**PubMed Article Summarizer Documentation**

**Overview**

This document outlines the development process for a web application that allows users to upload CSV files containing PubMed articles for summarization. The application utilizes the T5 model for text summarization and is built using Python with Flask framework.

**Process Outline**

1. **Data Exploration and Preparation:**
   * Identify and prepare a sample dataset of PubMed articles in CSV format.
   * Ensure the dataset includes a column named article containing the full text of each PubMed article.
2. **Model Selection:**
   * Selected the T5 model from Hugging Face Transformers for text summarization.
   * Reasons for choosing T5:
     + State of the art performance in summarization tasks.
     + Ability to handle long documents efficiently.
     + Pre-trained weights and fine-tuning capabilities.
3. **Model Fine-Tuning:**
   * Downloaded and fine-tuned the T5 model on a local machine using TensorFlow or PyTorch.
   * Fine-tuning steps:
     + Preprocessed the PubMed articles by cleaning and tokenizing the text.
     + Defined training parameters and performed multiple epochs to optimize model performance.
     + Saved the fine-tuned model checkpoints for deployment.
4. **Web Application Development:**
   * Developed a web application using Flask, a lightweight Python web framework.
   * Designed and implemented the front-end using HTML/CSS for user interaction.
   * Integrated back-end functionality to:
     + Handle file uploads (CSV files containing PubMed articles).
     + Use the fine-tuned T5 model to summarize each article.
     + Display the original articles and their summarized versions on the web interface.

### Setup and Running Instructions

**Prerequisites**

Ensure the following are installed:

* Python (3.x recommended)
* Flask (pip install Flask)
* Transformers library (pip install transformers)
* NLTK (pip install nltk)

**Steps to Run the Application**

1. **Clone the Repository**:

bash

Copy code

git clone https://github.com/your/repository.git

cd repository-name

1. **Install Dependencies**:

bash

Copy code

pip install -r requirements.txt

1. **Download NLTK Resources**: Run Python and execute the following:

python

Copy code

import nltk

nltk.download('stopwords')

nltk.download('punkt')

1. **Fine-Tune the T5 Model**:
   * Run the code given for the model to fine tune it.
   * Ensure the fine-tuned model weights are saved in a directory accessible to the Flask application.
2. **Run the Flask Application**:

bash

Copy code

python app.py

This will start the Flask development server.

1. **Access the Web Application**: Open a web browser and go to http://127.0.0.1:5000/.
2. **Upload CSV Files**:
   * Use the provided form to upload a CSV file containing PubMed articles.
   * Each row in the CSV file should have an article column with the full text of a PubMed article.
3. **View Summarized Articles**:
   * Upon submission, the application will summarize each article using the fine-tuned T5 model.
   * The original articles and their summarized versions will be displayed in a table format on the web interface.

### Additional Considerations

* **Error Handling**: Ensure robust error handling for file uploads, model inference, and application logic.
* **Performance**: Optimize the application for performance, considering large CSV files and complex text summarization tasks.
* **Security**: Implement necessary security measures to protect user data and prevent vulnerabilities.

### Conclusion

This documentation provides a detailed overview of the PubMed Article Summarizer web application, covering data preparation, model selection and fine-tuning, Flask application development, setup instructions, and additional considerations. Follow these steps to deploy and utilize the application effectively.