**Text completion using Generative AI**

**Phase 2: Project Execution and Demonstration**

**1. Project Title:**

**Sentence Generator using Generative AI**

**2. Objective Recap:**

The objective of this project is to build a Sentence generation System using Generative AI models. The system accepts an initial sentence or paragraph as input and generates coherent, contextually appropriate continuations using a pre-trained language model. This demonstrates the model’s ability to understand context and extend text intelligently—useful for applications like story generation, writing assistants, and chatbot responses.

**3. Technologies Used:**

* Python
* HuggingFace Transformers
* Google Colab / Jupyter Notebook
* Pre-trained GPT-2 Model (gpt2-large)

**4. Full Code Implementation:**

**Step 1: Install Required Libraries**

!pip install transformers torch

**Step 2: Import Required Libraries**

import torch

from transformers import GPT2LMHeadModel, GPT2Tokenizer

import warnings

warnings.filterwarnings('ignore')

**Step 3: Load the Pretrained GPT-2 Model and tokenizer**

model\_name = "gpt2-large"

tokenizer = GPT2Tokenizer.from\_pretrained(model\_name)

model = GPT2LMHeadModel.from\_pretrained(model\_name)

model.eval()

# Set pad token

if tokenizer.pad\_token is None:

tokenizer.pad\_token = tokenizer.eos\_token

model.generation\_config.pad\_token\_id = tokenizer.pad\_token\_id

**Step 4: Define Text Generation Function**

def generate\_text\_from\_model(prompt, max\_length=100, temperature=0.7):

input\_ids = tokenizer.encode(prompt, return\_tensors="pt")

output = model.generate(

input\_ids,

max\_length=max\_length,

num\_return\_sequences=1,

no\_repeat\_ngram\_size=2,

top\_k=50,

top\_p=0.95,

temperature=temperature,

)

generated\_text = tokenizer.decode(output[0], skip\_special\_tokens=True)

return generated\_text

**Step 5: Run the text\_generation loop**

def text\_generation():

prompt = input("Enter prompt: ")

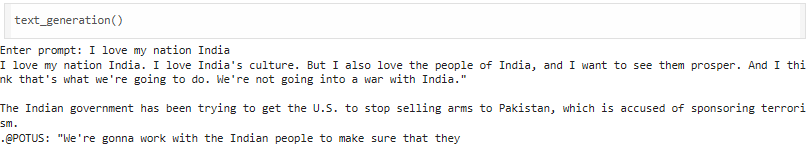
generated\_text = generate\_text\_from\_model(prompt, max\_length=100)

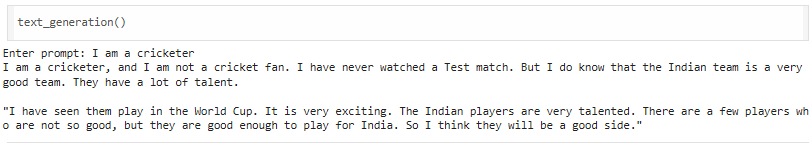
print("\nGenerated Text:\n")

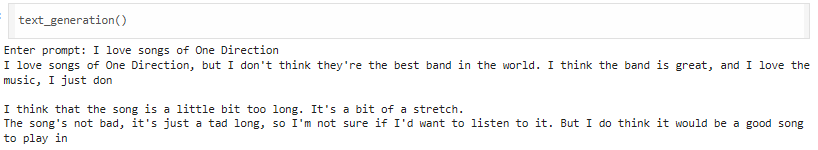
print(generated\_text)

text\_generation()

**5. Output Screenshots:**







**6. Conclusion:**

This project successfully showcases the use of a pre-trained GPT-2 model to generate meaningful and contextually relevant continuations of a given text input. By leveraging the capabilities of generative AI, the system demonstrates potential applications in content creation, writing assistance, and conversational AI. The results highlight how transformer-based models can enhance user productivity by generating fluent and coherent text in real time.

**7. References:**

* HuggingFace Transformers Documentation
* OpenAI GPT-2 Research and API
* NLP sing GPT-2 documentation