Travel Planning Agentic System using LLMchain

This project is a simulated agentic workflow using LLMChain:

LLMChain = deterministic task executor (1 prompt -> 1 response)

Use SequentialChain to coordinate the "agents"

Each agent = 1 chain with its own prompt

For agents like "Itinerary Planner", where no tools are needed and you just want a clear response, we do not require to use agents at all. We can achieve the same thing with LLMChain. In this the agents were not explicitly defined as standalone objects. Instead, LLMChain instances are used to perform the roles of agents.

Langchain has two levels of abstraction of AI_driven workflows:

1. LLMChain (these are not full "agents" - they are deterministic chains of prompt + LLM). Suitable for sequential tasks. Not decision-making, tool selection, or memory
2. LangChain Agents (using this, we will define true agents. initialize_agents, AgentExecutor+tools, Tool use + memory + reasoning)

1.Setup & Environment

```
!pip install langchain langchain_openai langchain_community python-dotenv
```

```
Requirement already satisfied: openai<2.0.0,>=1.99.9 in /usr/local/lib/python3.12/dist-packages (fr
Requirement already satisfied: tiktoken<1,>=0.7 in /usr/local/lib/python3.12/dist-packages (from la
Requirement already satisfied: aiohttp<4.0.0,>=3.8.3 in /usr/local/lib/python3.12/dist-packages (fr
Requirement already satisfied: tenacity!=8.4.0,<10,>=8.1.0 in /usr/local/lib/python3.12/dist-packag
Collecting dataclasses-json<0.7,>=0.5.7 (from langchain_community)
  Downloading dataclasses_json-0.6.7-py3-none-any.whl.metadata (25 kB)
Requirement already satisfied: pydantic-settings<3.0.0,>=2.4.0 in /usr/local/lib/python3.12/dist-pa
Requirement already satisfied: httpx-sse<1.0.0,>=0.4.0 in /usr/local/lib/python3.12/dist-packages (
Requirement already satisfied: numpy>=1.26.2 in /usr/local/lib/python3.12/dist-packages (from lang
Requirement already satisfied: aiohappyeyeballs>=2.5.0 in /usr/local/lib/python3.12/dist-packages (
Requirement already satisfied: aiosignal>=1.4.0 in /usr/local/lib/python3.12/dist-packages (from ai
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.12/dist-packages (from aioh
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.12/dist-packages (from a
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.12/dist-packages (from
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.12/dist-packages (from a
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.12/dist-packages (from a
Collecting marshmallow<4.0.0,>=3.18.0 (from dataclasses-json<0.7,>=0.5.7->langchain_community)
  Downloading marshmallow-3.26.1-py3-none-any.whl.metadata (7.3 kB)
Collecting typing-inspect<1,>=0.4.0 (from dataclasses-json<0.7,>=0.5.7->langchain_community)
  Downloading typing_inspect-0.9.0-py3-none-any.whl.metadata (1.5 kB)
```

```
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.12/dist-packages (fr
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from reques
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from
Requirement already satisfied: greenlet>=1 in /usr/local/lib/python3.12/dist-packages (from SQLAlch
Requirement already satisfied: regex>=2022.1.18 in /usr/local/lib/python3.12/dist-packages (from ti
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.12/dist-packages (from httpx
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.12/dist-packages (from httpcore=
Requirement already satisfied: jsonpointer>=1.9 in /usr/local/lib/python3.12/dist-packages (from js
Collecting mypy-extensions>=0.3.0 (from typing-inspect<1,>=0.4.0->dataclasses-json<0.7,>=0.5.7->lan
  Downloading mypy_extensions-1.1.0-py3-none-any.whl.metadata (1.1 kB)
Downloading langchain_openai-0.3.32-py3-none-any.whl (74 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 74.5/74.5 kB 5.1 MB/s eta 0:00:00
Downloading langchain_community-0.3.27-py3-none-any.whl (2.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.5/2.5 MB 78.1 MB/s eta 0:00:00
Downloading dataclasses_json-0.6.7-py3-none-any.whl (28 kB)
Downloading marshmallow-3.26.1-py3-none-any.whl (50 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 50.9/50.9 kB 3.7 MB/s eta 0:00:00
Downloading typing_inspect-0.9.0-py3-none-any.whl (8.8 kB)
Downloading mypy_extensions-1.1.0-py3-none-any.whl (5.0 kB)
Installing collected packages: mypy-extensions, marshmallow, typing-inspect, dataclasses-json, lang
Successfully installed dataclasses-json-0.6.7 langchain_community-0.3.27 langchain_openai-0.3.32 ma
```

```python
import os
from dotenv import load_dotenv
from langchain.chat_models import ChatOpenAI
from langchain.prompts import PromptTemplate
from langchain.chains import LLMChain, SequentialChain


# Load API Keys
load_dotenv()
openai_api_key = os.getenv("OPENAI_API_KEY")

if not openai_api_key:
    raise ValueError("Missing OpenAI or Tavily API keys")


# Initialize LLM
llm = ChatOpenAI(model="gpt-3.5-turbo", temperature=0.7, openai_api_key=openai_api_key)
```

2. Prompt Templates for each LLMChain Role

```python
# Destination Researcher
destination_prompt = PromptTemplate.from_template("""
Research comprehensive travel information about {destination}:
Include best seasons to visit, safety, etiquette, costs, language, transport options, etc.
Travel Dates: {travel_dates}, Budget: {budget}.
""")

# Attractions Specialist
attractions_prompt = PromptTemplate.from_template("""
List top attractions and experiences in {destination} for travelers who enjoy {preferences}.
Budget: {budget},
Trip Duration: {duration_days} days.
""")

# Itinerary Planner
itinerary_prompt = PromptTemplate.from_template("""
Created a detailed {duration_days}-day travel itinerary for {destination} starting {travel_dates}.
Make sure the itinerary reflects preferences like: {preferences}, and fits within a {budget} Budget.
""")

# Local Guide
guide_prompt = PromptTemplate.from_template("""
Provide local tips, cultural insights, and safety advice for visiting {destination}.
Tailor to interests: {preferences}.
Budget level: {budget}.
""")
```

3. Define LLMChain for each Role

```
destination_chain = LLMChain(llm=llm, prompt=destination_prompt, output_key="destination_report")
attractions_chain = LLMChain(llm=llm, prompt=attractions_prompt, output_key="attractions")
itinerary_chain = LLMChain(llm=llm, prompt=itinerary_prompt, output_key="itinerary")
guide_chain = LLMChain(llm=llm, prompt=guide_prompt, output_key="guide")
```

4. Take Input from User

```
destination = input("Destination: ")
travel_dates = input("Travel Dates (default June 1, 2023): ") or "June 1, 2023"
duration_days = int(input("Duration (default 3): ") or "3")
preferences = input("Preferences (default: Cultural & Museums, Food & Culinary): ") or "Cultural & Museu
budget = input("Budget (default: moderate): ") or "moderate"

input_variables = {
    "destination": destination,
    "travel_dates": travel_dates,
    "duration_days": duration_days,
    "preferences": preferences,
    "budget": budget
}
```

```
Destination: chicago
Travel Dates (default June 1, 2023): 3
Duration (default 3):
Preferences (default: Cultural & Museums, Food & Culinary):
Budget (default: moderate):
```

5. Use SequentialChain to Run All Chains in Order

```
travel_chain = SequentialChain(
    chains = [destination_chain, attractions_chain, itinerary_chain, guide_chain],
    input_variables=["destination", "travel_dates", "duration_days", "preferences", "budget"],
    output_variables=["destination_report", "attractions", "itinerary", "guide"],
    verbose=True
)

# Run the LLMChain-based agentic system
results = travel_chain.invoke(input_variables)
```

```
> Entering new SequentialChain chain...

> Finished chain.
```

6. Print the Final Travel Plan Output

```
print("\n--- Travel Plan ---")
print("Destination Overview:\n", results["destination_report"])
print("\n Attractions:\n", results["attractions"])
print("\n Itinerary:\n", results["itinerary"])
print("\n Local Guide:\n", results["guide"])
```

```
--- Travel Plan ---
Destination Overview:
 Best Seasons to Visit:
The best seasons to visit Chicago are in the spring (April to June) and fall (September to October

Safety:
Chicago, like any major city, has its share of crime, but as a tourist, you can stay safe by being

Etiquette:
In Chicago, it is important to be polite and respectful to others. When dining out, it is customary

Costs:
Chicago can be an expensive city to visit, but there are plenty of budget-friendly options availab

Language:
English is the primary language spoken in Chicago, but you may also hear Spanish, Polish, and other

Transport Options:
Chicago has a comprehensive public transportation system, including buses and trains, that can eas

Overall, Chicago is a vibrant and diverse city with something for everyone. With a moderate budget,

 Attractions:
 1. Art Institute of Chicago - Explore one of the oldest and largest art museums in the United Stat

2. Millennium Park - Visit this iconic public park to see the famous Cloud Gate sculpture, also kno

3. Chicago Cultural Center - Discover the city's cultural heritage through art exhibitions, perform

4. Chicago Architecture River Cruise - Take a boat tour along the Chicago River to learn about the

5. Chinatown - Indulge in authentic Chinese cuisine and explore the vibrant culture and shops in Ch

6. Garfield Park Conservatory - Escape the city and immerse yourself in nature at this expansive bo

7. Greektown - Enjoy traditional Greek dishes and explore the lively atmosphere of this vibrant ne

8. Field Museum - Learn about natural history, anthropology, and culture through interactive exhib

9. Chicago Food Tour - Join a guided food tour to sample various culinary delights from the city's

10. National Museum of Mexican Art - Explore Mexican art and culture through a diverse collection o

 Itinerary:
 Day 1:
- Morning: Start your day with a visit to the Art Institute of Chicago, one of the oldest and larg
- Lunch: Head to Chicago's famous deep-dish pizza joint, Giordano's, for a delicious and filling lu
- Afternoon: Take a stroll down the Magnificent Mile, a renowned shopping district in downtown Chic
- Evening: Enjoy a dinner at the Chicago French Market, where you can sample a variety of gourmet

Day 2:
- Morning: Start your day with a visit to the Museum of Science and Industry, one of the largest sc
- Lunch: Grab a bite to eat at Portillo's, a Chicago institution known for its hot dogs and Italia
- Afternoon: Visit the Chicago Cultural Center, a historic landmark that hosts free art exhibitions
- Evening: Indulge in a food tour of Chicago's diverse neighborhoods, sampling local favorites lik

Day 3:
```

Supervisor with tools Architecture - A supervisor agent uses a tool-calling agent to decide which tool to use.