

## **Submission 2 Template**

Name	Dhanya Prabhu
SRN	PES2UG23CS169
Section	C

## **Instagram Caption Creator**

### **Problem Statement**

#### **Instagram Caption Creator**

- **Goal:** Upload a photo description (e.g., "Sunset at the beach") and get 3 cool captions.
- **Tech:** Text generation prompted with "Caption for this photo:".

### **Abstract**

Social media platforms like Instagram rely heavily on short, creative captions to engage users. Writing attractive captions for every post can be time-consuming and sometimes difficult. This project demonstrates the use of a Generative AI model to automatically generate creative Instagram captions from a simple photo description. By using a pre-trained transformer-based text generation model, the system produces multiple caption suggestions that can be directly used or modified by users.

The project highlights how decoder-based language models can be applied to real-world creative tasks using prompt engineering and text generation techniques covered in Unit 1.

### **What I Understood**

From this project, I understood that the **architecture of a language model determines the type of task it can perform well**. Decoder-only transformer models like GPT-2 are designed to generate text by predicting the next word in a sequence, making them suitable for creative tasks such as caption writing.

I also learned how **prompt engineering** and **sampling parameters** like temperature and top-p affect creativity and variation in generated text. Running the same input multiple times can produce different outputs, which is useful for generating multiple caption ideas.

## What I Built

I built an Instagram Caption Creator that takes a short photo description as input and generates three creative Instagram captions.

The system uses the HuggingFace pipeline for text generation with a pre-trained DistilGPT-2 model. The implementation is done in a Jupyter Notebook, making it easy to test different inputs and observe the outputs.

The project does not involve training or fine-tuning the model. Instead, it focuses on applying an existing transformer model to a practical use case using concepts taught in Unit 1.

An interactive user interface was created inside the notebook to allow users to experiment with different inputs and creativity levels.

## Screenshots:

The screenshot shows a Jupyter Notebook cell with the following Python code:

```
caption_generator = pipeline(  
    "text-generation",  
    model="distilgpt2"  
)
```

Below the code, a progress bar indicates "Loading weights: 100%" with "76/76 [00:00<00:00, 123.74it/s, Materializing param=transformer.wte.weight]".

Output from the GPT2LMHeadModel LOAD REPORT command is shown:

```
GPT2LMHeadModel LOAD REPORT from: distilgpt2  
Key | Status | |  
-----+-----+  
transformer.h.{0, 1, 2, 3, 4, 5}.attn.bias | UNEXPECTED | |
```

Notes section:

- UNEXPECTED :can be ignored when loading from different task/architecture; not ok if you expect identical arch.

The user interface for generating captions is displayed below:

- Photo Descr.: Mountain view during sunrise
- Captions: 3
- Creativity: 1.10
- Generate Captions button
- Generated Captions:
  - Setting 'pad\_token\_id' to 'eos\_token\_id':50256 for open-end generation.  
Both 'max\_new\_tokens' (=256) and 'max\_length' (=60) seem to have been set. 'max\_new\_tokens' will take precedence. Please refer to the documentation for more information. (<https://huggingface.co>)
  - 1. "Climbing over the valley," a picture taken at the Mojave Desert in the early 1990s. Courtesy: Yosemite National Park, C
  - 2. "Sailor Mountain View is a beautiful place. This place, and all mountains and all mountains. This lake represents one of
  - 3. Mountain view during sunrise

