

# Advanced Text Generation Application Using Large Language Models



UNIVERSITY  
*of* VIRGINIA

SCHOOL *of* ENGINEERING  
& APPLIED SCIENCE

# The Goal of This LLM Quiz

- Leverage state-of-the-art natural language processing (NLP) techniques to generate coherent and contextually relevant text based on the input.
- Explore the creative capabilities of AI in text generation by utilizing the Hugging Face “transformers” library and the GPT-2 model.

# Overview

- Create an application using the Hugging Face “transformers” library and GPT-2 model.
- Input a prompt and customize text generation parameters.
- Save and compare the results from different parameters and make your own conclusion.

# Step 1: Install Required Packages

1. Install Python 3.7 or later version.

<https://www.python.org/downloads/>

1. Open your terminal or command prompt:

Windows: Press “Win + R” and search “cmd”

MacOS: Press “Cmd + Space” and search “Terminal”.

Linux: Press “Ctrl + Alt + T”.

# Step 1: Install Required Packages

3. Install the necessary packages in your terminal or command prompt (you may use 'pip3' here).

```
pip install transformers torch
```

## Step 2: Create the python script and implement the code

1. Create a new python file with the name 'text\_generator.py' or any name you prefer.
2. Copy and paste the code from the quiz introduction in the Canvas "Files" folder into your created 'text\_generator.py' file.

# Step 2: Create the python script and implement the code

```
generator = pipeline('text-generation', model='gpt2', device=-1) # Use CPU

def generate_text(prompt, max_length=100, temperature=0.7,
                  num_return_sequences=1):
    results = generator(prompt, max_length=max_length,
                        num_return_sequences=num_return_sequences, temperature=temperature)
    return [result['generated_text'] for result in results]
```

Create the text generator with the required parameters:

**prompt:** Input text that leads to the results.

**max\_length:** The maximum length of your generated texts (include the input prompts).

**temperature:** The randomness of your generated texts.

**num\_return\_sequences:** The number of generated paragraphs.

# Step 2: Create the python script and implement the code

In the context of **Large Language Models (LLMs)**, "temperature" refers to a parameter that controls the randomness of the model's output during text generation. It influences how the model selects the next word in a sequence based on the predicted probabilities. Here's how it works:

- **Lower Temperature (e.g., 0.1):**
  - Makes the model more deterministic and focused.
  - It prioritizes the most probable predictions, reducing randomness.
  - This is useful for tasks where precision and accuracy are important, such as factual answers.
- **Higher Temperature (e.g., 1.0 or more):**
  - Increases randomness and creativity in the output.
  - The model is more likely to select less probable predictions, leading to varied and diverse results.
  - This is beneficial for creative tasks like storytelling or brainstorming.
- **Temperature = 0:**
  - The model always chooses the highest-probability word, resulting in fully deterministic outputs.
- In summary, adjusting the temperature allows you to control the balance between creativity and reliability in the model's responses, tailoring it to different use cases.



# Step 2: Create the python script and implement the code

```
def save_to_file(text, filename='generated_text.txt'): # Change the
    filename to save the results in a different file
    with open(filename, 'w', encoding='utf-8') as file:
        file.write(text)
    print(f"Generated text saved to {filename}")
```

Save your generated texts into the target .txt file.

Change the “filename” to save the generated texts into a different file.

# Step 2: Create the python script and implement the code

```
def main():  
    print("Welcome to the Advanced Text Generator!")  
  
    prompt = input("Enter your prompt text: ")  
    max_length = int(input("Enter the maximum length of generated text  
(recommended 30-100): "))  
    temperature = float(input("Enter temperature (recommended range 0.5-  
1.0, higher means more random): "))  
    num_return_sequences = int(input("Enter the number of text paragraphs  
to generate: "))
```

Allow you to enter your own prompt, max\_length, temperature, and num\_return\_sequences in the terminal or command prompt.

## Step 2: Create the python script and implement the code

```
def main():  
  
    ''' previous code '''  
  
    # Start timing  
    start_time = time.time()  
    # Generate text  
    generated_texts = generate_text(prompt, max_length=max_length,  
    temperature=temperature, num_return_sequences=num_return_sequences)  
    # Stop timing  
    end_time = time.time()  
    print(f"\nText generation took: {end_time - start_time:.2f} seconds\n")
```

Generate the texts using defined function “generate\_text()”.

Collect the time required to generate the texts.

# Step 2: Create the python script and implement the code

```
def main():  
  
    ''' previous code '''  
  
    for i, generated_text in enumerate(generated_texts, 1):  
        print(f"Generated Text Paragraph {i}:")  
        print(generated_text)  
        print("-" * 40)
```

Print out the generated texts in different paragraphs.

## Step 2: Create the python script and implement the code

```
def main():  
  
    ''' previous code '''  
  
    save_choice = input("Would you like to save the generated text to a  
file? (yes/no): ")  
    if save_choice.lower() == 'yes':  
        all_texts = "\n\n".join(generated_texts)  
        save_to_file(all_texts)
```

Save your generated texts using the defined function “save\_to\_file()”.

# Step 3: Run the Application

1. Open your terminal or command prompt and navigate to the directory of 'text\_generator.py'.
2. Run the python script 'text\_generator.py' (you may use 'python3' here).

```
python text_generator.py
```

# Step 4: Test and Experiment

1. Try different prompts and explore how the model responds to various input prompts.
2. Try different values for maximum length, temperature, and number of outputs. Observe how it influences the generated text.

# Step 5: Observation and Conclusion

1. Make your own observation and conclusion about how the results will change when you have different parameters, especially different temperatures.
2. Please write down your observations and conclusions in a Word document, and take a screenshot of your outputs and put it in the Word document. Once completed, upload the document to the quiz.



# Step 5: Observation and Conclusion

Sample of your observation and conclusion:

When the temperature changes from XX to XX, the different generated text paragraphs tends to be more coherent/creative, and they are more similar/different with each other.

Thus the temperature will affect the XXX of the text generation process, where the low/high temperature generates more predictable texts and vice versa.