**PubMed Article Summarization using Generative AI (T5 Model)**

**Table of Contents:**

1. Introduction
2. Data Exploration and Preparation
3. Model Selection and Fine-tuning
4. Evaluation
5. Web Application Development
6. Running the Application
7. Additional Setup
8. Conclusion

**1. Introduction**

This project aims to develop a web application that summarizes PubMed articles using a fine-tuned T5 (Text-to-Text Transfer Transformer) model. The T5 model is a powerful generative AI model designed for various text-based tasks. This documentation details the process of data exploration, model selection, fine-tuning, evaluation, and web application development.

**2. Data Exploration and Preparation**

**Data Exploration**

We use the "ccdv/pubmed-summarization" dataset from Hugging Face, specifically the PubMed subset. This dataset contains **Articles** and their **Abstracts**.

**Data Preparation**

To prepare the data for training, we need to firstly clean the text in then dataset (train, test, validation), removing digits, special characters and unnecessary spaces. Then involves tokenizing the articles and abstracts and ensuring they are properly formatted for the model.

**3. Model Selection and Fine-tuning**

**Model Selection**

We use the T5 model, which is available from Hugging Face's Transformers library. We start with the t5-small variant for this project.

**Fine-tuning**

Fine-tuning the model involves training it on the tokenized dataset. We use the Trainer class from the Transformers library to manage the training process. After fine-tuning, the model and tokenizer are saved for later use in the web application.

**4. Evaluation**

Evaluating the model's performance is crucial. We use the ROUGE metric for this purpose. This involves comparing the generated summaries to the reference summaries (abstracts) in the dataset.

**5. Web Application Development**

**Flask Setup**

I used Flask to create a web application that allows users to input or upload PubMed articles for summarization. The application has the following features:

* A form to input PubMed articles.
* A backend that processes the input and generates a summary using the fine-tuned T5 model.
* A results page that displays the original article and the generated summary.

**HTML Templates**

I created HTML templates for the main page and the results page. The main page contains a form for article input, and the results page displays the original and summarized articles.

**6. Running the Application**

**Activate Virtual Environment**

Ensured that virtual environment or Local Host is activated. This is important for managing dependencies and ensuring that the correct versions of libraries are used.

**Run the Flask Application**

Run the Flask application to start the web server. Then access the application through a web browser.

**Access the Application**

Open your web browser and go to http://127.0.0.1:5000/ to access the PubMed Article Summarizer.

**7. Additional Setup**

**Install Dependencies**

Ensured that I have the required dependencies installed like transformers, and datasets.

**Model and Tokenizer**

Ensured that fine-tuned model and tokenizer are saved in the appropriate directory. This is crucial for loading the model in the Flask application.

**8. Conclusion**

This documentation provides a comprehensive guide to developing a PubMed article summarization web application using the T5 model. The steps include data exploration and preparation, model selection and fine-tuning, evaluation, and web application development. Follow the instructions to fine-tune the model, evaluate its performance, and run the web application successfully.