exercise4

September 4, 2024

1 Import Libraries and set up LLM

[93]:
```
import os
import requests
from langchain_openai import ChatOpenAI
from langchain.prompts import ChatPromptTemplate, MessagesPlaceholder
from langchain.agents import tool
from langchain.agents import AgentExecutor
from langchain.agents.format_scratchpad.openai_tools import␣
 ↪format_to_openai_tool_messages
from langchain.agents.output_parsers.openai_tools import␣
 ↪OpenAIToolsAgentOutputParser

llm = ChatOpenAI(temperature=0)
```

2 Define the tool functions

[94]:
```
@tool
def get_weather_data(city:str):
    """
    Calls the weather Api and returns the weather data

    Args:
    city:str

    Returns:
    str
    """
    url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={os.
 ↪getenv('WEATHER_API_KEY')}&units=metric"
    response = requests.get(url)

    return str(response.json())
```

[95]:
```
@tool
def get_city_name(location: str) -> str:
```

```
    """Calls the Location API and returns the address data
    Args:
        location: str
    Returns:
        str
    """
    url = f"https://nominatim.openstreetmap.org/search?
↪q={location}&format=json&limit=1"
    headers = {
    'User-Agent': 'MyGeocodingApp/1.0 (your-email@example.com)'
}
    response = requests.get(url, headers=headers)
    if(len(response.json()) > 0):
        return response.json()[0]
    return "City not found"
```

3 Bind the tool with the llm

[97]:
```
tools = [get_city_name ,get_weather_data]
```

[98]:
```
llm_with_tools = llm.bind_tools(tools)
```

4 System Prompt

[96]:
```
prompt = ChatPromptTemplate.from_messages(
    [(
        "system",
        """
        You are a very powerful weather data expert designed to provide users␣
↪with accurate and up-to-date weather information. Your main functions␣
↪include:

        1. Call the API: Retrieve weather information using the location␣
↪provided by the user. Ensure you include parameters for current weather,␣
↪forecasts, and any relevant alerts.
        2. Display Information: Present all available details from the API␣
↪response, including:
            Current temperature
            High and low temperatures
            Feels like temperature
            Humidity
            Wind speed
            Sunrise and sunset times
            Any additional relevant weather conditions or alerts
```

```
    3. Validate Location: If the user provides an invalid city name, use a␣
↪tool to find and suggest a valid city name in English.
    Respond in a clear and organized manner to ensure users receive␣
↪comprehensive and easy-to-understand weather updates. Refer to the examples␣
↪below:

    # Examples
    ## Example 1: Valid City Name

    User Input: "What's the weather in San Francisco?"

    System Response: "Sure! Here is the current weather in San Francisco:

        Temperature: 65°F
        High/Low: 70°F / 55°F
        Feels Like: 63°F
        Humidity: 75%
        Wind Speed: 8 mph
        Sunrise: 6:45 AM
        Sunset: 7:15 PM
        Conditions: Partly cloudy

    Let me know if you need any additional information!"

    ## Example 2: Invalid City Name

    User Input: "What's the weather in Springfield?"

    System Response: "I found multiple locations with the name 'Springfield.
↪' Could you please specify the state or provide additional details? For␣
↪example, Springfield, IL or Springfield, MA."

    ## Example 3: Location Not Specified

    User Input: "I need the weather forecast."

    System Response: "Please provide a city name or location so I can␣
↪retrieve the weather forecast for you. For example, 'New York City' or␣
↪'London.'"

    ## Example 4: Weather Alert

    User Input: "Are there any weather alerts for Miami?"

    System Response: "Here is the current weather for Miami:

        Temperature: 82°F
```

```
            High/Low: 86°F / 78°F
            Feels Like: 88°F
            Humidity: 85%
            Wind Speed: 12 mph
            Sunrise: 6:30 AM
            Sunset: 7:00 PM
            Conditions: Thunderstorms

        Alert: Severe thunderstorm warning in effect until 8:00 PM. Please take␣
 ↪necessary precautions."
        """
    ),
    ("user", "{input}"),
    MessagesPlaceholder(variable_name="agent_scratchpad")
    ]
)
```

5 Build the Agent

[99]:
```
agent = (
    {
        "input": lambda x: x["input"],
        "agent_scratchpad": lambda x: format_to_openai_tool_messages(
            x["intermediate_steps"]
        ),
    }
    | prompt
    | llm_with_tools
    | OpenAIToolsAgentOutputParser()
)
```

[100]:
```
agent_executor = AgentExecutor(agent=agent, tools=tools, verbose=True)
```

6 User Input

[101]:
```
result=list(agent_executor.stream({"input": "what is the weather like in␣
 ↪nice"}))
```

```
> Entering new AgentExecutor chain…
```

```
Invoking: `get_weather_data` with `{'city': 'Nice'}`
{'coord': {'lon': 7.25, 'lat': 44}, 'weather': [{'id': 501,
'main': 'Rain', 'description': 'moderate rain', 'icon': '10d'}], 'base':
'stations', 'main': {'temp': 15.83, 'feels_like': 15.4, 'temp_min': 14.31,
'temp_max': 17.5, 'pressure': 1016, 'humidity': 74, 'sea_level': 1016,
'grnd_level': 870}, 'visibility': 7515, 'wind': {'speed': 2.08, 'deg': 183,
'gust': 2.01}, 'rain': {'1h': 3.07}, 'clouds': {'all': 100}, 'dt': 1725450247,
'sys': {'type': 2, 'id': 2085874, 'country': 'FR', 'sunrise': 1725425886,
'sunset': 1725472932}, 'timezone': 7200, 'id': 2990439, 'name': 'Arrondissement
de Nice', 'cod': 200}Sure! Here is the current weather in Nice:
- Temperature: 15.83°C
- High/Low: 17.5°C / 14.31°C
- Feels Like: 15.4°C
- Humidity: 74%
- Wind Speed: 2.08 m/s
- Sunrise: 5:38 AM
- Sunset: 6:08 PM
- Conditions: Moderate rain
Let me know if you need any additional information!

> Finished chain.
```

7 Output

[102]:
```
print(result[-1]["output"])
```

```
Sure! Here is the current weather in Nice:

- Temperature: 15.83°C
- High/Low: 17.5°C / 14.31°C
- Feels Like: 15.4°C
- Humidity: 74%
- Wind Speed: 2.08 m/s
- Sunrise: 5:38 AM
- Sunset: 6:08 PM
- Conditions: Moderate rain
```

Let me know if you need any additional information!