News Summarization and Text-to-Speech(TTS) Application

Overview

This project provides a web-based application that extracts key details from multiple news articles related to a given company, performs sentiment analysis, conducts a comparative analysis, and generates a text-to-speech (TTS) output in Hindi

Documentation Requirements

1. Project Setup

Steps to Install and Run the Application

Backend Setup

1. **Clone the Repository**

```
git clone https://github.com/kiruba11k/NewsSummarizerTTSHuggingSpace.git

cd backend
```

2. **Create a Virtual Environment**

```
python -m venv venv

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

3. **Install Dependencies**

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

4. **Run the Flask Application**

```
python api.py
```

Frontend Setup

1. **Install Gradio**

```
pip install gradio requests
```

2. **Run the Gradio Interface**

```
python app.py
```

---

2. Model Details

The application utilizes the following models

Summarization

- Extracts key information from news articles.
- Utilizes **TextRank Algorithm** or **TF-IDF-based extraction**.

Sentiment Analysis

- **NLTK's Vader SentimentIntensityAnalyzer** is used to classify sentiment.
- Provides insights into **positive, negative, and neutral** sentiments.

Text-to-Speech (TTS)

- **gTTS (Google Text-to-Speech)** is used for generating Hindi audio summaries.
- Converts final sentiment analysis text into an audio file.

---

3. API Development

The Flask backend provides RESTful APIs for:

Endpoints:

1. Analyze News

Method : POST

https://kiruba11-news-tts-backend.hf.space/analyze_news

Request:

```
{
 "company": "Tesla"
}
```

Response:

```
{
 "Company": "Tesla",
 "Articles": [...],
 "Comparative Sentiment Score": {...},
 "Final Sentiment Analysis": "Tesla's latest news coverage leans positive.",
 "Audio": "https://generated-audio-url.mp3"
}
```

2. Get Audio

Method : GET

https://kiruba11-news-tts-backend.hf.space/analyze_news/get_audio

Request:

```
GET /get_audio?file_path=<filename>
```

Response: Returns the generated speech file.

---

4. API Usage

Third-party APIs Used

- **Google News RSS Feed:** Fetches recent news articles for a given company.
- **Google Translator:** Converts text from English to Hindi.
- **gTTS:** Converts text to speech for audio output.

---

5. Frontend (Gradio UI)

app.py

- **Imports:**

```
import gradio as gr
import requests
```

- **Interacts with Flask API:**

```
FLASK_API_URL = "https://Kiruba11-NEWS-TTS-Backend.hf.space/analyze_news"
```

- **Launches Gradio Interface:**

```
iface.launch()
```

---

6. Backend (api.py)

Key Functionalities:

1. **Fetches news articles** from Google News.
2. **Analyzes sentiment** using NLTK's Vader.
3. **Generates Hindi summaries** using Google Translator.
4. **Creates text-to-speech output** using gTTS.

Dependencies:

- Flask, nltk, gtts, requests, bs4, deep_translator, waitress

---

7. Utility Functions (utils.py)

- fetch_articles(): Fetches news articles.
- analyze_sentiment(): Determines sentiment polarity.
- extract_topics(): Identifies key topics.
- text_to_speech(): Converts Hindi text to audio.