Objective:

To understand how to send prompts to a large language model (LLM) API using Python and interpret responses from the model.

Materials Required

- Python environment
- Internet Connection
- API Key for the LLM engine (Claude, OpenAI, etc.)
- API URL for the LLM endpoint

Background

Large Language Models (LLMs) like Claude, GPT, etc., can generate human-like text based on given prompts. These models are used in various applications, including chatbots, language translation, content generation, and more.

In this experiment, we will use Python to communicate with an LLM by sending a prompt and receiving a response. This exercise will demonstrate how to structure an API request, send it to the model, and display the result.
import requests

Python Code

```
def query_llm_engine(prompt, api_url, api_key, model="claude"):
    Query an LLM engine like Claude with a given prompt.
    Parameters:
        prompt (str): The prompt text to send to the engine.
        api_url (str): The API endpoint URL for the LLM engine.
        api_key (str): The API key for authenticating the request.
        model (str): The model to use (default: "claude").
    Returns:
        str: The response text from the LLM engine.
```

```
    headers = {
        "Authorization": f"Bearer {api_key}",
        "Content-Type": "application/json".
    payload = {
        "model": model,
        "messages": [{"role": "user", "content": prompt}],
        "temperature": 0.7,  # adjust for more creativity if needed
    }
    response = requests.post(api_url, json=payload, headers=headers)
    if response.status_code == 200:
        return response.json().get("choices", [{}])[0].get("message", {}).get("content", "No response received")
    else:
        return f"Error: {response.status_code} - {response.text}"
# Example usage
api_url = "https://api.anthropic.com/v1/engines/claude/completions"  # Replace with the actual API URL for Claude
api_key = "YOUR_API_KEY"  # Replace with your actual API key for Claude
prompt_text = "Explain the process of photosynthesis in simple terms."
response = query_llm_engine(prompt_text, api_url, api_key)
print("LLM Response:", response)
```

Example Observations

After running the code, note down:

1. The prompt you used.

2. The response given by the model.
3. Any unexpected results or errors.

Observation

- Prompt: "Explain the process of photosynthesis in simple terms."
- Response: Photosynthesis is the process by which green plants, algae, and some bacteria convert light energy from the sun into chemical energy stored in sugars like glucose. This process is vital for life on Earth, as it produces oxygen and is the foundation of the food chain.

During photosynthesis, plants take in carbon dioxide (CO_2) from the atmosphere and water (H_2O) from the soil. These raw materials enter plant cells, particularly in the chloroplasts, where the green pigment chlorophyll captures sunlight. Chlorophyll absorbs light energy, which is then used to split water molecules, releasing oxygen (O_2) as a byproduct. The energy from the sunlight excites electrons in the chlorophyll, and this energy is harnessed to create two molecules, ATP (adenosine triphosphate) and NADPH, which act as energy carriers.

The absorbed energy is then used to power a series of reactions that convert carbon dioxide into glucose ($C_6H_{12}O_6$). This glucose provides an energy source for the plant and can be stored for later use.

In essence, photosynthesis transforms sunlight into chemical energy, which sustains the plant and, through the food chain, supports nearly all life on Earth.

Analysis

1. Response Relevance: Did the response make sense for the given prompt :**YES**

Conclusion

In this experiment, I learned how to interact with a large language model (LLM) using Python by sending prompts to an API and receiving responses. By following the steps to set up an API request, we observed the process of constructing headers, payload, and managing responses in Python.This experiment demonstrates how even small adjustments, like changing the temperature setting, can significantly affect the LLM's response creativity. Additionally, the relevance and clarity of prompts play a crucial role in obtaining useful and accurate responses from the model.