Prompt Engineering project

Introduction

This project, *Prompt Engineering Playground with GPT-2*, provides an interactive platform for users to experiment with prompt engineering techniques to generate creative outputs using the GPT-2 language model. Prompt engineering is a critical aspect of leveraging large language models, enabling users to craft inputs that guide the model to produce desired results. The app facilitates learning and exploration of prompt tuning and hyperparameter adjustments in a user-friendly, visually appealing interface.

Methodologies

The project employs an iterative approach to text generation, focusing on three core aspects:

1. **Prompt Optimization**: Users can input a wide range of prompts to explore how different phrasing influences the generated text.
2. **Hyperparameter Tuning**: The app enables users to modify parameters such as max_length, temperature, and top_k sampling to fine-tune the behavior of GPT-2.
    - **Max Length** controls the length of generated text.
    - **Temperature** adjusts randomness, with higher values yielding more diverse outputs.
    - **Top-k Sampling** limits the model to the top-k probable words, balancing creativity and coherence.
3. **Interactive UI**: A vibrant, responsive interface ensures an intuitive user experience. Custom styles and interactive widgets enhance the usability and aesthetics of the app.

Tools and Libraries

- **Streamlit**: Used to create the app's interactive and visually engaging user interface. Features like sliders and text areas facilitate user input and configuration.
- **Transformers Library**: The pipeline from Hugging Face's Transformers library enables seamless integration of the GPT-2 model for text generation.
- **Python**: The project's core language for implementation, ensuring modularity and ease of customization.
- **CSS**: Inline CSS styles were used to design a vibrant and visually appealing UI, with custom colors, gradients, and button styles.

Conclusion

This project demonstrates the power of prompt engineering as a tool to harness the full potential of language models like GPT-2. By allowing users to experiment with prompts and parameters in real time, the app serves as a hands-on learning platform for understanding how language models operate and how their outputs can be influenced. The app's intuitive design and functionality make it a valuable tool for students, researchers, and AI enthusiasts interested in natural language processing. Future extensions could include adding support for advanced models like GPT-4, multi-language support, and additional metrics to evaluate output quality.

Python Code:

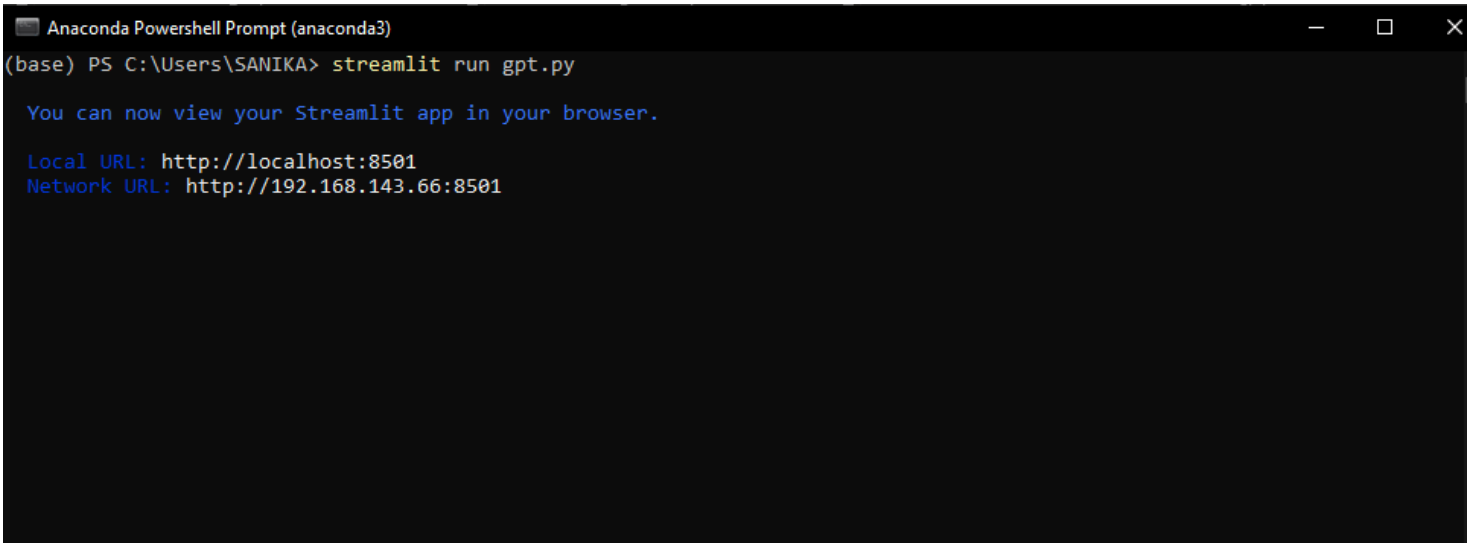
In [6]:
```
import streamlit as st
from transformers import pipeline

# Set Streamlit page config
st.set_page_config(
    page_title="Prompt Engineering Playground",
    page_icon="✨",
    layout="wide",
    initial_sidebar_state="expanded",
)

# Load the text generation pipeline
@st.cache_resource
def load_model():
    return pipeline("text-generation", model="gpt2")

text_generator = load_model()


# Custom CSS for vibrant UI
st.markdown(
    """
    <style>
    body {
        background: linear-gradient(135deg, #ff9a9e 0%, #fad0c4 99%, #fad0c4 100%);
        color: #000000;
    }
    .stApp {
        background: transparent;
    }
    .sidebar .sidebar-content {
        background: linear-gradient(135deg, #a18cd1 0%, #fbc2eb 100%);
    }
    }
    h1, h2, h3, h4, h5, h6 {
        color: #333333 !important;
    }
    .css-1fcdlhh {
        background-color: #ffffff !important;
    }
    .css-1fcdlhh:hover {
        background-color: #f2f2f2 !important;
    }
    .stButton > button {
        background-color: #ff6f61;
        color: white;
        border-radius: 8px;
        border: none;
        font-size: 18px;
        font-weight: bold;
        height: 50px;
        width: 200px;
    }
    .stButton > button:hover {
        background-color: #ff856e;
    }
    </style>
    """,
    unsafe_allow_html=True,
)
```

```python
# App title
st.title("✨ Prompt Engineering Playground with GPT-2 ✨")

# Sidebar: Prompt settings
st.sidebar.header("⚙️ Settings")
st.sidebar.markdown(
    """
    Customize your text generation settings here:
    """
)
max_length = st.sidebar.slider("Max Length:", min_value=10, max_value=200, value=50)
temperature = st.sidebar.slider("Temperature:", min_value=0.0, max_value=1.5, step=0.1, value=0.7)
top_k = st.sidebar.slider("Top-k Sampling:", min_value=1, max_value=50, value=10)

# Main: User prompt input
st.header("🎯 Experiment with Prompts")
prompt = st.text_area("Enter your prompt:", "Once upon a time in a world powered by AI,")
st.markdown("---")

# Generate button
col1, col2, col3 = st.columns([1, 2, 1])
with col2:
    generate_btn = st.button("🚀 Generate Text")

if generate_btn:
    with st.spinner("Generating response..."):
        try:
            response = text_generator(
                prompt,
                max_length=max_length,
                temperature=temperature,
                top_k=top_k,
                num_return_sequences=1,
            )
```

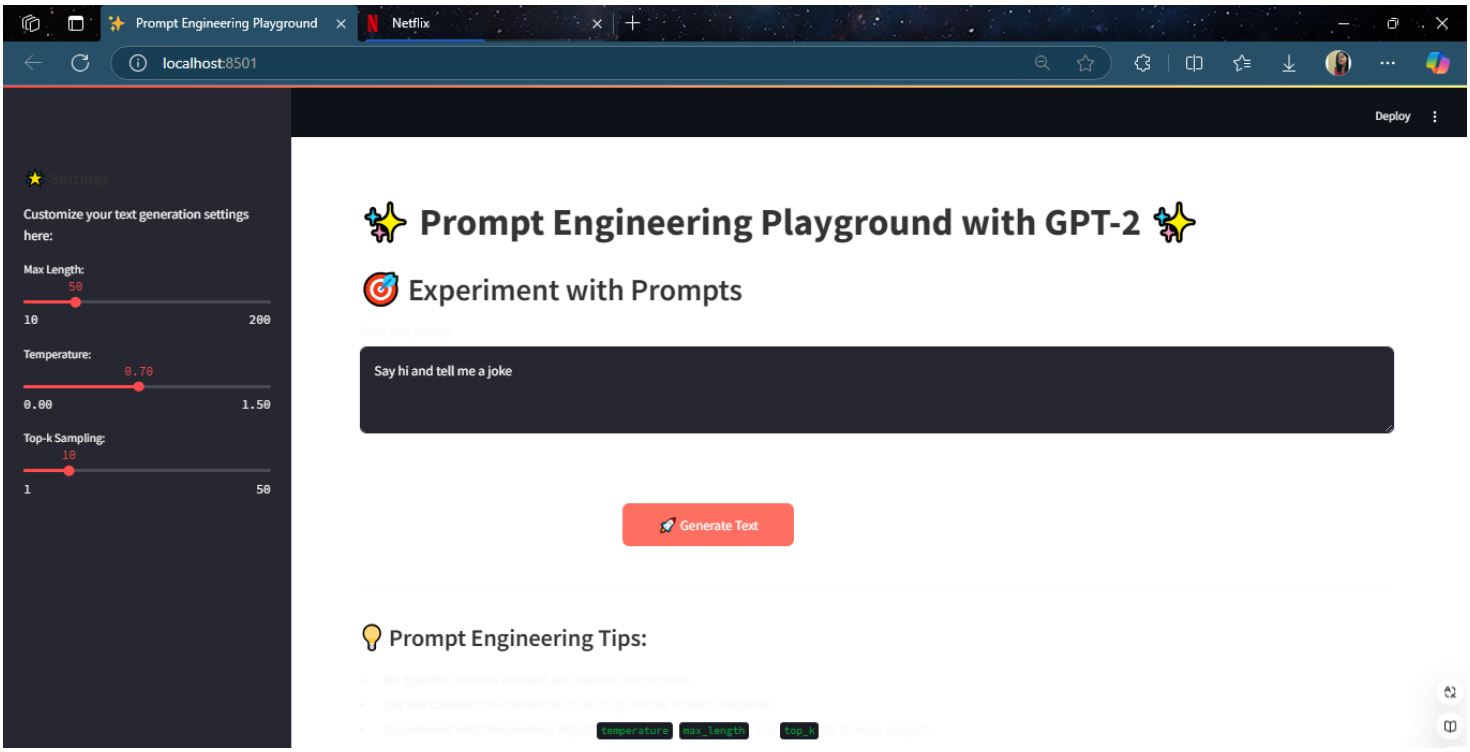
```python
if generate_btn:
    with st.spinner("Generating response..."):
        try:
            response = text_generator(
                prompt,
                max_length=max_length,
                temperature=temperature,
                top_k=top_k,
                num_return_sequences=1,
            )
            result = response[0]["generated_text"]
            st.success("Generated Text:")
            st.write(
                f"""
                <div style="background: #ffffff; color: #000000; padding: 20px; border-radius: 10px;
                box-shadow: 0px 4px 6px rgba(0, 0, 0, 0.1); font-size: 16px;">
                {result}
                </div>
                """,
                unsafe_allow_html=True,
            )
        except Exception as e:
            st.error(f"Error: {e}")
```

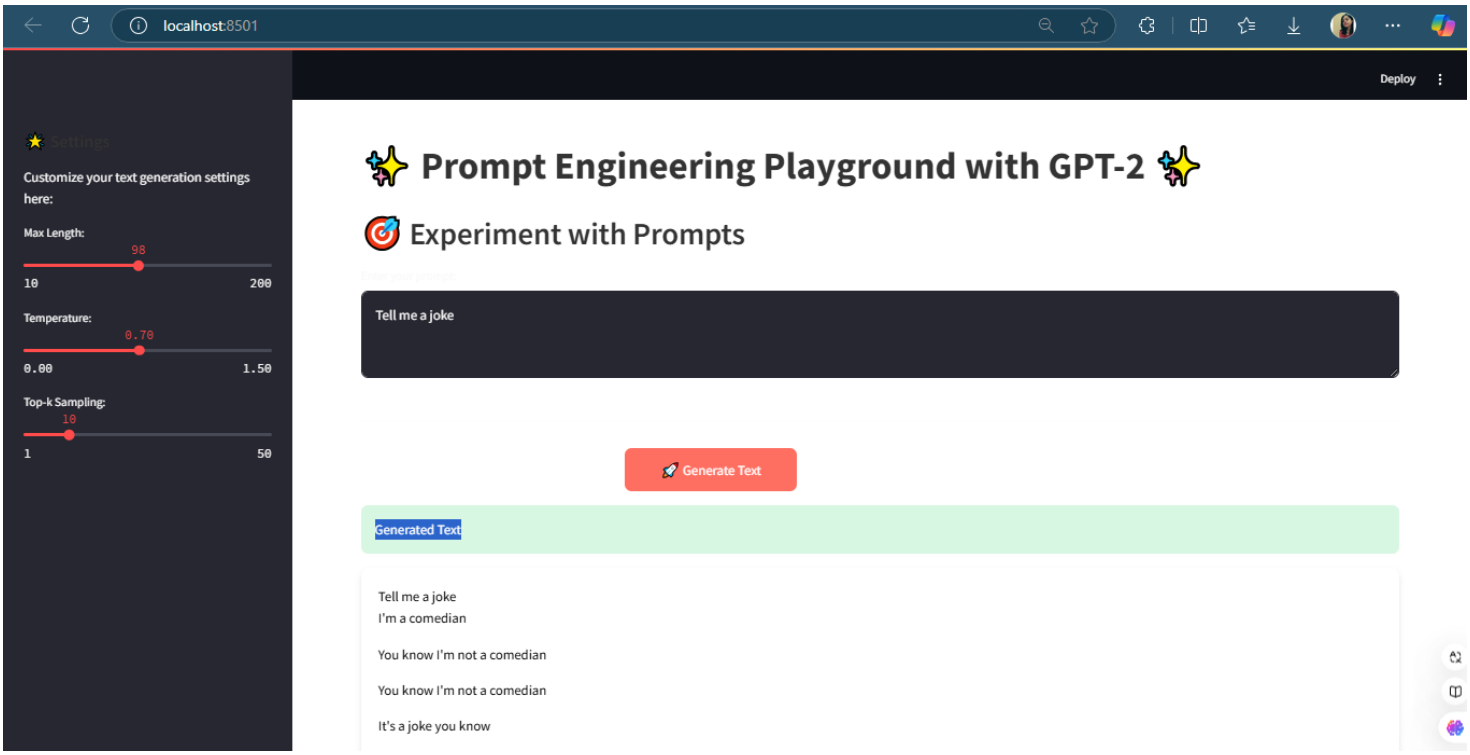
Output:

The above file is saved as a python file(gpt.py) and is run on command prompt using streamlit run gpt.py



User Interface (UI)




localhost:8501
Deploy
Settings
Customize your text generation settings here:
Max Length:
98
10
200
Temperature:
0.70
0.00
1.50
Top-k Sampling:
10
1
50
✨ Prompt Engineering Playground with GPT-2 ✨
🎯 Experiment with Prompts
Tell me a joke
🚀 Generate Text
Generated Text
Tell me a joke
I'm a comedian
You know I'm not a comedian
You know I'm not a comedian
It's a joke you know