**GPT-2 Chatbot Prototype Documentation**

# Overview

This documentation provides a comprehensive guide to the GPT-2-based chatbot prototype. The prototype allows users to interact with a fine-tuned GPT-2 language model through a web-based interface.

# Components

## 1. Flask Server (app.py)

* Purpose:
  + Serves as the backend for handling user input and generating bot responses.
  + Utilizes the Flask web framework for seamless development.
* Functionality:
  + Listens for user input through a simple HTML form.
  + Utilizes a fine-tuned GPT-2 model to generate coherent responses.
  + Responds with bot-generated messages through API endpoints.

## 2. HTML Interface (index.html)

* Purpose:
  + Provides a user-friendly web interface for interacting with the chatbot.
  + Displays chat-like conversation history.
* Key Elements:
  + Input form for user messages.
  + Chat history display area.

## 3. Stylesheet (styles.css)

* Purpose:
  + Enhances the visual presentation of the HTML elements.
* Styles:
  + Differentiates user and bot messages.
  + Creates a clean and modern chat interface.

## 4. Fine-Tuned GPT-2 Model

* Purpose:
  + Generates contextually relevant responses based on user input.
* Training Steps:
  + Dataset Preparation:
    - Creation of a conversation dataset with diverse topics.
  + Tokenization and Special Tokens:
    - Tokenization of the dataset using the GPT-2 tokenizer.
    - Introduction of special tokens for improved training.
  + Model Fine-Tuning:
    - Fine-tuning the GPT-2 model on the custom dataset.
    - Adjustment of hyperparameters for optimal performance.
  + Model Evaluation:
    - Evaluation of the model on a validation dataset.
    - Utilization of metrics such as perplexity for assessment.
  + Saving the Fine-Tuned Model:
    - Saving the fine-tuned GPT-2 model and tokenizer for future use.

# Usage

## 1. Setup

* Requirements:
  + Python and pip installed on the system.
  + Required Python packages installed using:
    - ***pip install Flask transformers torch***

## 2. Running the Server

* Command:
  + ***python app.py***
* Access:
  + Visit http://127.0.0.1:5000/ in a web browser.

## 3. Interacting with the Chatbot

* Steps:
  + Type a message in the input field.
  + Press "Send" to interact with the chatbot.
  + View the chat history displaying user and bot messages.

## 4. Customizations

* Initial Bot Response:
  + Removal of the initial part of the bot's response if identical to the user input.
* Response Truncation:
  + Truncation of the bot's response after encountering ".", ",", or "!" for brevity.
* Adjusting Response Length:
  + The max\_length parameter in the model.generate call can be adjusted for desired response length.

# Considerations

* The prototype uses a simplified tokenization approach; advanced tokenization may be explored for complex input structures.
* Fine-tuning was performed on a limited dataset, and responses may vary based on input.

# Testing

