PROJECT DOCUMENTATION

Project Setup

Steps to Install and Run the Application

1. Clone the Repository:

   ```
   git clone https://github.com/your-username/company-news-sentiment-analysis.git
   cd company-news-sentiment-analysis
   ```

2. Create a Virtual Environment:

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

3. Install Dependencies:

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

4. Set Up API Keys:
   a. **NewsAPI:** Sign up at NewsAPI and get your API key. Place the NEWS_API_KEY in news_fetcher.py.
   b. **Groq API:** Sign up at Groq and get your API key. Place the api_key in sentiment_analysis.py and summarizer.py.
5. Run the Application:

   ```
   streamlit run app.py
   ```

Model Details

Models Used in the Project

1. Sentiment Analysis and Summarization:
   a. Model: qwen-2.5-32b (via Groq Cloud API).
   b. Purpose:

i. Sentiment Analysis: Analyzes the sentiment of each article and provides a score (ranging from -1 to +1) and sentiment category (Positive, Negative, Neutral).

```
system_prompt = """You are a sentiment analysis
model and summarizer. The user will give the
company name and news article as input.
You have to analyze the news concerning the company
to generate the output. You need to determine
whether the news affects the company positively or
negatively.
The output should be in JSON format:
{
    "Score": ,
    "Sentiment": ,
    "Summary": ,
    "Keywords":
}
- Score must be in range [-1,+1] with 2 decimal
places
- Sentiment must be Positive/Neutral/Negative based
on score
- Summary should be 2-3 lines focusing on company
impact
- Keywords should be 3-5 most important topics"""
```

ii. Summarization: Combines individual article summaries and sentiment scores into an overall summary.

```
system_prompt = """You are a professional
summarizer and sentiment analyst. The user will
provide a company name and a list of news articles
about the company.
Each article includes a summary and a sentiment
score (ranging from -1 to +1, where -1 is negative,
close to 0 is neutral, and +1 is positive).
Your task is to combine these summaries into a
single, concise, and coherent overall summary.
The overall summary should:
1. Be 3-4 sentences long.
```

```
2. Include an analysis of the overall sentiment
based on the individual sentiment scores.
3. Focus on the key points and themes across all
relevant summaries.
4. Highlight the most important developments or
news about the company.
5. Avoid redundancy and repetition.
6. Maintain a neutral and professional tone."""
```

c. Integration: The Groq API is used to interact with the qwen-2.5-32b model.

2. Text-to-Speech (TTS):
    a. Model: googletrans + gTTS.
    b. Purpose:
        i. Translates the overall summary into Hindi using googletrans.
        ii. Converts the Hindi text into an audio file using gTTS.

API Development & Usage

How third-party APIs Are Used,

1. Groq API:
    a. Purpose: Used for sentiment analysis and summarization.
    b. Integration: The sentiment_analysis.py and summarizer.py modules interact with the Groq API.
    c. Input: Article content or summaries.
    d. Output: Sentiment scores, summaries, and keywords.
2. NewsAPI:
    a. Purpose: Used to fetch news articles related to a given company.
    b. Integration: The news_fetcher.py module interacts with the NewsAPI.
    c. Input: Company name.
    d. Output: List of articles with titles, URLs, and content.

3. Bing News RSS:
    a. Purpose: Used as an alternative to NewsAPI for fetching news articles.
    b. Integration: The news_fetcher2.py module scrapes news articles from Bing News RSS.
    c. Input: Company name.
    d. Output: List of articles with titles, URLs, and content.

Accessing APIs via Postman

1. Groq API:
    a. Endpoint: https://api.groq.com/openai/v1/chat/completions
    b. Method: POST
    c. Authorization: Bearer <GROQ_API_KEY>
    d. Content-Type: application/json
    e. Body:

```
{
  "model": "qwen-2.5-32b",
  "messages": [
    {"role": "system", "content": "You are a sentiment analysis model_____"},
    {"role": "user", "content": <article_content>}
  ]
}
```

2. NewsAPI:
    a. Endpoint: https://newsapi.org/v2/everything
    b. Method: GET
    c. Query Parameters:
        i. q: Company name.
        ii. pageSize: Number of Articles
        iii. apiKey: Your NewsAPI key.

    Example: https://newsapi.org/v2/everything?q=Reliance&apiKey=<NEWS_API_KEY>

Assumptions & Limitations

Assumptions

1. **News Sources:** The application assumes that NewsAPI and Bing News RSS will provide sufficient and relevant news articles for any given company.
2. **Article Content:** The application assumes that the full content of articles can be scraped using BeautifulSoup for non-JS websites.
3. **Sentiment Analysis:** The sentiment analysis model (qwen-2.5-32b) is assumed to provide accurate sentiment scores and summaries.
4. **Translation and TTS:** The translation and TTS models (googletrans and gTTS) are assumed to work reliably for converting English text to Hindi speech.

Limitations

1. **NewsAPI Limitations:** NewsAPI has a limited number of free requests per day. For large-scale usage, a paid plan is required.
2. **Scraping Limitations:** The application can only scrape non-JS websites. Websites with dynamic content (e.g., JavaScript-rendered content) may not be fully scraped.
3. **Translation Accuracy:** The translation accuracy depends on the googletrans library, which may not always produce perfect translations.
4. **TTS Quality:** The quality of the Hindi TTS output depends on the gTTS library, which may not always produce natural-sounding speech.

Repository Structure

The repository includes the following files:

1. **app.py:** The main Streamlit application script.
2. **news_fetcher.py:** Fetches news articles using NewsAPI.
3. **news_fetcher2.py:** Fetches news articles using Bing News RSS.
4. **sentiment_analysis.py:** Performs sentiment analysis using the Groq API.
5. **summarizer.py:** Generates an overall summary using the Groq API.
6. **tts.py:** Translates and generates Hindi audio using googletrans and gTTS.
7. **README.md:** Provides setup and usage instructions.
8. **requirements.txt:** Lists all Python dependencies.

Conclusion

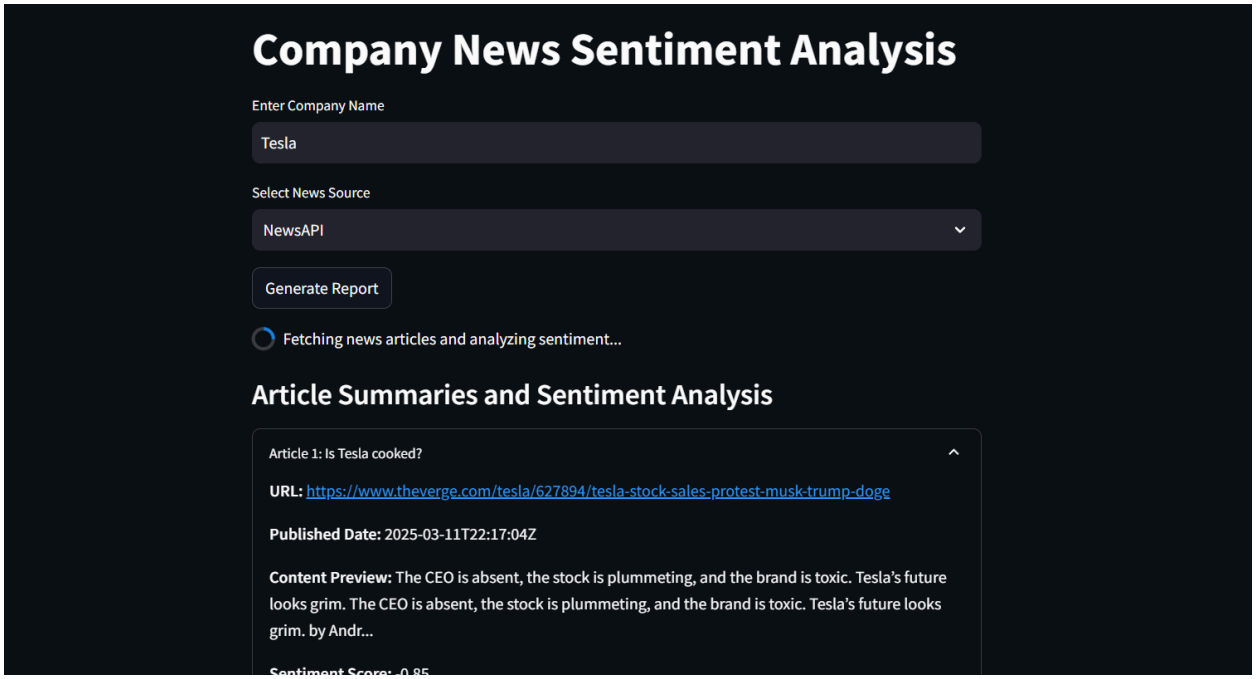
This project successfully fulfills the requirements by:

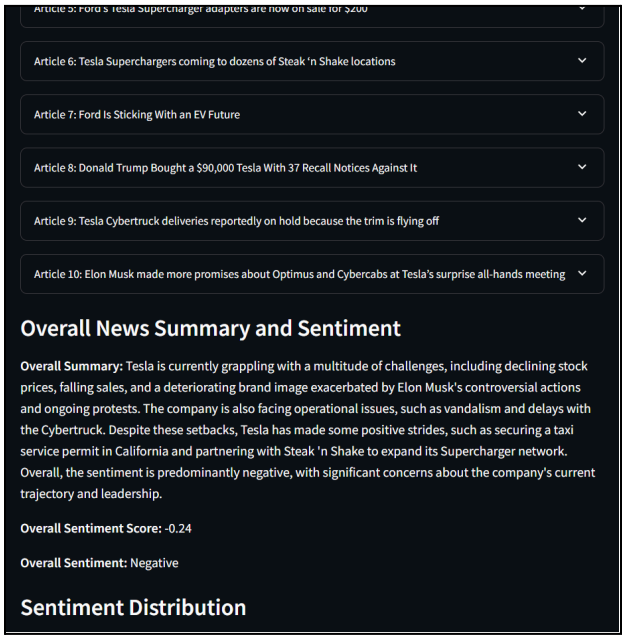
- Extracting news articles using NewsAPI and Bing News RSS.
- Optimizing the scraping process for dynamic websites.
- Performing sentiment analysis and summarization using the Groq API.
- Generating a Hindi TTS output using googletrans and gTTS.
- Providing a user-friendly interface using Streamlit.
- Gives the users the option to download the generated files and a report

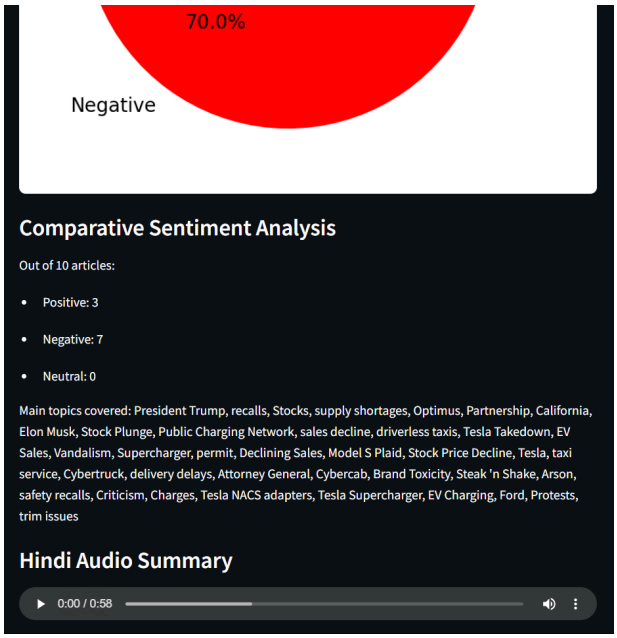
Some improvements that could be implemented are,

- A better TTS module with access to SOTA text-to-speech models/APIs.
- Optimizing the scraping process for dynamic websites.
- Adding support for more news sources with access to paid APIs

Deployment







The application is deployed on Hugging Face Spaces. It is accessible here.
It is also accessible on GitHub here.