College Name - VIT Bhopal University
Student Name – Srishti Jitpure
Registration No. – 22BCE10802

Phase 2: Project Execution and Demonstration

1. Project Title:

Sentence Completion using Generative AI

2. Objective Recap:

This project aims to implement a sentence completion system using a pre-trained GPT-2 model. The system takes an incomplete sentence as input and generates a single, grammatically correct, and contextually relevant completion sentence using temperature control and repetition penalty.

3. Technologies Used:

- **Python** (Primary programming language)
- **Transformers** (For model implementation)
- **IPyWidgets** (For interactive interface)
- **Google Colab** (Development and testing)
- **Pre-trained GPT-2 Model** (For sentence generation)

4. Full Code Implementation:

Step 1: Install Required Libraries

```
!pip install transformers ipywidgets
```

Step 2: Import Required Libraries

```
import torch
from transformers import GPT2LMHeadModel, GPT2Tokenizer
import ipywidgets as widgets
from IPython.display import display
```

Step 3: Load the Pre-trained GPT-2 Model

```
# Load pre-trained model and tokenizer
model_name = 'gpt2'
tokenizer = GPT2Tokenizer.from_pretrained(model_name)
model = GPT2LMHeadModel.from_pretrained(model_name)

tokenizer.pad_token = tokenizer.eos_token
```

Step 4: Implement the Interactive Interface

```
# Define mode selection
input_box = widgets.Textarea(
    placeholder='Enter a sentence to complete...',
    description='Input Sentence:',
    layout={'width': '80%', 'height': '100px'}
)

# Define parameter controls
temperature_slider = widgets.FloatSlider(
    value=0.7,
    min=0.5,
    max=1.5,
    step=0.1,
    description='Temperature:',
)
```

```
repetition_slider = widgets.FloatSlider(
    value=1.5,
    min=1.0,
    max=2.0,
    step=0.1,
    description='Repetition Penalty:',
)

output_box = widgets.Textarea(
    value='',
    placeholder='Model output will appear here...',
    description='Completed Sentence:',
    layout={'width': '80%', 'height': '100px'},
    disabled=True
)

# Define generation function
def complete_sentence(_):
    prompt = input_box.value.strip()  # Get input sentence and
remove leading/trailing spaces
    temperature = temperature_slider.value
    repetition_penalty = repetition_slider.value

    if not prompt:
        output_box.value = "Please enter a sentence to
complete."
        return

    inputs = tokenizer(prompt, return_tensors='pt')
```

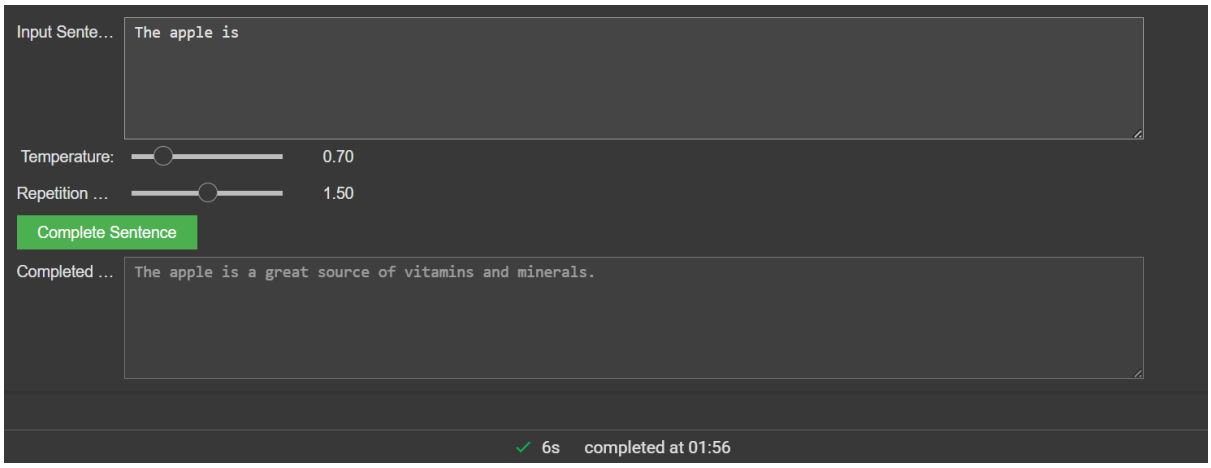
```
    outputs = model.generate(
        inputs['input_ids'],
        temperature=temperature,
        repetition_penalty=repetition_penalty,
        pad_token_id=tokenizer.pad_token_id,
        num_return_sequences=1,
        do_sample=True,
        eos_token_id=tokenizer.encode('.')[0],
        max_length=50,
    )
    completed_text = tokenizer.decode(outputs[0],
skip_special_tokens=True)


    if not completed_text.endswith('.'):
        completed_text += '.'
    completed_text = completed_text.replace(prompt,
prompt.strip(), 1)


    output_box.value = completed_text


# Define generate button
generate_button = widgets.Button(
    description='Complete Sentence',
    button_style='success'
)
generate_button.on_click(complete_sentence)


# Display widgets
display(input_box, temperature_slider, repetition_slider,
generate_button, output_box)
```

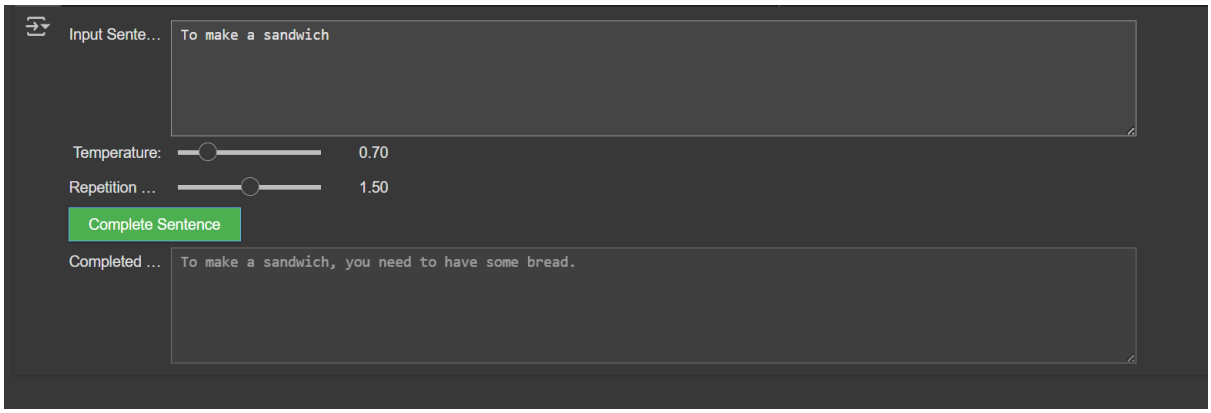
College Name - VIT Bhopal University
Student Name – Srishti Jitpure
Registration No. – 22BCE10802

Step 5: Execute the Code in Google Colab

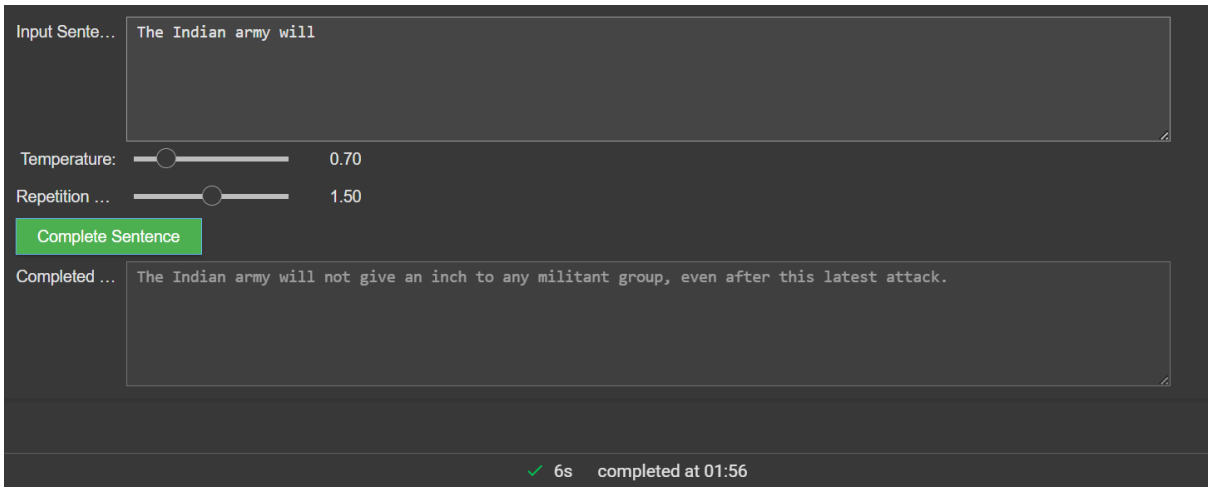
- Ensure the runtime is set to **Python 3** with GPU support for optimal performance.
- Execute each cell sequentially.
- Enter a partial sentence in the input box and adjust temperature/repetition settings as needed.
- Click **Complete Sentence** to generate a single, coherent continuation sentence.

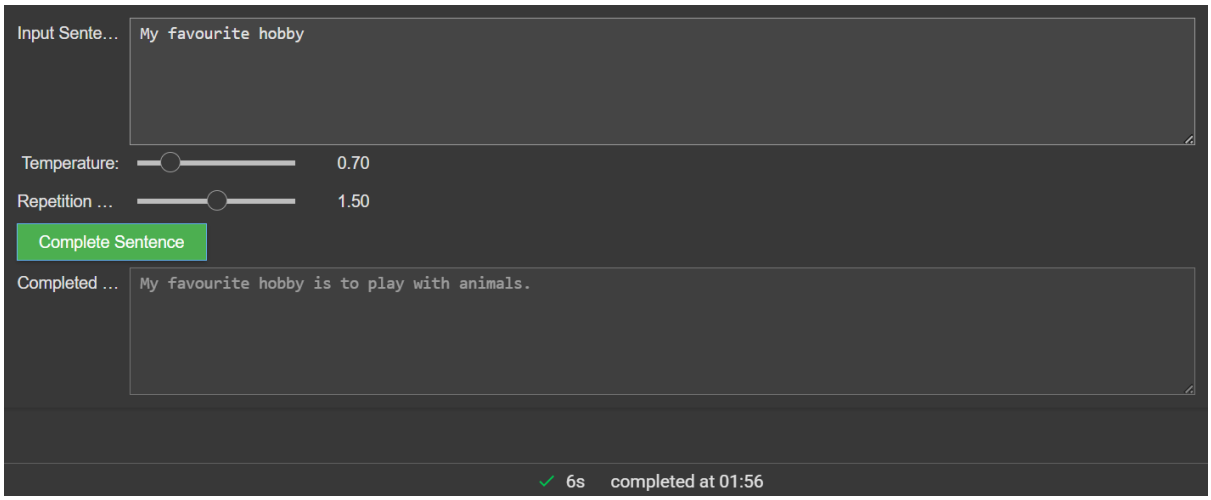
5. Output Screenshots:

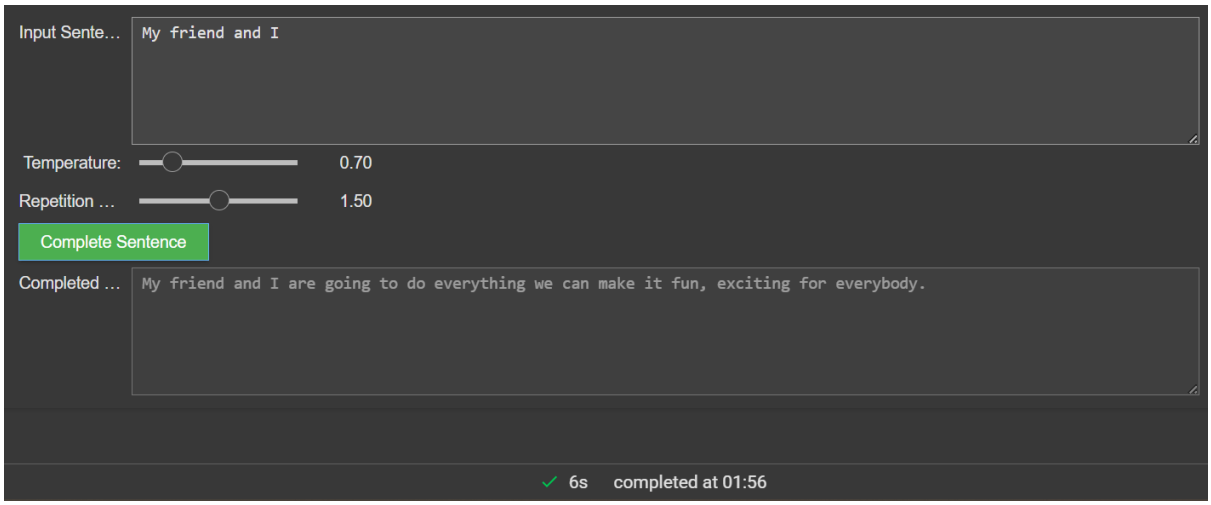




College Name - VIT Bhopal University
Student Name – Srishti Jitpure
Registration No. – 22BCE10802







College Name - VIT Bhopal University
Student Name – Srishti Jitpure
Registration No. – 22BCE10802

6. Conclusion:

The implementation successfully demonstrates the use of a pre-trained GPT-2 model to complete a given sentence based on context. By adjusting parameters like temperature and repetition penalty, users can control the creativity and coherence of the generated output. This project highlights the potential of Generative AI for intelligent text completion and sentence prediction.

7. References:

- https://www.ibm.com/think/topics/transformer-model#:~:text=Transformer%20models%20such%20as%20relational,a%20series%20of%20matrix%20multiplications.
- https://huggingface.co/openai-community/gpt2
- IPyWidgets Documentation