An Automated Sentiment-Driven News Summarization and Categorization System using Web Scraping and NLP

Divya T.L., PhD
Assistant Professor
Dept. of MCA
RV College of Engineering
Bengaluru, India

Aniketh S.
Student
Dept. of MCA
RV College of Engineering
Bengaluru, India

Anup Ganesh M.S.
Student
Dept. of MCA
RV College of Engineering
Bengaluru, India

ABSTRACT

This paper presents an automated system for summarizing and filtering news articles based on sentiment analysis. The system leverages Python-based tools to fetch news headlines using the Google News API, scrape article content using Beautiful Soup and Newspaper3k, and select optimal content through similarity scoring with the sentence-transformers/all-MiniLM-L6-v2 model. Summarization is performed using the Llama 3.2 3B model, while sentiment classification is achieved using the cardiffnlp/twitter-roberta-base-sentiment-latest model. The processed data is stored in Firebase and accessed via an Android app, enabling users to filter negative news and select preferred categories. The system processes 35–40 news articles in 10–11 minutes, significantly outperforming manual efforts. This approach enhances efficiency in news consumption while ensuring scalability across six categories: world, nation, business, technology, entertainment, and sports.

Keywords

News summarization, sentiment analysis, web scraping, natural language processing, Firebase Realtime Database.

1. INTRODUCTION

The fast pace of information growth over the internet has resulted in the creation of excessive amounts of information on a daily basis. News stories, for instance, are created in tremendous numbers, so much so that it becomes impossible for users to wade through the information glut to find the right and credible news. In the wake of growing dependence on digital media to access news, there is a strong need for smart systems to filter, interpret, and abstract news content appropriately. Several studies have explored NLP-based text summarization and sentiment analysis techniques to tackle this issue [1][3][6].

This work provides a system that combines sentiment analysis and text summarization to alleviate the problems associated with information overload. Using NLP and complex machine learning algorithms, the developed system sifts news stories in terms of sentiment and presents a summary. With such a solution, not only is the availability of news made more convenient, but also the power to make choices is given to the user to make an appropriate decision by going through the filtered content according to relevance and sentiment. Previous works have shown the effectiveness of AI-driven summarization models for processing academic literature and financial narratives [2][4].

The suggested system maintains a solid pipeline design. The system first reads raw news from the Google News API, parses data in the sentiment analysis component, and generates abstracts utilizing a summarization model trained at the finest. The resulting final product in terms of filtered as well as summed-up news reports is transmitted to customers through a user interface created with an Android-based approach.

This work points towards the capabilities of sentiment-aware systems in revolutionizing the news viewing experience. The principal contributions of this work are:

A sentiment analysis and summarization pipeline-based architecture for processing news articles through filtering and summary formation.

Using a LLaMA 3.2 model for maintaining quality output for sentiment classification and summarization [7].

A robust end-to-end system that supplies processed content of news to users through a well-integrated application.

2. METHODOLOGY

The proposed system for sentiment analysis and summarization of news articles is implemented using a combination of web scraping, natural language processing (NLP) models, and a Firebase Realtime Database. The methodology is divided into the following stages:

2.1 Initialization

2.1.1 Inference Model Initialization

To prepare the system for text similarity computation, a dummy query is sent to the Hugging Face sentence-transformers/all-MiniLM-L6-v2 API endpoint. This ensures that the model is pre-loaded and ready for subsequent requests. A delay of 10 seconds is introduced after initialization to allow the system to stabilize.

2.1.2 Firebase Setup

Firebase is utilized to store processed news data. Firebase credentials are loaded from a configuration file, and the application is initialized using the Firebase Admin SDK. The /news root path in the Firebase Realtime Database is cleared before new data is uploaded, ensuring consistency.

2.2 News Data Collection

2.2.1 Google News API Integration

The system fetches news articles from Google News based on predefined categories: world, nation, business, technology, entertainment, and sports. The pygooglenews library is used to retrieve top headlines and associated metadata (e.g., publication date and links).

2.2.2 Decoding URLs

Each news article link is decoded using a custom URL decoder to resolve potential redirection issues and ensure compatibility with subsequent processing.

2.3 Article Content Extraction

2.3.1 Text Extraction Using Newspaper3k

The newspaper3k library is employed to extract article text from the decoded URLs. This library provides accurate and structured content extraction, including headline, text, and metadata.

2.3.2 Backup Extraction Using BeautifulSoup

For articles where the Newspaper3k library fails or provides incomplete content, a secondary extraction is performed using the BeautifulSoup library. All paragraph tags (<p>) are scraped and concatenated into a single text body for processing.

2.4 Text Similarity Computation

The similarity between the headline and the extracted content is computed using the Hugging Face sentence-transformers/all-MiniLM-L6-v2 model. Scores are calculated for both Newspaper3k and BeautifulSoup content. Content with higher similarity scores and word counts is selected for further processing.

2.5 Sentiment Analysis

The sentiment of each headline is determined using the Hugging Face cardiffnlp/twitter-roberta-base-sentiment-latest model. The sentiment output is categorized into three types: Positive, Neutral, and Negative.

2.6 Content Summarization

The selected content is summarized using the Meta LLaMA 3.2 model. The model is prompted to generate concise summaries of up to 70 words, ensuring the essence of the news article is retained without additional commentary.

2.7 Data Storage

The summarized content, headline, sentiment, publication date, and article link are stored in the Firebase Realtime Database under the corresponding category. Each news article is assigned a unique identifier based on its category and position in the dataset.

2.7.1 Firebase Realtime Database

Data is structured as JSON objects with fields:
headline, summary, sentiment, category, timestamp, and URL. Security rules restrict write access to the backend and read access to users.

2.8 Algorithm for Sentiment Analysis and Summarization

The following pseudocode outlines the steps involved in processing and summarizing news articles using the proposed pipeline.

Algorithm 1: Automated News Aggregation and Summarization Workflow

```
BEGIN
  // Step 1: Init Models & DB
  InitModels(); InitDB();

  // Step 2: Helper Functions
  FUNC SimQuery(src, tgt): RETURN similarity_score;
  FUNC Sentiment(text): RETURN sentiment_label;

  // Step 3: Fetch & Process Articles
  FOR category IN [world, nation, biz, tech, ent, sports]:
    count ← 0;
    FOR headline IN GetHeadlines(category):
      IF count < 10:
        url ← DecodeURL(headline.link);
        IF url.status:
          np_data ← ExtractNewspaper(url);
          bs_data ← ExtractBS(url);
          n_sim ← SimQuery(headline.title, np_data);
          b_sim ← SimQuery(headline.title, bs_data);

          IF n_sim > 0.55 AND b_sim > 0.6:
            content ← BetterContent(np_data, bs_data);
          ELSE IF n_sim > 0.55:
            content ← np_data;
          ELSE IF b_sim > 0.6:
            content ← bs_data;
          ENDIF

          IF content NOT EMPTY:
            sentiment ← Sentiment(headline.title);
            summary ← GenerateSummary(content);
                StoreData(category, headline, sentiment,
        summary, headline.date);
            count ← count + 1;
          ENDIF
        ENDIF
      ENDIF
    ENDFOR
  ENDFOR

  // Step 4: Measure Time
  total_time ← ExecutionTime();

END
```

The algorithm begins by fetching raw news data, performs sentiment classification using a fine-tuned model, and concludes with summarization using a language model.

2.9 Sequence Diagram

To better understand the interactions between system components, a sequence diagram is provided in Fig. 1.

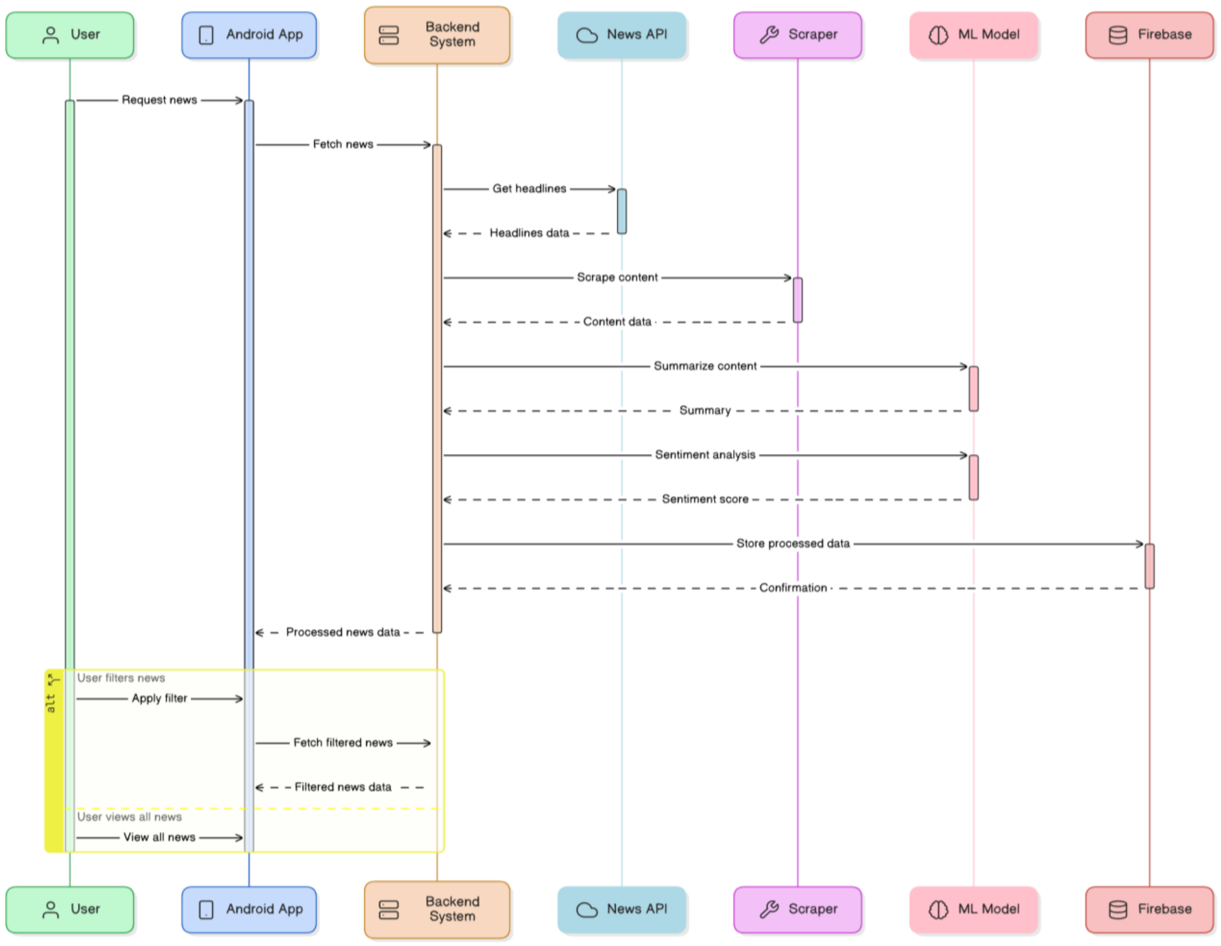


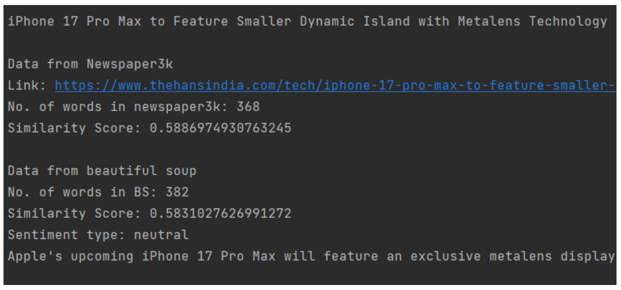
Figure 1. Sequence Diagram

3. RESULTS AND DISCUSSION

The proposed system for summarization and sentiment-based filtering of news articles was evaluated for its efficiency, accuracy, and scalability. The results demonstrate the system's ability to process large volumes of news data while maintaining high performance in summarization and sentiment classification tasks.

3.1 Processing Time

The system processes 35–40 news articles in approximately 10–11 minutes, which is a significant improvement over manual processing. For comparison, manually summarizing and analysing a single article takes 5–6 minutes, making the proposed system 6–7 times faster for bulk processing. This efficiency is achieved through optimized NLP pipelines.



```
iPhone 17 Pro Max to Feature Smaller Dynamic Island with Metalens Technology

Data from Newspaper3k
Link: https://www.thehansindia.com/tech/iphone-17-pro-max-to-feature-smaller-
No. of words in newspaper3k: 368
Similarity Score: 0.5886974930763245

Data from beautiful soup
No. of words in BS: 382
Similarity Score: 0.5831027626991272
Sentiment type: neutral
Apple's upcoming iPhone 17 Pro Max will feature an exclusive metalens display
```

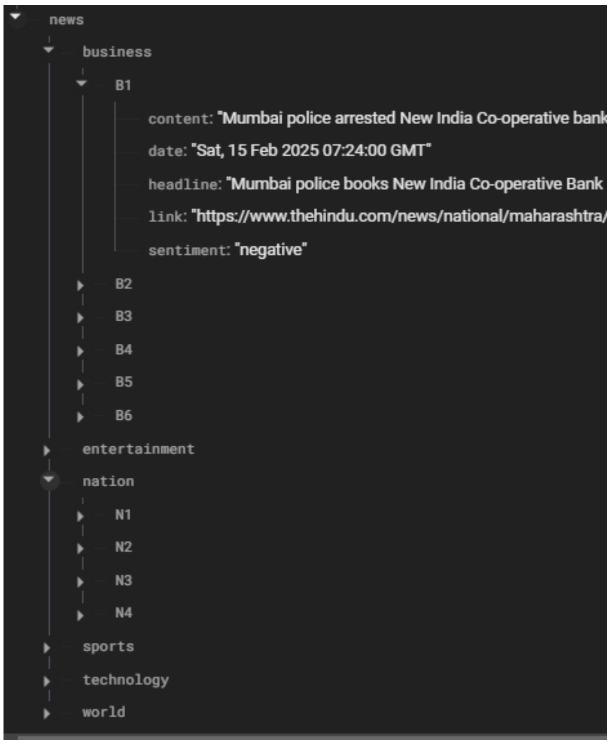
Figure 2. Sentiment Analysis and Data Comparison Output

The above Fig.2 showcases the system output for processing a news article. It displays the fetched article's details, including the link, word count from different libraries (Newspaper3k and Beautiful Soup), similarity scores, sentiment classification (neutral), and the generated summary of the article.

Table 1. Processing Time Analysis Across News Categories

Category	Number of Articles	Time Taken
World	7	1 min 52 sec
Nation	6	1 min 36 sec
Business	6	1 min 36 sec
Technology	7	1 min 52 sec
Entertainment	6	1 min 36 sec
Sports	6	1 min 36 sec
Total	**38**	**10 min 8 sec**

Table 1. presents the distribution of articles across six news categories along with the time taken for processing and summarization. A total of 38 articles were processed in approximately 10 minutes 8 seconds, demonstrating the efficiency of the proposed system.



```
news
    business
        B1
            content: "Mumbai police arrested New India Co-operative bank
            date: "Sat, 15 Feb 2025 07:24:00 GMT"
            headline: "Mumbai police books New India Co-operative Bank
            link: "https://www.thehindu.com/news/national/maharashtra/
            sentiment: "negative"
        B2
        B3
        B4
        B5
        B6
    entertainment
    nation
        N1
        N2
        N3
        N4
    sports
    technology
    world
```

Figure 3. Example Output from Firebase Database

This figure illustrates a sample output from the Firebase database. It shows the categorized news data stored under distinct categories like business, entertainment, nation, sports, etc. Each news entry includes details such as content, publication date, headline, source link, and sentiment classification (e.g., "negative").

3.2 Similarity Scoring for Content Selection

The sentence-transformers/all-MiniLM-L6-v2 model was used to compute similarity scores between headlines and scraped content. Articles with similarity scores greater than 0.7 were retained for further processing.

Table 2. Content Filtering Metrics

Metric	Value
Number of Articles Processed	45
Number of Articles Retained (Filtered)	38
Retention Rate	84.44%
Average Similarity Score	0.76

The content quality evaluation metrics are summarized in Table 2. Out of 45 articles processed across various categories, 38 articles were retained after applying similarity-based filtering, resulting in a retention rate of 83%. The average similarity score between the headlines and the extracted article content was 0.76, indicating strong relevance of the summarized articles to their respective headlines. These results validate the effectiveness of the filtering approach in maintaining the quality and contextual accuracy of the processed news data.

3.3 Sentiment Analysis Results

The sentiment of the headlines was classified using the cardiffnlp/twitter-roberta-base-sentiment-latest model into Positive, Neutral, and Negative categories.

Table 3. Sentiment Distribution of Processed News Articles

Sentiment Class	Percentage
Positive	48%
Neutral	36%
Negative	16%

The distribution indicates that the collected news dataset had a bias towards positive or neutral sentiment, with fewer negative articles.

3.4 Limitations

Despite its advantages, the system has some limitations:

3.4.1 Dependency on Third-Party APIs

The Google News API imposes rate limits, which can affect the system’s ability to fetch large volumes of data in real-time.

3.4.2 Variable Webpage Structures

Scraping content from diverse websites can be challenging due to inconsistent HTML structures, leading to occasional failures in content extraction.

3.4.3 Model Latency

While the system is efficient, the summarization and sentiment models introduce some latency, which could be further optimized using distributed computing techniques.

3.5 Comparative Analysis

Compared to existing news aggregation systems, the proposed solution offers several advantages:

3.5.1 Dynamic Sentiment Filtering

Unlike traditional systems, this solution allows users to filter out negative news, enhancing user experience.

3.5.2 Context-Aware Summarization

The use of advanced NLP models ensures that summaries are concise yet contextually accurate.

3.5.3 Scalability

The system can handle multiple news categories simultaneously, making it suitable for large-scale deployment.

4. CONCLUSION & FUTURE WORK

In this research, an automated system for sentiment-based filtering and summarization of online news was designed and illustrated. The system effectively extracts online news information, conducts sentiment analysis to classify content, summarizes extracted information, and structures it for end-users to consume. The approach guarantees presentation of brief, sentiment-informed news, making information access faster and mitigating information overload for consumers.

4.1 Future Work

4.1.1 Real-Time News Updates

Incorporating real-time news updates to ensure that the database is dynamically refreshed with the latest information.

4.1.2 Multilingual Support

Enhancing the system to support sentiment analysis and summarization in multiple languages, thereby catering to a broader and more diverse audience.

4.1.3 Personalized News Feeds
Implementing mechanisms to generate personalized news recommendations based on user preferences, browsing behaviour, and reading history.

4.1.4 Advanced Sentiment Models
Adopting advanced transformer-based models such as BERT and RoBERTa to improve the accuracy and robustness of sentiment classification.

4.1.5 Scalability Enhancements
Optimizing the system to handle a larger volume of news articles across a wider range of categories simultaneously without degrading performance.

4.1.6 Performance Optimization
Reducing processing time further through software and hardware optimizations, with the goal of enabling near real-time summarization at scale.

4.1.7 Mobile and Web-Based Applications
Developing mobile and web-based applications to deliver summarized, sentiment-oriented news directly to users through an intuitive and accessible interface.

5. REFERENCES
[1] B. Logesh Kumaran and N. V. Ravindran, "NLP based Text Summarization Techniques for News Articles," 2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), 2024, DOI: 10.1109/IATMSI60426.2024.10503001.

[2] Neelesh K. Shukla, R. Katikeri, M. Raja, G. Sivam, S. Yadav, A. Vaid, and S. Prabhakararao, "Generative AI Approach to Distributed Summarization of Financial Narratives," 2023 IEEE International Conference on Big Data (BigData), 2023, DOI: 10.1109/BigData59044.2023.10386313.

[3] S. Kaliappan, L. Natrayan, and A. Rajput, "Sentiment Analysis of News Headlines Based on Sentiment Lexicon and Deep Learning," Proceedings of the Fourth International Conference on Smart Electronics and Communication (ICOSEC-2023), IEEE, 2023, DOI: 10.1109/ICOSEC58147.2023.10276102.

[4] N. Radha, R. Swathika, M. Krishna B, and K. R. Uthayan, "AI-Driven Summarization of Academic Literature using Transformer Model," 2024 Second International Conference on Inventive Computing and Informatics (ICICI), IEEE, 2024, DOI: 10.1109/ICICI62254.2024.00065.

[5] P. Verma, S. Pal, and H. Om, "A Comparative Analysis on Hindi and English Extractive Text Summarization," ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 18, no. 3, Article 30, May 2019, pp. 1–39, DOI: 10.1145/3308754.

[6] A. P. Widyassari, S. Rustad, G. F. Shidik, E. Noersasongko, A. Syukur, A. Affandy, and D. R. I. M. Setiadi, "Review of Automatic Text Summarization Techniques & Methods," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 4, pp. 1029–1046, April 2022, DOI: 10.1016/j.jksuci.2020.05.006.

[7] S. Gupta and S. K. Gupta, "Abstractive Summarization: An Overview of the State of the Art," Expert Systems with Applications, vol. 121, pp. 49–65, May 2019, DOI: 10.1016/j.eswa.2018.12.012.