**AI Based Content Summarization System**

**1. Scope:**

The objective of this project is to develop a text summarization tool capable of generating concise summaries from large amounts of data, such as articles, research papers, PDF files, and other forms of content.

The tool aims to provide a quick and efficient way for users, including students and teachers, to understand the main points and overall content of a given topic without having to read through lengthy documents in their entirety.

Some of the Applications of the text summarization are: -

* Financial research
* Social Media Marketing
* Literature and books
* Class assignments and e-learning
* Legal contract analysis

**2. Select:**

Our project aims to develop a robust text summarization tool using the **Gemini AI API**, powered by Google, within the **Jupiter Notebook (Python)** environment. This tool will leverage the advanced natural language generation capabilities of Gemini AI to provide users with concise and informative summaries of input text.

Gemini AI offers a comprehensive suite of features, including content generation, text summarization, creative writing assistance, personal assistants and chatbots, language translation, data augmentation, and content curation and recommendation. These capabilities enable our text summarization tool to cater to a wide range of applications and use cases, from automated content creation to enhancing user interactions through conversational agents.

By utilizing the interactive environment of **Jupyter Notebook**, developers can efficiently develop and test the text summarization algorithm, leveraging Python's extensive ecosystem of libraries and tools. The flexibility of Jupyter Notebook allows for easy iteration and experimentation, facilitating the refinement of the text summarization tool and integration with Gemini AI API.

The App\_key is generated via Get API key in google AI Studio

A screenshot of a computer

Description automatically generated

**3. Adapt & Align:**

The Gemini AI is customized to integrate the specified generative AI service and comply with individual project needs.

The LLM (Language Model) used is the **"gemini-pro"** model from the Google Generative AI API. It stands for Large Language Model, which is a type of deep learning model capable of understanding and generating human-like text. The "**gemini-pro"** model is specifically designed for text generation tasks and is one of the models available within the Google Generative AI API.

In this setting up the API and establishing the connection between the Gemini Api and the UI (user interface)

A screenshot of a computer code

Description automatically generated

A diagram of a model

Description automatically generated

**Model => “Gemini-Pro”**

**Input:** User API\_key (Unique for every user), File, Prompt.

**Output:** Summary of the given content.

**4. Application Integration:**

Integrate generative AI functionality into an application workflow to provide advanced text summarization capabilities. This involves developing functions to handle various input file formats, such as PDF, DOC, and more, and extracting text from these files. Additionally, we will implement functions to allow users to enter prompts and generate summaries to their specific needs.

Designing the GUI interface allows users to easily interact with the application, providing input prompts, selecting input files, and viewing the generated summaries.

Initializing the Gemini AI API with the provided API key, leveraging its capabilities to generate summaries based on user prompts and input files, and presenting the results through the GUI interface.

**Imports:**

* **google. generativeai:**  This library enables integration with Google's generative AI models, likely for summarizing the extracted text.
* **PyPDF2:** This library facilitates reading and extracting text from PDF files.
* **tkinter:** This library provides the foundation for building the graphical user interface (GUI).
* **filedialog:**  This module helps create file selection dialogs within the GUI.
* **messagebox:** This module allows displaying message boxes to the user.

**Functions:**

**Initialize model:** Initializes the Generative Model with the provided API key.

**Extract\_text\_from\_pdf:** Extracts text from a PDF file.

**Generate\_summary:** Generates the summary using the Generative AI model and displays it in the GUI.

**Browse\_file:** Opens a file dialog for selecting a PDF file.

**Creating the Main Window:** Initializes the tkinter main window with the title "Text Summarization".

**Styling:** Defines custom styles for labels, buttons, entry fields, and text areas.

**API Key Entry:** Allows the user to enter the API key required for accessing the Generative AI model.

**File Selection:** Provides an option to select a PDF file using a file dialog.

**Prompt Entry:** Allows the user to enter a prompt for the text summarization process.

**Summary Display:** Displays the generated summary in a text area.

**Generate Button:** Executes the generate\_summary function when clicked.

**Running the GUI:** Initiates the tkinter event loop to run the GUI.

**GUI Elements:**

**Labels:** enter API key, select a file, and enter prompt.

**Entry fields:** Allow users to input API key, file path, and prompt.

**Button:** Triggers the summary generation process.

**Text box:** Displays the generated summary of the PDF content.

**Workflow:-**

Start Application

↓

Enter API Key

↓

Select PDF File

↓

Enter Prompt

↓

Generate Summary

↓

Display Summary

↓

End Application

**Output display: -**

**A screenshot of a computer

Description automatically generated**

**Input file: - Artificial intelligence research paper.**

**Outputs: -**

**Prompt: -** What are the anticipated implications of developing Artificial Intelligence for the future.

**A screenshot of a computer

Description automatically generated**

**Output: - Zero-shot Inference**

**Prompt:-** What is the main focus of the research paper?

**A screenshot of a computer

Description automatically generated**

**Output: one-shot Inference**

**Prompt:** What is the author's conclusion on the implementation of AI in many fields?

**A screenshot of a computer

Description automatically generated**

**Output: few-shot Inference:**

**Prompt:** How is the architecture of the Siri virtual assistant described in the paper?

**A screenshot of a computer

Description automatically generated**