Deep Research AI Agentic System Implementation Guide

Overview

This project implements a **dual-agent AI system** for deep research using **Tavily for web crawling** and **LangGraph & LangChain for organizing information**. The system has:

1. A **Research Agent** that gathers information from the web.
2. An **Answer Drafting Agent** that processes the gathered data and generates a response.

Technologies Used

- **LangGraph & LangChain**: To structure the agent workflow.
- **Tavily Search API**: For online information retrieval.
- **OpenAI API**: To generate responses from gathered data.

System Design

1. **Research Agent**
    - Uses the **Tavily API** to collect online data.
    - Stores retrieved results for further processing.
2. **Answer Agent**
    - Uses **GPT-4 –o-mini (OpenAI)** to analyze collected data.
    - Generates a well-structured, detailed response.

Implementation Steps

1. Setup Environment

Ensure you have Python installed and run:

pip install langchain langgraph langchain-community tavily-api openai

2. Configure API Keys

Set environment variables:

```
export TAVILY_API_KEY="your_tavily_api_key"
export OPENAI_API_KEY="your_openai_api_key"
```

3. Run the System

Save the Python script and execute:

```
python ai_research_agent.py
```

Expected Output

Given a query, the system:

1. Fetches web-based information.
2. Processes and summarizes the content.
3. Outputs a **final answer** in a structured format.