Detailed Documentation for Text Processing with Flask Application

Project Overview

This project is a Flask-based web application for text processing tasks including text summarization, keyword extraction, and sentiment analysis. It leverages various Natural Language Processing (NLP) libraries and models to provide these functionalities. The application has a user-friendly front-end built with HTML and CSS, and it allows users to input text, process it, and view the results.

Directory Structure

```
your_project/
│
├── venv/
│
├── app.py
├── requirements.txt
├── README.md
├── templates/
│   └── index.html
└── static/
    └── styles.css
```

Installation and Setup

Prerequisites

Python 3.6 or higher
Virtual environment tool (venv)

Steps

1. Desh will give it to you:

2. Create and activate a virtual environment:

```
python -m venv venv
.\venv\Scripts\activate  # On Windows
```

```
source venv/bin/activate  # On macOS/Linux
```

3. Install dependencies:

```
pip install -r requirements.txt
```

4. Run the application:

```
python app.py
```

5. Open your web browser and go to http://127.0.0.1:5000/.

Application Details

1. Text Summarization

Description:
Text summarization is the process of shortening a text document to create a summary that retains the most important points of the original document.

Implementation:
Model Used: sshleifer/distilbart-cnn-12-6 from Hugging Face.
Library: transformers by Hugging Face.
Method: A pipeline for text summarization is created using the transformers library. The model is loaded and used to generate summaries from the input text.

API Endpoint:
URL: /summarize
Method: POST
Input: JSON containing the text to summarize.
Output: JSON containing the summarized text.

Example:
Input:

```
{
  "text": "Natural Language Processing (NLP) is a sub-field of artificial intelligence (AI) that focuses on the interaction between computers and humans through natural language. The ultimate objective of NLP is to enable computers to understand, interpret, and respond to human languages in a valuable way."
}
```

Output:

```
{
  "summary": "NLP is a sub-field of AI that focuses on the interaction between computers and humans through natural language, aiming to enable computers to understand,
interpret, and respond to human languages."
}
```

2. Keyword Extraction

Description:
Keyword extraction is the process of extracting the most relevant words and phrases from a text document.

Implementation:
Library: yake (Yet Another Keyword Extractor).
Method: The YAKE library is used to extract keywords from the input text. It uses statistical methods to identify keywords that are most relevant to the text.

API Endpoint:
URL: /keywords
Method: POST
Input: JSON containing the text to extract keywords from.
Output: JSON containing the extracted keywords.

Example:
Input:

```
{
  "text": "Machine learning (ML) is the study of computer algorithms that improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so."
}
```

Output:

```
{
  "keywords": ["Machine learning", "computer algorithms", "artificial intelligence", "training data", "predictions"]
}
```

3. Sentiment Analysis

Description:
Sentiment analysis is the process of determining whether a piece of text is positive, negative, or neutral.

Implementation:
Library: textblob.
Method: The TextBlob library is used to analyze the sentiment of the input text. It calculates a sentiment polarity score that ranges from -1 (negative) to 1 (positive).

API Endpoint:
URL: /sentiment
Method: POST
Input: JSON containing the text to analyze.
Output: JSON containing the sentiment score and category (positive, negative, neutral).

Example:
Input:

```
{
  "text": "I had a wonderful experience at the new restaurant. The food was delicious and the staff were very friendly and attentive. I will definitely be coming back!"
}
```

Output:

```
{
  "sentiment": "Positive (0.43198051948051946)"
}
```