent synthesizes a coherent response.

Agent Architecture

1. Research Agent:

- Uses Tavily to perform web searches based on the query.
- Extracts and loads content from top URLs.
- Splits content into chunks and embeds it into a vector store.

2. Answer Drafter Agent:

- Loads the vector store.
- Uses LangChain's RetrievalQA to answer questions using the stored research.

LangGraph Flow

The system uses LangGraph to coordinate agents:

- Input Node: Accepts the user query.
- Research Node: Calls the Research Agent.
- Answer Node: Calls the Answer Drafter Agent.
- Final Output: Presents the generated response.

Technologies Used

- Tavily API for live web crawling
- LangChain for chaining tools and LLMs
- LangGraph for flow orchestration

- FAISS for vector storage
- OpenAI (ChatGPT) for embeddings and final answers

Usage Instructions

1. Install dependencies: ``pip install -r requirements.txt``
2. Add your API keys to the ``.env`` file.
3. Run ``main.py`` to execute both agents.
4. Use ``langgraph_flow.py`` to run the orchestrated agent flow.

Credits & Author

This system was generated using AutoGPT Agent.

Visit <https://agilayer.com> for more autonomous agent workflows.