Project Report: News Sentiment Analysis with Hindi TTS

1. Project Title:

News Sentiment Analysis with Hindi TTS

---

2. Objective:

The objective of this project is to develop a web-based application that:

- Extracts key details from multiple news articles related to a given company.
- Performs sentiment analysis on the extracted content.
- Conducts a comparative analysis across articles.
- Generates Hindi Text-to-Speech (TTS) output from the summarized content.
- Provides a user-friendly interface to display the results.

---

3. Technologies Used:

Backend (Flask)

- **Flask:** RESTful API for handling data processing.
- **BeautifulSoup4:** Web scraping for extracting news articles.
- **newspaper3k:** Extracting and parsing article content.
- **gTTS:** Generating Hindi Text-to-Speech (TTS) audio.
- **pandas:** Data structuring and manipulation.
- **textblob:** Sentiment analysis.

Frontend (Streamlit)

- **Streamlit:** Web-based user interface.
- **requests:** API communication with the backend.
- **pandas:** Displaying and formatting article details.

Data Handling & Storage:

- **CSV files:** Storing extracted article details.
- **TTS Audio Files:** MP3 files generated and stored in the output directory.

Version Control:

- **GitHub:** Repository for code management and sharing.

---

4. Project Flow

The project follows a streamlined flow with the following steps:

✅ **Step 1: User Input**

- The user provides the company name via the Streamlit frontend interface.
- The company name is sent as a POST request to the Flask backend API.

✅ **Step 2: News Extraction**

- The backend scrapes news articles using BeautifulSoup and newspaper3k.
- It extracts article details, including:
    - **Title**
    - **Summary**
    - **URL**
    - **Publish Date**
    - **Domain**
- The extracted data is stored in a CSV file for later processing.

✅ **Step 3: Sentiment Analysis**

- The backend reads the CSV file containing the articles.
- Sentiment analysis is performed using TextBlob to classify the articles into:
    - **Positive**
    - **Negative**
    - **Neutral**
- A comparative analysis calculates the sentiment distribution.

✅ **Step 4: Text-to-Speech (TTS) Generation**

- The summarized content of all articles is combined.
- The combined text is converted into Hindi speech using gTTS.
- The TTS audio is saved as an MP3 file in the output directory.

✅ **Step 5: API Response**

- The backend sends a JSON response containing:
    - Sentiment summary (Positive, Negative, Neutral distribution)
    - List of extracted articles with titles, summaries, and URLs
    - The path of the Hindi TTS audio file

✅ **Step 6: Displaying Results**

- The frontend displays:

- **Sentiment summary** in JSON format
- **List of articles** with clickable URLs
- **Hindi TTS audio player** with a download option

---

5. Functionality & Features

✅ **1. News Extraction**

- Scrapes news articles related to a given company using BeautifulSoup and newspaper3k.
- Extracts and stores the following details:
  - **Title**
  - **Summary**
  - **URL**
  - **Publish Date**
  - **Domain**

✅ **2. Sentiment Analysis**

- Analyzes article content using TextBlob.
- Classifies each article as:
  - **Positive**
  - **Negative**
  - **Neutral**
- Performs a comparative analysis to calculate sentiment distribution across multiple articles.

✅ **3. Text-to-Speech (TTS)**

- Converts the summarized content into Hindi speech using gTTS.
- Stores the audio output as an MP3 file in the output directory.

✅ **4. Web-Based User Interface**

- Streamlit-based interface allows users to:
  - Input a company name.
  - Trigger the news extraction and sentiment analysis.
  - View the sentiment report and article details.
  - Listen to or download the Hindi TTS audio output.

✅ **5. API Communication**

- Uses a RESTful Flask API to handle backend processing.

- Frontend communicates with the backend via API calls.

✅ **6. Results Display**

- Displays:
    - **Sentiment distribution** (Positive, Negative, Neutral).
    - **Extracted articles** with clickable links.
    - **Hindi TTS audio** with a download option.

---

6. Results and Analysis

The application successfully extracts and analyzes news articles related to multiple companies.

Example Output for Reliance:

✅ **Sentiment Distribution:**

- Positive: 60%
- Negative: 30%
- Neutral: 10%

✅ **Extracted Articles:**

- **Title:** "Reliance Stocks Surge in Market Rally"
- **Summary:** "Reliance Industries witnessed a sharp increase in stock prices..."
- **URL:** Read More

✅ **Hindi TTS Output:**

- **File:** reliance_tts_hindi.mp3
- **Text:** Summarized content in Hindi speech format.

---

7. Key Challenges and Solutions

📌 **1. Scraping JavaScript-heavy websites**

- **Challenge:** Many news websites use JavaScript to dynamically load content, making it difficult to scrape using BeautifulSoup.
- **Solution:** Focused on non-JS websites compatible with BeautifulSoup for reliable extraction.

📌 **2. Hindi TTS Quality**

- **Challenge:** Initial TTS output quality was low.
- **Solution:** Used gTTS for better-quality Hindi TTS output.

📌 **3. Sentiment Accuracy**

- **Challenge:** TextBlob provides basic sentiment analysis, which may not be accurate for complex articles.
- **Solution:** Future enhancement could use VADER or BERT-based models for better accuracy.

---

8. GitHub Repository

The entire project source code is available on GitHub:
GitHub Link: <GITHUB_REPO_LINK>

---

9. Future Enhancements

1. **Improved Sentiment Analysis:**
    - Replace TextBlob with VADER or DistilBERT for more accurate classification.
2. **Visualization:**
    - Use Matplotlib or Plotly for graphical representation of sentiment trends.
3. **Multi-Language TTS:**
    - Add support for multiple regional languages.
4. **Real-Time News Extraction:**
    - Use **RSS feeds** or **APIs** for real-time news extraction.
5. **Deployment:**
    - Deploy the project on **Hugging Face Spaces** or **Render** for public access.

---

10. Conclusion

The **News Sentiment Analysis with Hindi TTS** project successfully:

- Extracts and analyzes news articles.
- Performs sentiment analysis.
- Generates Hindi TTS output.
- Provides a user-friendly interface using Streamlit.
- Offers structured results through a RESTful Flask API.

✅ The project demonstrates efficient **news extraction, sentiment analysis, and Hindi TTS generation**, making it a practical solution for monitoring company-related news trends.