Exploring Prompting Techniques for AI Audio Generation

Aim

The experiment aims to explore how different prompt styles influence AI-generated audio, such as music, sound effects, and speech. It focuses on understanding prompt engineering and optimizing prompt design for better and more relevant outputs.

Software Requirements

1. **Python**: Version 3.8+.
2. **Libraries**:
    - openai for ChatGPT.
    - requests for API handling.
    - plyer for notifications (optional).
    - gTTS or Google Cloud Text-to-Speech for audio output.
3. **API Keys**:
    - OpenAI for task organization.
    - Google Cloud Text-to-Speech (optional for audio output).

Experiment Design

Experiment 1: Basic Task Organization

- **Prompts Used**:
    - Basic Prompt: *"Here are my tasks: [list of tasks]. Help me prioritize them."*
    - Detailed Prompt: *"I have [list of tasks], with a focus on [priority]. Suggest a schedule."*
    - Contextual Prompt: *"I feel [mood]. My tasks include [list of tasks]. Plan a balanced day for me."*

- **Python Code**:

python

Copy code

```
import openai

openai.api_key = "your_openai_api_key"

def organize_tasks(prompt):
    response = openai.Completion.create(
        engine="text-davinci-003",
        prompt=prompt,
        max_tokens=300,
        temperature=0.7,
    )
    return response.choices[0].text.strip()

# Basic usage
tasks = ["Write a report", "Attend meeting", "Buy groceries", "Exercise", "Call a friend"]
priority = "Work"
mood = "Overwhelmed"

# Create a contextual prompt
prompt = f"""
I have the following tasks: {', '.join(tasks)}.
My main focus today is {priority}, and I feel {mood}.
Please:
1. Prioritize my tasks.
2. Suggest a schedule for my day.
3. Provide motivational tips to stay productive.
"""
```

```
# Get organized tasks
output = organize_tasks(prompt)
print(output)
```

Experiment 2: Enhanced Task Planning

- **Prompts Used**:
  - Mood-Specific: *"I feel [motivated/overwhelmed]. Balance work and breaks in my schedule."*
  - Detailed: *"Plan a day with 60% work, 20% fitness, and 20% relaxation."*
- **Optional Add-On**: Include **audio output** for the generated schedule.
  - Using Google Cloud Text-to-Speech:

python

Copy code

```
import requests

def generate_speech(text, output_file="schedule.mp3"):
    API_KEY = "your_google_api_key"
    url = "https://texttospeech.googleapis.com/v1/text:synthesize"
    payload = {
        "input": {"text": text},
        "voice": {"languageCode": "en-US", "name": "en-US-Wavenet-D"},
        "audioConfig": {"audioEncoding": "MP3"}
    }
    headers = {"Authorization": f"Bearer {API_KEY}"}
    response = requests.post(url, headers=headers, json=payload)
    if response.status_code == 200:
        with open(output_file, "wb") as f:
            f.write(response.content)
        print(f"Audio schedule saved to {output_file}")
```

```
    else:
        print("Error generating audio:", response.json())

# Generate audio for the output
generate_speech(output)
```

Experiment 3: Task Tracking and Notifications

- **Prompts Used**:
    - Simple: *"Remind me to [task] at [time]."*
    - Detailed: *"Send periodic motivational messages during my tasks."*
- **Code**:

python

Copy code

```
from plyer import notification
import time

def set_reminders(tasks):
    print("Setting reminders...")
    for task in tasks:
        # Simulate reminder after 10 seconds
        time.sleep(10)
        notification.notify(
            title="Task Reminder",
            message=f"Time to: {task}",
            timeout=10
        )

# Example task list for reminders
set_reminders(["Write a report", "Attend a meeting"])
```

Output and Results

- **Basic Task Organization**: Simple prompts yield generic prioritization.
- **Enhanced Task Planning**: Detailed prompts result in balanced schedules and task-specific advice.
- **Audio Integration**: Generates dynamic, auditory outputs for productivity.
- **Task Tracking**: Notifications improve task adherence.