### Vipul V Gupta

**A-328**

### MSC CS PART 1 ADT MINI PROJECT

### AI-Based-Hybrid-Text-Summary-Project-NLP- Documentation

This project is a hybrid text summarization system that intelligently chooses between a local NLP model and the Gemini API for generating summaries. The system is designed to provide quick results for short texts using local processing while leveraging a powerful external API for more comprehensive summarization of longer documents.

### 1. Core Features

* **Hybrid Summarization Logic**: The system automatically determines the best summarization method based on the input text's length.
  + **Local NLP**: Used for shorter texts to provide fast, efficient summarization without an API call.
  + **Gemini API**: Utilized for longer documents to generate high-quality, nuanced summaries.
* **Manual Override**: Users have the option to manually select which summarization method they want to use, overriding the automatic selection.
* **Graceful Fallback**: If the Gemini API fails or is unavailable, the system automatically falls back to the local NLP model to ensure a summary is still generated.
* **Intuitive User Interface (UI)**: The UI provides clear visual feedback, including:
  + Indicators showing which method (Local NLP or Gemini API) was used.
  + Clear error messages if a process fails.

### 2. Technology Stack

This project uses the following technologies:

* **Python**: The primary programming language for the backend logic.
* **Flask**: A micro-framework used to build the web application and handle HTTP requests.
* **Local NLP Library**: The specific library is not explicitly named in the repository files, but common choices for this type of project would include **NLTK**, **spaCy**, or **Gensim**.
* **Gemini API**: An external API used for advanced text summarization.
* **HTML**: Used to build the user interface (index.html).
* **CSS/JavaScript**: Likely used for styling and front-end interactivity, though not explicitly detailed in the provided file list.

### 3. File Structure

The repository contains the following key files:

* app.py: This is the main backend application file. It contains the Flask server, the logic for text length detection, the API integration for Gemini, the local NLP summarization function, and the handling for the manual override and fallback mechanisms.
* index.html: The front-end user interface. This file contains the HTML for the text input area, the buttons for manual override, the display area for the summary, and the visual indicators for the method used.
* requirements.txt: Lists all the necessary Python libraries and their versions that need to be installed to run the project. This ensures a consistent and reproducible environment.
* README.md: Provides a general overview of the project, its purpose, and core features.
* .gitignore: Specifies files and directories that Git should ignore, such as virtual environment files or API keys.

### 4. Setup and Installation

To set up and run this project locally, follow these steps:

1. **Clone the repository**:  
   git clone https://github.com/Vipul7362/AI-Based-Hybrid-Text-Summary-Project-NLP-.git  
   cd AI-Based-Hybrid-Text-Summary-Project-NLP-
2. **Create a virtual environment** (recommended):  
   python -m venv venv
3. **Activate the virtual environment**:
   * **On Windows**: venv\Scripts\activate
   * **On macOS/Linux**: source venv/bin/activate
4. **Install the required libraries**:  
   pip install -r requirements.txt
5. **Set up the Gemini API key**: You will need to obtain a Gemini API key and configure it in app.py or as an environment variable for the application to function correctly.
6. **Run the application**:  
   python app.py  
     
   The application will start, and you can access it by navigating to http://127.0.0.1:5000 in your web browser.