Project Report: News Summarization and Text-to-Speech Application

Shikhar Pandey
Kalinga Institute of Industrial Technology, Bhubaneswar
pandeyshikhar07@gmail.com
+91 8830282004

March 23, 2025

Abstract

This project report describes the development of a web-based application that extracts news articles for a given company, performs sentiment analysis, and summarizes the content in English. Additionally, the application translates each article's concise summary into Hindi and generates a combined Hindi audio report using text-to-speech (TTS). The final solution includes a user-friendly interface built with Streamlit, API integration, and deployment on Hugging Face Spaces.

1 Introduction

The objective of this project is to build an application that:

- Extracts key details (title, summary, and metadata) from multiple news articles.
- Performs sentiment analysis to classify articles as positive, negative, or neutral.
- Summarizes the content to generate concise reports.
- Translates the English summaries into Hindi and generates a Hindi audio report using TTS.
- Provides a web-based user interface for easy interaction.

2 Project Links

- **GitHub Repository:** `https://github.com/pandeyshikhar18/News_Summarizer_TTS`
- **Hugging Face Spaces:** `https://huggingface.co/spaces/pandeyshikhar/news_summarizer_tts`
- **Video Demo:** `https://drive.google.com/file/d/1DB-nL0k_D5HgH-kf87Pze1r9ZbiNtFU1/view`

3 Objectives

1. **News Extraction:** Scrape at least 10 unique news articles using BeautifulSoup.
2. **Sentiment Analysis:** Classify each article's content into positive, negative, or neutral sentiment.
3. **Summarization:** Generate concise summaries for each article.
4. **Translation and TTS:** Translate each article's summary into Hindi and generate a combined Hindi audio report.
5. **User Interface and API:** Develop a clean, responsive web interface using Streamlit and expose functionality through APIs.
6. **Deployment:** Deploy the application on Hugging Face Spaces.

4 System Architecture and Design

The system is divided into several modules:

- **News Extraction Module:** Uses Python libraries `requests` and `BeautifulSoup` to scrape news articles.
- **NLP Processing Module:** Utilizes Hugging Face Transformers pipelines for sentiment analysis and summarization.
- **Translation Module:** Uses the Helsinki-NLP `opus-mt-en-hi` model to translate English summaries into Hindi.
- **Text-to-Speech Module:** Converts the combined Hindi summary into speech using gTTS.
- **User Interface:** Built with Streamlit, enabling users to input a company name and view the results.
- **API Module:** Provides endpoints (in `api.py`) for frontend-backend communication.

5 Implementation Details

5.1 News Extraction

A static list of news URLs is used to scrape article titles, summaries, and publication dates. The extraction logic uses BeautifulSoup to parse HTML content.

5.2 Sentiment Analysis and Summarization

The application uses Hugging Face pipelines:

- **Sentiment Analysis:** Derives sentiment from the article summary.
- **Summarization:** Summarizes the article content using the `sshleifer/distilbart-cnn-12-6` model.

5.3 Translation and TTS

Each article's concise summary (in English) is translated individually into Hindi. These Hindi summaries are then combined to generate a single Hindi text, which is passed to gTTS to produce the Hindi audio report.

5.4 User Interface

The user interface is implemented in Streamlit:

- The final English report is displayed along with detailed article information.
- A separate section presents the translated Hindi summaries.
- An audio player is provided to listen to the Hindi audio report.

6 Deployment and API

The application is deployed on Hugging Face Spaces for public testing. An API is provided (via `api.py`) to allow the frontend to communicate with the backend. API documentation is included in the repository's README.

7 Assumptions and Limitations

- **Assumptions:** The news URLs are accessible and contain the expected HTML structure. The translation and summarization models provide satisfactory output.
- **Limitations:** The translation quality depends on the structure of the English text. Some news sites may require additional handling due to dynamic content or paywalls.

8 Conclusion

This project successfully integrates news scraping, sentiment analysis, summarization, translation, and TTS into a cohesive application. The final product meets the assignment requirements by providing a structured report in English along with a translated Hindi summary and audio report. Future improvements could include model fine-tuning, GPU acceleration for faster inference, and enhanced error handling.

References

- **Hugging Face Transformers:** `https://huggingface.co/docs/transformers/`
- **gTTS:** `https://pypi.org/project/gTTS/`
- **BeautifulSoup:** `https://www.crummy.com/software/BeautifulSoup/bs4/doc/`