EDUCATIONAL ROBOT PLATFORM

INTRODUCTION:

The project we're working on involves creating an Educational Robot Platform (AI) chatbot that can respond to user queries based on predefined intents and also search for relevant information within a PDF document or through YouTube videos. The chatbot uses natural language processing (NLP) techniques to understand and classify user inputs, providing responses from either a set of predefined intents or external sources such as PDFs and YouTube.

Key Components:

1. **Intents JSON File**: This file contains a set of predefined user intents, each with associated patterns (possible user inputs) and responses.
2. **Natural Language Processing (NLP)**: This involves tokenizing, stemming, and vectorizing the user inputs to classify them into one of the predefined intents.
3. **Neural Network Model**: The model is trained to classify user inputs into the appropriate intents based on the processed input data.
4. **PDF Search Functionality**: The chatbot can search for specific terms or phrases within multiple PDF documents and provide relevant text as a response.
5. **YouTube Search Functionality**: The chatbot can search YouTube for relevant videos based on the user's query and provide links to these videos.
6. **Speech Recognition**: The chatbot can take voice input from the user, process it, and respond accordingly using text-to-speech for output.

Dataset Description:

Intents JSON File

The intents file (intents.json) contains the data required to train the model and respond to user queries. This file is structured as follows:

- **Intents**: A list of intents where each intent has:
    - **Tag**: A unique identifier for the intent.
    - **Patterns**: Various user inputs that correspond to the intent.
    - **Responses**: Potential responses the chatbot can give when an intent is matched.
    - **Context** (optional): Contextual information for handling follow-up questions.

PDF Search

The search_pdf function utilizes PyPDF2 to read the PDF documents and RecursiveCharacterTextSplitter to split the text into manageable chunks. It searches these chunks for the query term and returns the relevant text.

YouTube Search

The search_youtube function uses the YouTube Data API to search for relevant videos based on the user's query. It returns the title and URL of the most relevant video.

Chatbot Functionality

The chatbot combines the intent-based response system with the PDF and YouTube search functionalities to provide comprehensive answers:

1. **Intent Matching**: If the user query matches one of the predefined intents, a corresponding response is provided.
2. **PDF and YouTube Search**: For queries starting with specific phrases like "what is" or "tell me about", the chatbot searches the PDF and YouTube for relevant information.

SREAMLIT:

The Streamlit application allows users to interact with the chatbot through a web interface, either by typing their queries or speaking into a microphone. The chatbot can respond with text or provide links to YouTube videos, along with relevant text from PDF documents.

Libraries and Dependencies:

The following libraries and dependencies are used in the project:

- **Streamlit**: For creating the web interface.
- **NumPy**: For handling arrays and numerical computations.
- **TensorFlow and TFLearn**: For building and training the neural network model.
- **NLTK**: For natural language processing tasks like tokenization and stemming.
- **PyPDF2**: For reading and extracting text from PDF files.
- **LangChain**: For splitting large texts into manageable chunks.
- **SpeechRecognition**: For recognizing speech input from the user.
- **pyttsx3**: For text-to-speech functionality.
- **Google API Client**: For searching YouTube videos.

CONCLUSION:

- The chatbot platform detailed in this documentation showcases a sophisticated and versatile approach to user interaction, combining natural language processing, machine learning, and multimodal input methods to deliver accurate and helpful responses. By leveraging advanced technologies and a structured architecture, this chatbot can not only provide immediate answers to user queries but also continually improve its performance over time.
- Key features, such as the integration of PDF and YouTube search capabilities, voice interaction, and a user-friendly web interface, highlight the platform's adaptability and broad application potential. Whether used for educational purposes, customer service, or other domains, this chatbot offers a robust solution that can be tailored to meet specific needs.
- As development continues, incorporating user feedback and expanding the dataset will further enhance the chatbot's accuracy and functionality. This platform sets a solid foundation for creating interactive, intelligent, and responsive chatbot applications, promising a significant impact on user engagement and information accessibility.