NewsSnips

Global News Article Summarization And Classification

A Synopsis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

BACHELOR OF TECHNOLOGY

in

Department of Artificial Intelligence & Machine Learning

by

Kajal Metkar[A39]
Mohd Talha Ansari[A47]

G H Raisoni College of Engineering and Management

Wagholi, Pune

Department of Artificial Intelligence and Artificial Intelligence & Machine Learning

**(An Empowered Autonomous Institute (NAAC A+ Grade)
Affiliated to SPPU, Pune)**

June, 2024

Introduction

In today's fast-paced digital world, the sheer volume of news being published every minute across various online platforms poses a significant challenge for users seeking timely and relevant information. With countless articles available on topics ranging from global events to niche interests, users often struggle to find content that is both concise and tailored to their preferences. The constant influx of news leads to information overload, where users are either bombarded with too much irrelevant content or miss out on key updates due to the difficulty of sorting through the massive volume of available information.

The project "Global News Summarization and Categorization" addresses this issue by leveraging advanced Natural Language Processing (NLP) techniques. These techniques allow the system to automatically filter, categorize, and summarize news articles from various sources, reducing the need for users to manually sift through irrelevant or overly long content. By categorizing articles into predefined topics (such as politics, technology, and sports) and generating concise summaries, the system ensures that users can quickly access the most relevant news without being overwhelmed by excessive or redundant information.

Objectives

- Develop an efficient system to process and manage global news articles from multiple sources.
- Implement NLP techniques to classify and categorize news articles by topics.
- Generate concise, accurate news summaries for quick consumption.
- Tailor news content based on user preferences and interests.
- Provide a streamlined, user-friendly experience for accessing relevant and focused news.

Literature Review

Despite advancements, existing platforms like Google News and Apple News focus primarily on personalization by recommending articles based on user behavior but often fall short in providing efficient summarization and multi-label classification. While these platforms can suggest relevant content, they lack the ability to condense long articles into concise summaries and struggle with categorizing news articles that belong to multiple topics, limiting their efficiency in handling vast amounts of news data comprehensively. This gap underscores the need for more advanced systems capable of both summarizing content effectively and supporting multi-label classification for a more refined user experience.

Methodology

The project "Global News Summarization and Categorization"'s NLP-driven approach not only improves the efficiency of news consumption but also personalizes the experience. By tailoring content to individual interests and providing clear, focused summaries, users can stay informed about topics that matter to them, all while avoiding the pitfalls of information overload in today's data-saturated digital environment.

1. **Data Collection and Preparation**

Gather news articles from diverse sources, or predfined news Data set

Preprocess data to standardize format and remove noise.

2. **NLP Model Development**

Build a robust NLP model for article filtering.

Implement text processing techniques like tokenization.

3. **News Filtering**

Filter articles based on user queries.

4. **News Categorization**

classifying the news using algo like word search / kmeans classifier or NLP preprocessing

5. **News Searching**

using algo linear search / regular expression / suffix tree

6.**News summarization**

Summarize articles using techniques like Cosine Similarities , TF-IDF.

Tools and Technologies

- **Frontend:** HTML, CSS and Javascript, React.js for the graphical user interface.
- **Backend:** Using Javascript ,Python for backend logic
- **Flask :** Data fetching,API Development,Efficient handling of data
- **Data Preprocessing:** Library of Pandas
- **Natural Language Processing (NLP):** Filtering,Categorization,Summarization,tokenization, stemming
- **Summarization Algorithm:** TF-IDF (Term Frequency-Inverse Document Frequency),Cosine Similarity,Sentence Ranking

Project Plan and Timeline

The project was completed over a six-month period, divided into the following phases:

1. Phase 1 - Requirement Analysis and Design (0.5 month)

- Activities:

· **Scope Definition**: Clearly outline the system's purpose, objectives, and user requirements.

· **Architecture Planning**: Design the system's architecture, identifying the main components (frontend, backend, database) and their interactions.

· **Functional Specifications**: Define the core functionalities such as news collection, categorization, summarization, and user interface features.

· **Non-functional Requirements**: Identify performance metrics, scalability, security, and usability requirements.

· **Technology Stack**: Decide on the technologies to be used, such as React.js for the frontend, Flask for the backend, and NLP libraries for text processing.

2. Phase 2 - Frontend Development (0.5 months)

- Activities:

· **User Interface (UI) Design**: Create a responsive and intuitive UI using React.js, focusing on simplicity and ease of navigation.

· **Component Development**: Break down the interface into reusable components (e.g., article list, category filters, search bar).

· **State Management**: Implement efficient state management using tools like Redux or React Context to handle data across the application.

· **User Interaction Features**: Add features such as search functionality, real-time filtering, and personalized content recommendations based on user preferences.

· **Responsive Design**: Ensure the application is fully responsive, adapting seamlessly to different screen sizes and devices (desktop, mobile, tablet).

3. Phase 3 - Backend Development (0.5 month)

- Activities:

· **API Design**: Set up RESTful APIs using Flask to manage requests from the frontend and interact with the backend.

· **Data Collection**: Implement API endpoints for fetching news articles from various sources.

· **Data Storage**: Design and implement a database structure to store news articles, user preferences, and categories (using tools like MongoDB or SQL databases).

· **Backend Logic**: Write logic to process data, handle user authentication, manage real-time news updates, and serve summarized/categorized content.

· **Security**: Incorporate security measures, such as encryption and secure data handling practices, to protect user data and ensure safe API interactions.

4. Phase 4 - NLP Techniques (1 month)

- Activities.

· **Text Preprocessing**: Implement NLP techniques such as tokenization, stopword removal, and stemming/lemmatization to prepare articles for analysis.

· **Summarization Models**: Use extractive summarization methods like **TF-IDF** and **Cosine Similarity** to generate concise summaries from long articles.

· **Text Classification**: Apply machine learning models (e.g., Naive Bayes, SVM) to categorize articles into predefined topics like Politics, Technology, Sports, etc.

· **Personalization Algorithms**: Develop algorithms to analyze user behavior and preferences, allowing for personalized article recommendations and filtering.

· **Multi-label Classification**: Ensure articles that belong to more than one category are properly tagged for better content organization.

5. Phase 5 - Integration and Testing (1 month)

- Activities:

· **Frontend and Backend Integration**: Merge the frontend and backend, ensuring smooth communication between React.js and Flask APIs.

· **Unit Testing**: Test individual components of both the frontend and backend to ensure each module works correctly in isolation.

· **Integration Testing**: Test the interaction between the frontend, backend, and database to ensure data is being processed and displayed correctly.

· **User Acceptance Testing (UAT)**: Gather user feedback on the overall system, testing its ease of use, response times, and content relevance.

· **Performance Testing**: Evaluate the system's scalability, ensuring it can handle large volumes of data and user requests efficiently.

· **Security Testing**: Conduct tests to identify any vulnerabilities in the system, ensuring that APIs and data handling processes are secure.

6. Phase 6 - Final Adjustments and Deployment (0.5 month)

- Activities:

· **Bug Fixes**: Address any bugs or issues identified during testing, refining the code and functionality to ensure a smooth user experience.

· **Performance Optimization**: Fine-tune the system to improve load times, responsiveness, and backend efficiency.

· **User Feedback Integration**: Implement any additional features or tweaks based on user feedback from testing phases.

· **UI/UX Enhancements**: Make final adjustments to the user interface, improving the visual design, layout, and overall user experience.

· **Documentation**: Ensure all components are fully documented, including code, APIs, and user manuals for easy reference.

· **Deployment**: Prepare the system for launch, setting up deployment pipelines and ensuring the platform is ready for public use.

Expected Outcomes

The project is designed to deliver several key outcomes that aim to significantly improve the news consumption experience for users. These outcomes ensure that users can efficiently access relevant information without being overwhelmed by the sheer volume of global news content. Here are the expected results in more detail:

Personalized News:
The system provides news content tailored to user interests and preferences by analyzing browsing history and engagement. Powered by NLP and machine learning, the recommendation engine refines user preferences over time for more relevant content.

Quick Summaries:
Delivers concise summaries of lengthy articles using techniques like TF-IDF and Cosine Similarity, helping users quickly grasp key points without reading the full text. This is ideal for users with limited time.

Categorized News:
Articles are categorized into sections like Technology, Business, and Sports using machine learning. Multi-label classification ensures that articles spanning multiple topics are accurately tagged for better navigation.

User-Friendly Interface:
The platform features a responsive, easy-to-navigate interface built with React.js, allowing users to browse, search, and personalize their news feed seamlessly. Real-time updates and customizable settings enhance the experience across devices.

References

1.Naïve Bayes Classifier for Classification
Authors: Aji Wibawa, Ahmad Chandra Kurniawan, Della Murbarani, Prawidya Murti, Risky Perdana Adiperkasa
Published on: International Journal of Recent Contributions from Engineering Science & IT (iJES)
Year: June 2019

2.Implementation of TF-IDF and Cosine Similarity Algorithms for Classification of Documents Based on Abstract Scientific Journals
Authors: Paska Marto Hasugian, Jonson Manurung Logaraz, Uzitha Ram
Published on: JURNAL INFOKUM
Year: June 2021

3.Analysis of Component Libraries for React JS
Authors: Mukthapuram Praneeth Reddy
Published on: IARJSET
Year: June 2021

4.Title: "Text Summarization Techniques: A Review"
Authors: Priyanka Dey, M. A. Awan
Published on: International Journal of Computer Applications -- 2020

5.Title: "The Effectiveness of TF-IDF and Cosine Similarity in Text Classification"
Authors: Shereen Albitar, Sébastien Fournier, Bernard Espinasse
Published on: Journal of Data Science 2018

_____ _____

GUIDE **HOD**