AI News Summariser Documentation

Project Setup

Prerequisites:

Ensure you have Python 3.8+ installed.

Installation Steps:

1. Clone the repository:

```
git clone <https://github.com/amaanglen/ai_news_summariser>

cd ai_news_summariser
```

2. Create and activate a virtual environment:

```
python -m venv venv
source venv/bin/activate  # On Windows use `venv\Scripts\activate`
```

3. Install required dependencies:

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

4. Create a .env file and add the API keys:

```
GEMINIAPI_KEY=<your_google_gemini_api_key>
NEWS_KEY=<your_newsapi_key>
```

5. Run the application:

```
streamlit run app.py
```

Model Details

The application utilizes the following models:

- **Summarization Model**: Google Gemini AI generates summaries of the scraped news articles.
- **Sentiment Analysis Model**: Google Gemini AI analyzes the sentiment of news articles, categorizing them as positive, negative, or neutral.
- **Text-to-Speech (TTS) Model**: Google Text-to-Speech (gTTS) is used to convert Hindi summaries into audio output.

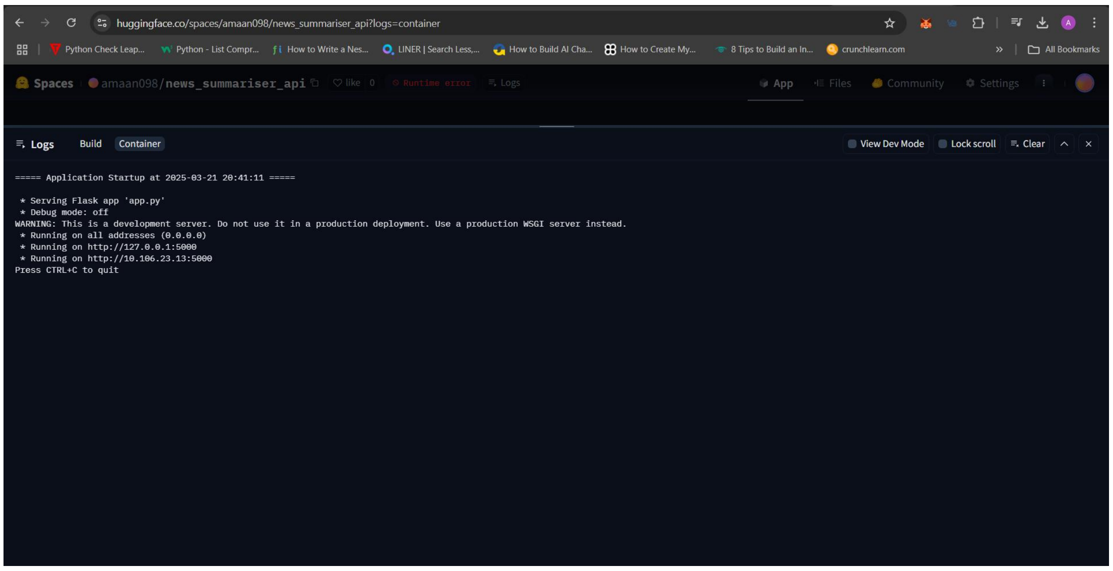
Web Scraping Process

The application scrapes news articles using the following approach:

- **Fetching HTML Content**: The scrape_webpage function sends an HTTP request to the news article URL and retrieves the HTML content.
- **Parsing the Content**: BeautifulSoup is used to parse the HTML and extract the main article text, removing unnecessary elements like ads and navigation menus.
- **Cleaning the Text**: The extracted text is cleaned by removing excessive whitespace, special characters, and non-relevant sections.
- **Passing to AI Model**: The cleaned article text is then sent to Google Gemini AI for summarization and sentiment analysis.

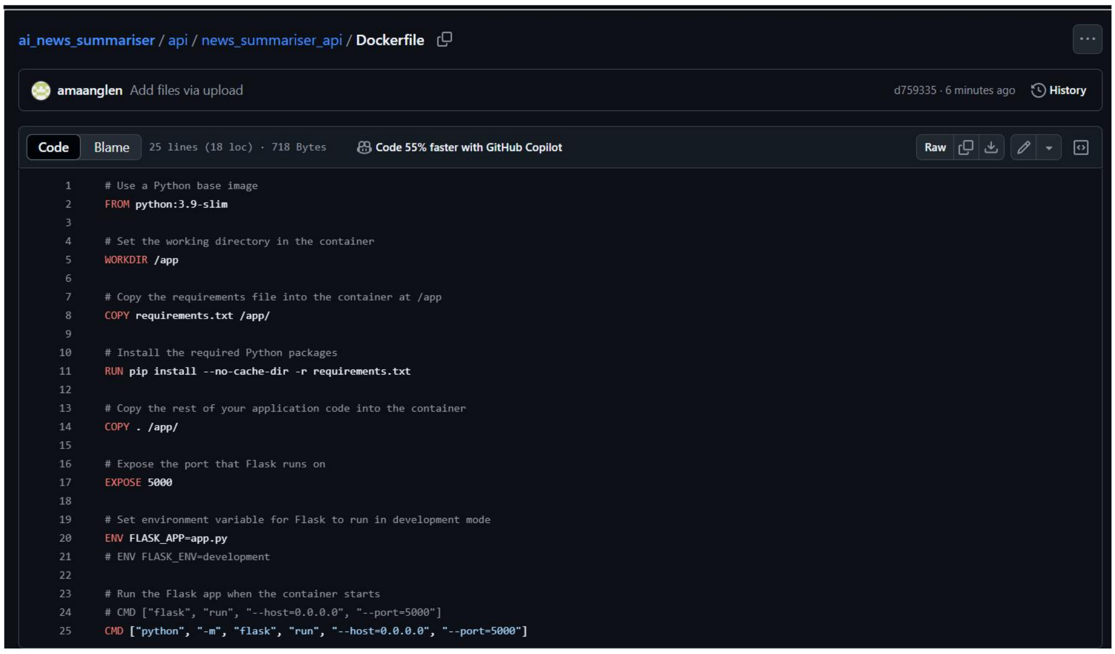
API Development

- **News Retrieval API**: The get_company_news function queries the NewsAPI to fetch the latest news articles related to a given company.
- **Web Scraping API**: The scrape_webpage function extracts the main text content from news articles using BeautifulSoup.
- **Analysis API**: The analyze_news function sends the extracted news content to Google Gemini AI for summarization and sentiment analysis.
- **Translation API**: The translate_to_hindi function translates the analysis results into Hindi.
- **Text-to-Speech API**: The get_audio_data function converts the Hindi summary into an audio file using gTTS.



```
===== Application Startup at 2025-03-21 20:41:11 =====

 * Serving Flask app 'app.py'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://10.106.23.13:5000
Press CTRL+C to quit
```


ai_news_summariser / api / news_summariser_api / Dockerfile
amaanglen Add files via upload
d759335 · 6 minutes ago
History
Code
Blame
25 lines (18 loc) · 718 Bytes
Code 55% faster with GitHub Copilot
Raw

```dockerfile
# Use a Python base image
FROM python:3.9-slim

# Set the working directory in the container
WORKDIR /app

# Copy the requirements file into the container at /app
COPY requirements.txt /app/

# Install the required Python packages
RUN pip install --no-cache-dir -r requirements.txt

# Copy the rest of your application code into the container
COPY . /app/

# Expose the port that Flask runs on
EXPOSE 5000

# Set environment variable for Flask to run in development mode
ENV FLASK_APP=app.py
# ENV FLASK_ENV=development

# Run the Flask app when the container starts
# CMD ["flask", "run", "--host=0.0.0.0", "--port=5000"]
CMD ["python", "-m", "flask", "run", "--host=0.0.0.0", "--port=5000"]
```

Deployed the API using Docker container on Hugging Face Spaces

Flask API deployed on Hugging Face using Docker container:

https://huggingface.co/spaces/amaan098/news_summariser_api

API Usage

- **NewsAPI**: Used to fetch the latest company news articles.
- **Google Gemini AI**: Used for summarization and sentiment analysis.
- **gTTS (Google Text-to-Speech)**: Used for generating Hindi audio summaries.

Assumptions & Limitations

- **Assumptions:**
    - The Google Gemini AI key is valid and has sufficient quota.
    - NewsAPI provides reliable and up-to-date articles.
    - Web scraping is permitted by the respective news websites.
- **Limitations:**
    - The free tier of NewsAPI has request limits.
    - Google Gemini AI requires API key-based authentication.
    - The summarization and sentiment analysis depend on the accuracy of the AI model.
    - gTTS may not handle long summaries efficiently.
    - Web scraping may fail if the article structure varies significantly.