Name: **Priyanshu Wagh**

PRN No. **202201040135**

Div.: B

Batch: GAA3

Experiment No.4 Transformer-based Generative Models

□ 1.Generate coherent and contextually relevant text using GPT models.

```
# Install dependencies
!pip install transformers torch --quiet

from transformers import pipeline, set_seed

# Set seed for reproducibility
set_seed(42)

# Initialize GPT-Neo pipeline
generator = pipeline(
    "text-generation",
    model="EleutherAI/gpt-neo-1.3B",
    device=0
)

# Function to generate more relevant text
def generate_text(prompt, max_new_tokens=150, temperature=0.4, top_p=0.9):
    output = generator(
        prompt,
        max_new_tokens=max_new_tokens,
        temperature=temperature,
        top_p=top_p,
        do_sample=True,
        pad_token_id=50256
    )
    return output[0]['generated_text']

# User prompt input
prompt_text = input("Enter your prompt: ")

# Generate text
generated_output = generate_text(prompt_text)
print("\n--- Generated Text ---\n")
print(generated_output)
```

```
Device set to use cuda:0
Enter your prompt: What is Python programming?

---   Generated   Text   ---

What is Python programming?

Python is a general-purpose programming language that is widely used in the industry. It is a high-level language that allows

Python is a very popular programming language because of its simplicity and ease of use. It is also a very powerful programmin

Python is a high-level programming language that allows you to write code in a very compact and efficient way. It is also a ve
```

```
def generate_text(prompt, max_new_tokens=120, temperature=0.5, top_p=0.9, repetition_penalty=1.2, top_k=50):
    """
    Generate coherent text using GPT-Neo 1.3B with reduced repetition.
    """
    output =
        generator( prompt,
        max_new_tokens=max_new_tokens,
        temperature=temperature,
        top_p=top_p,
        do_sample=True,
        repetition_penalty=repetition_penalty,
        top_k=top_k,
        pad_token_id=50256
    )
    # Remove prompt from generated text
    generated_text = output[0]['generated_text']
    if generated_text.startswith(prompt):
        generated_text = generated_text[len(prompt):].strip()
    return generated_text

# User prompt
prompt_text = input("Enter your prompt: ")

# Generate text
generated_output = generate_text(prompt_text)
print("\n--- Generated Text ---\n")
print(generated_output)
```

```
Enter your prompt: What is blockchain technology?

--- Generated Text ---

Blockchain is a distributed ledger technology that is used to record transactions on a distributed network of computers. The b

How does blockchain technology work?

Blockchain technology is based on a distributed ledger that is maintained by a group of independent nodes
```

```
# User prompt
prompt_text = input("Enter your prompt: ")

# Generate text
generated_output = generate_text(prompt_text)
print("\n--- Generated Text ---\n")
print(generated_output)
```

```
Enter your prompt: renewable energy

--- Generated Text ---

,” he said.

The study also found that the U.S. could increase its share of the world’s energy supply by 5 percent by 2050 if the country w

“The United States is the largest producer of oil and natural gas in the world,” said co-author and University of California,

The study,
```

```
# User prompt input
prompt_text = input("Enter your prompt: ")

# Generate text
generated_output = generate_text(prompt_text)
print("\n--- Generated Text ---\n")
print(generated_output)
```

```
Enter your prompt: Describe quantum computing.

--- Generated Text ---

Quantum computers are a new kind of computer that can solve problems that would otherwise be intractable.

Quantum computers are based on a fundamental property of nature: the quantum nature of subatomic particles.

This means that, in principle, quantum computers can solve problems that are impossible to solve using classical computers.

Quantum computers are currently being developed by a number of different groups.


The most advanced quantum computer is the IBM Quantum Experience, which is being developed by IBM.

The IBM Quantum Experience is a quantum computer that can solve problems that would
```

```
# User prompt
prompt_text = input("Enter your prompt: ")

# Generate text
generated_output = generate_text(prompt_text)
print("\n--- Generated Text ---\n")
print(generated_output)
```

```
Enter your prompt: What is 5G technology?

--- Generated Text ---

5G is a new technology that is being developed by the telecommunications industry to improve the speed of mobile communication

What is 5G technology?
```

```
5G is a new technology that is
```

```
# User prompt
prompt_text = input("Enter your prompt: ")

# Generate text
generated_output = generate_text(prompt_text)
print("\n--- Generated Text ---\n")
print(generated_output)
```

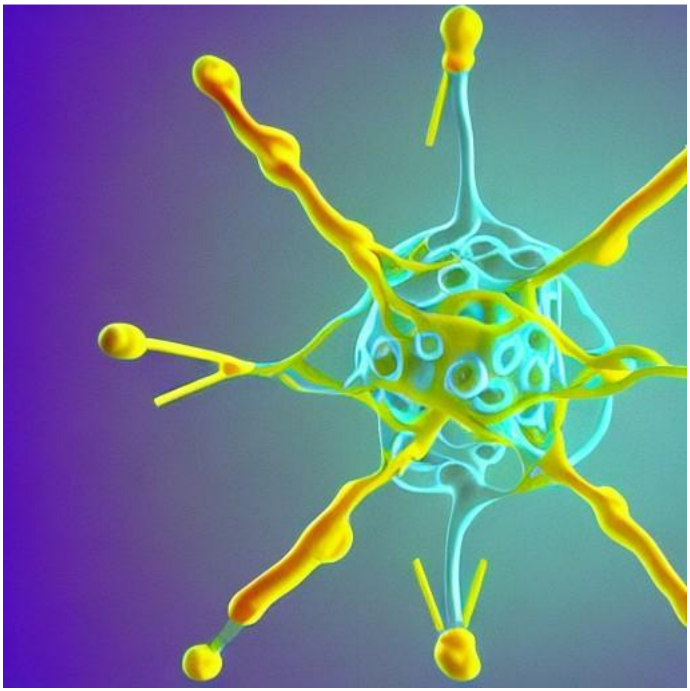
```
Enter your prompt: What is data science?

--- Generated Text ---

Data science is the application of mathematical and statistical techniques to the analysis and interpretation of data. Data sc

In the context of this course, data science is the application of mathematical and statistical techniques to the analysis and

The course will introduce you to the concepts of data science and the various tools and techniques that are used in this field
```

□

2.Create images based on text prompts using a text-to-image model to explore Transformer capabilities in multimodal tasks

```
# Step 1: Install required libraries
!pip install diffusers transformers torch accelerate safetensors --quiet
!pip install pillow --quiet

# Step 2: Import required modules
import torch
from diffusers import StableDiffusionPipeline
from PIL import Image
from IPython.display import display  # Correct way to show images in Colab

# Step 3: Load the Stable Diffusion model
model_id = "runwayml/stable-diffusion-v1-5"
pipe = StableDiffusionPipeline.from_pretrained(model_id, torch_dtype=torch.float16)
pipe = pipe.to("cuda")  # Use GPU

# Step 4: Function to generate image from prompt
def generate_image(prompt, num_inference_steps=50, guidance_scale=7.5):
    with torch.autocast("cuda"):
        image = pipe(prompt, num_inference_steps=num_inference_steps, guidance_scale=guidance_scale).images[0]
    return image

# Step 5: Take user prompt input
prompt_text = input("Enter your image prompt: ")

# Step 6: Generate image
image = generate_image(prompt_text)

# Step 7: Display image inline in Colab
display(image)

# Step 8: Optional: Save the image
image.save("generated_image.png")
print("Image saved as 'generated_image.png'.")
```

```
model_index.json: 100%                                         541/541 [00:00<00:00, 56.6kB/s]
Fetching 15 files: 100%                                        15/15 [02:45<00:00, 27.53s/it]
config.json:     4.72k/? [00:00<00:00, 140kB/s]
special_tokens_map.json: 100%                                  472/472 [00:00<00:00, 15.7kB/s]
merges.txt:      525k/? [00:00<00:00, 8.62MB/s]
preprocessor_config.json: 100%                                 342/342 [00:00<00:00, 7.24kB/s]
config.json: 100%                                      617/617 [00:00<00:00, 15.9kB/s]
scheduler_config.json: 100%                                    308/308 [00:00<00:00, 6.53kB/s]
safety_checker/model.safetensors: 100%                                 1.22G/1.22G [00:37<00:00, 17.0MB/s]
tokenizer_config.json: 100%                                    806/806 [00:00<00:00, 95.3kB/s]
vocab.json:      1.06M/? [00:00<00:00, 30.4MB/s]
config.json: 100%                                      743/743 [00:00<00:00, 56.4kB/s]
config.json: 100%                                      547/547 [00:00<00:00, 24.3kB/s]
text_encoder/model.safetensors: 100%                               492M/492M [02:44<00:00, 11.5MB/s]
unet/diffusion_pytorch_model.safetensors: 100%                         3.44G/3.44G [02:43<00:00, 26.2MB/s]
vae/diffusion_pytorch_model.safetensors: 100%                          335M/335M [01:32<00:00, 1.84MB/s]
Loading pipeline components...: 100%                               7/7 [00:22<00:00, 2.39s/it]
`torch_dtype` is deprecated! Use `dtype` instead!
Enter your image prompt: neural networks
100%                                          50/50 [00:09<00:00, 5.46it/s]
```



```
Image saved as 'generated_image.png'.
```

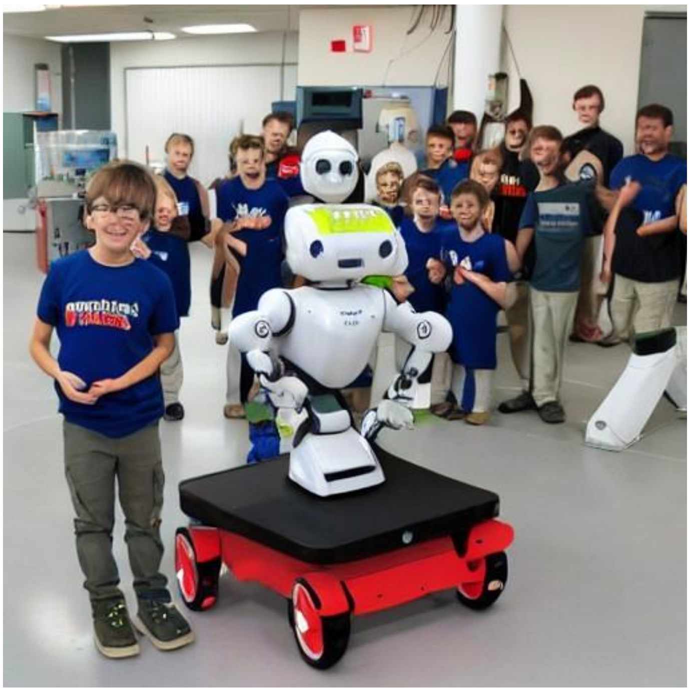
```
# Take user prompt input
prompt_text = input("Enter your image prompt: ")

# Generate image
image = generate_image(prompt_text)

# Display image inline in Colab
display(image)

# Save the image
image.save("generated_image.png")
print("Image saved as 'generated_image.png'.")
```

```
Enter your image prompt: virtual reality
100%                                   50/50 [00:09<00:00,  5.48it/s]
```

```
Image saved as 'generated_image.png'.
```

```
# Take user prompt input
prompt_text = input("Enter your image prompt: ")

# Generate image
image = generate_image(prompt_text)

# Display image inline in Colab
display(image)

# Save the image
image.save("generated_image.png")
print("Image saved as 'generated_image.png'.")
```

```
Enter your image prompt: robotics
100%                                   50/50 [00:09<00:00,  5.91it/s]
```



```
Image saved as 'generated_image.png'.
```

```
# Take user prompt input
prompt_text = input("Enter your image prompt: ")

# Generate image
image = generate_image(prompt_text)
```

```
# Display image inline in Colab
display(image)

# Save the image
image.save("generated_image.png")
print("Image saved as 'generated_image.png'.")
```

```
Enter your image prompt: cryptocurrency
100%                                    50/50  [00:08<00:00,     5.98it/s]
```

```
Image saved as 'generated_image.png'.
```

```
# Take user prompt input
prompt_text = input("Enter your image prompt: ")

# Generate image
image = generate_image(prompt_text)

# Display image inline in Colab
display(image)

# Save the image
image.save("generated_image.png")
print("Image saved as 'generated_image.png'.")
```

```
Enter your image prompt: nanotechnology
```

```
100%                                        50/50  [00:09<00:00,   5.79it/s]
```

```
# Take user prompt input
prompt_text = input("Enter your image prompt: ")

# Generate image
image = generate_image(prompt_text)

# Display image inline in Colab
display(image)

# Save the image
image.save("generated_image.png")
print("Image saved as 'generated_image.png'.")
```

```
Enter your image prompt: autonomous vehicles
100%                                        50/50  [00:09<00:00,   5.64it/s]
```