**Project-3: Text Summarization**

**Submitted By: Rajesh Kumar Sharma**

**Date: 29-Nov-24**

**Project-3:** **Develop a Text summarization application and expose it as a FastAPI**

**Code Link:** <https://github.com/rajeshksharmasls/GenAI/tree/master/GenAI/Text%20Summarization>

**Dataset:** *None*

**Process:** Below is the summarization of the code:

This code implements a **Text Summarization API** using **FastAPI** and Hugging Face's **T5-small** model. Here's a concise explanation:

**Key Components:**

1. **Model Initialization**:
   * Loads the pre-trained **T5-small** model and its tokenizer from Hugging Face.
   * The model specializes in text-to-text tasks like summarization.
2. **FastAPI Setup**:
   * Initializes a FastAPI application with a title and version.
   * Includes endpoints for text summarization and a welcome message.
3. **Input Validation**:
   * A **Pydantic model** (StoryRequest) validates the input to ensure a Story string is provided.
4. **Summarization Logic**:
   * **POST Endpoint (/Summarize)**:
     + Accepts a request containing a text story.
     + Validates that the input story is non-empty.
     + Prepares a prompt: "Please summarize the following text:\n\n{Story}".
     + Tokenizes the input and truncates it if it exceeds 512 tokens.
     + Uses the model to generate a summary with a maximum of 50 tokens.
     + Decodes the generated summary into human-readable text.
   * Returns the original story and the generated summary in JSON format.
   * Raises a 400 HTTP error if the input is empty.
5. **Root Endpoint**:
   * **GET Endpoint (/)**:
     + Returns a welcome message to confirm the API is running.

**Key Features:**

* **Text Summarization**:
  + Uses a lightweight pre-trained T5-small model for efficient summarization.
  + Can handle stories up to 512 tokens and outputs concise summaries.
* **API Accessibility**:
  + FastAPI ensures easy integration and efficient request handling.
  + Built-in validation with Pydantic for robust input checking.
* **Customizable**:
  + The prompt-based design can be adjusted for specific summarization requirements.

This API is suitable for summarizing user-provided texts and can serve as a lightweight, scalable backend for summarization tasks.

Requirements.txt

transformers

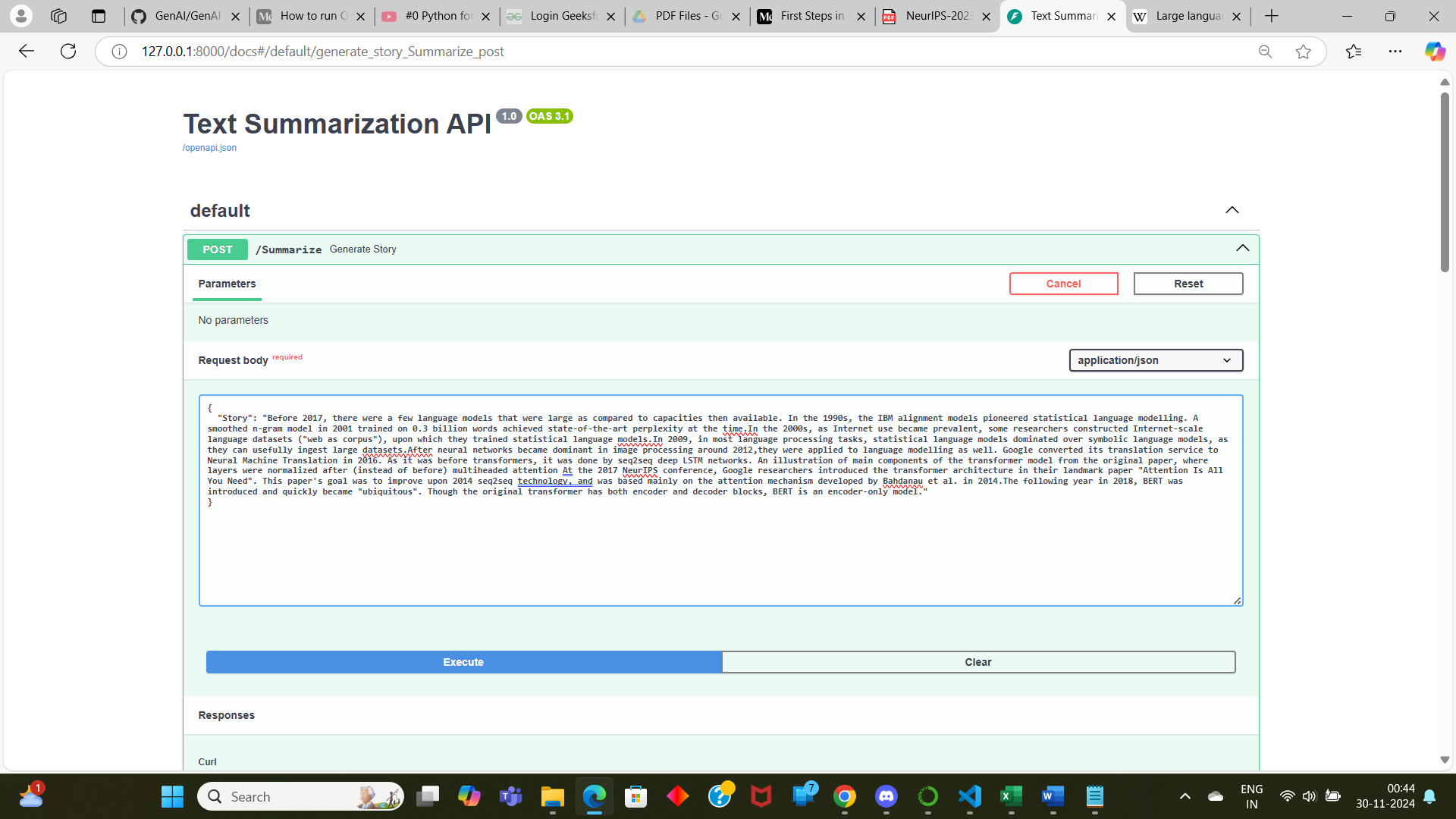
fastapi

uvicorn

torch

sentencepiece

**Snapshots:**

**

**Conclusion:** *Text is getting summarized as per the text provided.*