IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# Personalized News Aggregator with Sentiment Analysis

*Kunal Tanwar*
*Student*
*Dept. of Computer Engineering*
*Poornima institute of engineering & Technology*
*Jaipur, India*

*Harsh Saini*
*Student*
*Dept. of Computer Engineering*
*Poornima institute of engineering & Technology*
*Jaipur, India*

*Kartik Bhagwani*
*Student*
*Dept. of Computer Engineering*
*Poornima institute of engineering & Technology*
*Jaipur, India*

**Abstract:**

In the era of information overload, accessing relevant and meaningful news has become increasingly challenging for users.This research introduces a Personalized News Aggregator with Sentiment Analysis-a system that is designed to provide a user-centric content of news based on interests and preferences. This system aggregates news from diverse sources and then uses NLP to analyze the sentiment of news articles to help users better understand the emotional tone and the context behind the content being consumed.

Machine learning models of sentiment analysis will be used, combined with recommendation algorithms for user-specific personalization to ensure that the platform presents an issue a refined, engaging, and user-focused news experience. The solution proposed here remedies the shortcomings of the classic news mediums by providing filtering, keyword search, and sentiment-based categorization. This paper discusses its technical implementation, including data collection, preprocessing, model selection, and web-based deployment, and highlights its potential utility in improving information accessibility and user satisfaction.

**Introduction:**

The exponential growth of digital media has resulted in an overwhelming amount of news and information, which is hard for users to sift through and consume in a time-efficient manner. Traditional news platforms lack personalization, so users often stumble upon articles that do not interest or appeal to them. Most of the time, the emotional impact of news content remains unaddressed, as it plays a very significant role in engaging and influencing users' perceptions.

This project aims to resolve the challenges with personalization and incorporate both elements within a single, user-centered platform. It aggregates news articles from various sources and analyzes their emotional tone through natural language processing techniques to ensure that users receive the content of interest while gaining insights into the sentiment of the articles.

The platform offers features such as category filtering, keyword-based search, and sentiment-based sorting, providing users with a dynamic and engaging news-reading experience. In addition, it uses recommendation algorithms that continually redefine the personalization based on user interactions. This paper researches the development and implementation of the proposed system, underlining its probable effects on user satisfaction as well as amelioration in access to meaningful information.

This research contributes to the growing field of AI-driven personalization and can demonstrate the value of sentiment-aware systems in fostering more informed and emotionally aware consumption of digital news.

**Literature Review:**

The rise of artificial intelligence and natural language processing has transformed the way digital content is processed and delivered. Personalized content recommendation systems and sentiment analysis have been two of the major areas researched widely in recent years. Existing research and technologies relevant to the development of a Personalized News Aggregator with Sentiment Analysis are discussed in the following literature review.

I. **News Aggregation Systems**: News aggregation systems aggregate articles of various sources for presenting to the user in a central platform. Researches, such as [Li Gao;Yinghui Liu;2023 International Conference on Computer Science and Automation Technology (CSAT)], keep emphasizing the point that the APIs, RSS feeds, and web scraping techniques are fundamental to ensure a continuous stream of varied and timely information. However, traditional aggregation systems have failed to meet the users' demand for personalization.

II. **Personalization in News Delivery:** Personalization within news channels has been a subject of thorough studies, whereby recommendation algorithms such as collaborative filtering, content-based filtering, and hybrid models are becoming popular. For instance, studies by [Hiroaki Takatsu; Mayu Okuda; Yoichi Matsuyama; Hiroshi Honda; Shinya Fujie; Tetsunori Kobayashi; 2021 IEEE Spoken Language Technology Workshop (SLT) ] indicate that user profiling based on reading history and preferences can significantly enhance engagement and satisfaction. However, most platforms that currently exist rely only on clickstream data and never take into account sentiment or emotional resonance in making their recommendations.

III. **Sentiment Analysis in News Platforms:** Sentiment analysis is the subset of Natural Language Processing (NLP) that helps to find the emotional tone emanating from a piece of text. Tools such as VADER, TextBlob, and transformer-based models like BERT have high accuracy in determining sentiments across various domains. For example, [J. Wilson, S. E. Hosseini and S. Pervez, "Identification of Fake News in Social Media Using Sentimental Analysis," 2023 IEEE Industrial Electronics and Applications Conference (IEACon), Penang, Malaysia, 2023, pp. 220-224, doi: 10.1109/IEACon57683.2023.10370300.] The studies explore the application of sentiment analysis to news content and emphasize how it is applied for providing audience-related responses and reactions. However, how to apply such sentiment analysis in real-time personalization is still a challenge.

IV. **Challenges in Integrating Personalization with Sentiment Analysis:** Combining personalization and sentiment analysis into one system is not yet explored well. [Research by: S. Nabil, J. Elbouhdidi and M. Yassin Chkouri, "Recommendation System Based on Data Analysis-Application on Tweets Sentiment Analysis," 2018 IEEE 5th International Congress on Information Science and Technology (CiSt), Marrakech, Morocco, 2018, pp. 155-160, doi: 10.1109/CIST.2018.8596418.] observes that though personalized systems are sensitive to the needs of the users and improve user satisfaction, the introduction of sentiment analysis adds computational complexity and calls for improved models to capture the intricacies of the natural language. Additionally, ensuring the diversity of news content while maintaining relevance remains a critical concern.

V. **Technological Frameworks and Tools:** Recent studies have highlighted the role of machine learning frameworks, cloud technologies, and front-end development tools in building scalable and efficient personalized systems. For example, [R. Dhanvardini, P. Martina, R. Vijay, R. Amirtharajan and P. Pravinkumar, "Development and integration of dApp with blockchain smart contract Truffle Framework for user interactive applications," 2023 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2023, pp. 1-6, doi: 10.1109/ICCCI56745.2023.10128406.] talks about integrating React.js for the development of the user interface and Flask for any of the necessary backend services in the personalized platforms. Such studies act as a base reference for the development of the proposed system.

This literature review establishes the urgency in integrating sentiment analysis to new personalized news, comparing the current gaps that exist in today's systems. The proposed project is a consolidation of these studies to provide a holistic solution that not only personalizes but also brings a sense of emotional implication to its users, through news, thereby heightening user experience and engagement.

**Problem Statement:**

The rapid growth of digital news platforms has resulted in an overwhelming amount of information being made available to users. However, existing news aggregation systems face several critical challenges:

I. **Lack of Personalization:** Many news platforms fail to deliver content tailored to individual user preferences, often bombarding users with irrelevant or redundant information. This leads to user disengagement and inefficiency in news consumption.

II. **Emotional Impact of News:** The emotional tone of news articles significantly influences users' perception and decision-making. Current systems do not integrate sentiment analysis, leaving users unable to discern the emotional context of the news they consume.

III. **Information Overload:** The sheer volume of unfiltered information can overwhelm users, making it difficult for them to focus on what truly matters to them.

IV. **Absence of Unified Experience:** Users often need to navigate multiple platforms to access diverse news sources, lacking a centralized system that aggregates, personalizes, and categorizes news content effectively.

Given these challenges, there is a pressing need for an intelligent system that can efficiently filter, personalize, and provide sentiment-based insights into news content, enhancing user satisfaction and engagement. This project aims to address these gaps by developing a Personalized News Aggregator with Sentiment Analysis, offering users a unified platform for tailored, sentiment-aware news consumption.

**Methodology :**

The 'Personalized News Aggregator with Sentiment Analysis' methodology is centered on a methodical approach to the creation, development, and implementation of an AI-powered news platform. The procedure combines sentiment analysis, data preparation, news aggregation, and user personalization. A thorough explanation of the methodology is provided below:

**1. Data Collection and Aggregation:**

I. Sources: News data is collected from APIs (e.g., NewsAPI, Google News API) and RSS feeds of various reputable platforms.

II. Content Types: Includes headlines, articles, publication dates, and metadata such as author and source.

III. Frequency: The aggregation process runs at regular intervals to ensure updated and relevant news delivery.

**2. Data Preprocessing:**

I. Cleaning: Remove unnecessary HTML tags, special characters, and irrelevant text.
II. Tokenization: Split the content into meaningful components like sentences or words.
III. Normalization: Convert text to lowercase, handle spelling corrections, and remove stopwords for improved processing.
IV. Language Detection: Filter out non-English articles (if applicable) for uniform analysis.

**3. Sentiment Analysis:**

I. Model Selection:

a) Pre-trained models like VADER or transformer-based models like BERT for sentiment scoring.
b) Categorize articles into positive, negative, or neutral sentiment.

II. Implementation:

a) Sentiment scores are calculated based on keywords, phrases, and overall tone.
b) Results are stored in the database for quick access during personalization.

**4. Personalization Engine:**

I. User Profiling:

a) Create user profiles based on interests, past interactions, and explicit preferences (e.g., categories like technology, sports).

II. Recommendation System:

a) Use collaborative filtering or content-based filtering algorithms to deliver personalized news feeds.
b) Combine user preferences with sentiment insights for more refined recommendations.

**5. Web Development:**

I. Frontend:

a) A user-friendly interface designed using React.js or similar frameworks.
b) Features include keyword search, sentiment-based filtering, and category-based news display.

II. Backend:

a) Developed using Flask/Django for managing API calls, user data, and personalization logic.
b) Secure data storage in a relational database like PostgreSQL.

**6. Integration of Features:**

I. Search and Filtering: Allow users to filter articles by keywords, sentiment, and publication date.
II. Sentiment Indicators: Display sentiment scores or labels alongside articles to enhance user insights.
III. Bookmarks and History: Enable users to save articles and view their reading history.

**7. Testing and Validation:**

I. Unit Testing: Test individual modules such as the sentiment analyzer, API integration, and recommendation algorithms.
II. End-to-End Testing: Ensure smooth functionality of the entire system, including data flow from the backend to the frontend.
III. User Feedback: Gather feedback from a small group of test users to refine the platform.

**8. Deployment:**

I. Cloud Deployment: Host the system on platforms like AWS or Azure for scalability and reliability.
II. Continuous Updates: Regularly update the system to incorporate new features, address bugs, and enhance accuracy.

This methodology ensures a systematic and efficient development process for the 'Personalized News Aggregator with Sentiment Analysis', focusing on delivering a feature-rich and user-friendly platform.

**Implementation**

The Personalized News Aggregator with Sentiment Analysis project requires the seamless integration of several different parts. The procedure entails creating a frontend for user interaction and a backend for data processing, aggregation, and sentiment analysis. A thorough, step-by-step breakdown of the implementation procedure is provided below:

**1.** Backend Development:

I. News Aggregation:

a) APIs and Feeds: Integrate with APIs like NewsAPI or Google News API to fetch real-time articles.
b) Web Scraping: Use libraries like BeautifulSoup or Scrapy for additional data sources that lack APIs.
c) Scheduling: Automate data collection using task schedulers like Celery or Cron jobs to ensure updated news at regular intervals.

II. Data Storage:

a) Database Design:

i. Use a relational database like PostgreSQL to store articles, user profiles, and sentiment scores.
ii. Schema includes tables for articles, users, preferences, and interaction history.

b) Scalability: Opt for cloud database solutions (e.g., AWS RDS or Azure SQL) for handling large datasets.

III. Sentiment Analysis:

a) Preprocessing: Clean and normalize the text using libraries like NLTK or SpaCy.

i. Model Implementation: Integrate pre-trained sentiment analysis models such as VADER for simplicity or transformer-based models like BERT for higher accuracy.
ii. Categorize articles as positive, negative, or neutral.

b) Optimization: Fine-tune the sentiment analysis model using domain-specific datasets for better relevance to news articles.

IV. Recommendation Engine:

a) Algorithm Design:

i. Implement content-based filtering using user preferences and sentiment scores.
ii. Augment with collaborative filtering for cross-user recommendations.

b) Personalization: Continuously refine recommendations based on user interactions, such as reading history and likes/dislikes.

**2. Frontend Development:**

I. User Interface (UI):

a) Develop an intuitive interface using React.js or Angular for dynamic and responsive design.
b) Key Features:
c) A personalized feed displaying articles by category, sentiment, and relevance.
d) Sentiment indicators (e.g., positive/negative icons) alongside articles.
e) Search bar for keyword-based filtering and advanced filters for sentiment or publication date.

II. User Authentication:

a) Implement secure login and registration using JWT (JSON Web Tokens) or OAuth2.
b) Allow users to customize their preferences and manage bookmarks.

III. Integration with Backend:

a) Use RESTful APIs or GraphQL for seamless communication between the frontend and backend.
b) Ensure real-time data updates using WebSocket or polling mechanisms.

**3. Deployment:**

I. Hosting:

a) Host the application on a cloud platform like AWS, Azure, or Google Cloud for scalability.
b) Use services like AWS Elastic Beanstalk or Azure App Service for managing the backend.

II. Continuous Integration/Continuous Deployment (CI/CD):

a) Set up CI/CD pipelines using tools like GitHub Actions or Jenkins for automatic builds, testing, and deployment.

III. Performance Monitoring:

a) Monitor system performance using tools like New Relic or AWS CloudWatch to identify bottlenecks and optimize response times.

**4. Testing and Validation:**

I. Module Testing: Test each backend module, such as news aggregation, sentiment analysis, and recommendation algorithms, for accuracy and reliability.

II. Integration Testing: Ensure smooth interaction between the frontend and backend APIs.

III. User Testing: Conduct usability testing with potential users to gather feedback on personalization and sentiment analysis features.

**5. Maintenance and Updates:**

I. Regularly update the database with new sources and articles.

II. Incorporate user feedback to refine the recommendation engine and improve UI/UX.

III. Periodically retrain sentiment models to adapt to evolving language patterns and news trends.

By following this implementation strategy, the 'Personalized News Aggregator with Sentiment Analysis' ensures a robust, user-friendly platform that delivers personalized, sentiment-aware news content to its users.

**Results and Discussion:**

The 'Personalized News Aggregator with Sentiment Analysis' had significant results and insights after development and implementation. Here follows a comprehensive discussion on the outcome:

I. Functionality of the Aggregator:

a) It successfully aggregated news articles from diverse sources via APIs and RSS feeds, thus delivering complete and latest information.

b) Efficient preprocessing of the data allowed the easy processing of large-size datasets and, subsequently ensured clean, structured data for the analysis.

II. Performance of Sentiment Analysis:

a) The sentiment analysis model showed a good precision in classifying articles as *positive*, *negative, or *neutral*.
b) Transformer-based models, such as BERT, proved superior than traditional methods (for instance, VADER), especially in handling intricate and nuanced text like that found in articles with mixed sentiments.
c) Real-time sentiment tagging gave users an intuitive feel for the emotional tone of articles, enhancing engagement and decision-making.

III. Efficiency of Personalization:

a) The personalization recommendation algorithm was efficient in rendering user preferences, reading history, and sentiment insights to personalize news feeds.
b) Collaborative and content-based filtering algorithms met a higher user satisfaction rate with respect to delivering highly relevant and diversified content.
c) Users responded positively to features like category-based filtering and sentiment-specific sorting, reflecting the platform's usability and relevance.

IV. User Experience:

a) The frontend interface, developed using React.js, was well-received for its simplicity, responsiveness, and ease of navigation.
b) Features such as keyword-based search, sentiment indicators, and article bookmarking were frequently utilized, showcasing their value to users.
c) Test users' feedback showed that it decreases information overload and improves the news consumption experience, in general.

V. Challenges Faced:

a) -Data Diversity: In aggregating information from such diverse sources and maintaining the quality and relevance required a good extent of data validation.
b) -Model Efficiency: Real-time analysis of a large dataset was computationally intensive; hence, optimization of the model improved its performance for fast processing.
c) -Personalization vs. Diversity: The biggest challenge was maintaining a level of diversity in personalized recommendations without creating an echo chamber.

VI. Comparison with Existing Systems:

a) -This system compared to any other news aggregator did not have conventional personalization nor conventional sentiment analysis, giving more engagement and insight for the user.
b) Inclusion of sentiment insights increases the trust and emotional affinity of the users toward the platform.

VII. Future Implications:

a) The scalability and modular design of the system will allow for easy integration with voice assistants and mobile applications, making it accessible to a wider audience.
b) Continuous improvement in sentiment models and recommendation algorithms will further improve user satisfaction.
c) By overcoming the identified problems, the platform will be able to provide a benchmark for AI-driven news aggregation systems.

**Challenges and Limitations**

**1. Challenges:**

I. Quality and Diversity of Data: Aggregating trustworthy, unbiased, and diverse news sources was challenging because data formats were inconsistent, and quality standards varied widely.
II. Scalability: Dealing with massive volumes of real-time data without compromising performance at scale required considerable optimization in the backend processes.
III. Sentiment Accuracy: Nuances in words, like sarcasm or ambiguous wording, occasionally led to incorrect sentiment classification.
IV. Personalization Balance: Tailing recommendations with not entering echo chambers or limiting the exposure to more diverse content was hard.

**2. Limitations:**

I. Language Support: The system mainly focused on offering content only in the English language and limiting access for other users.
II. Real-Time Processing: While being optimized, sometimes the real-time sentiment analysis and recommendations were delayed at peak usage.
III. API Dependency: The functionality of this application is API-dependent on third-party APIs for news aggregation, which could face rate-limiting or service disruptions risks at any time.

IV. Bias in Recommendations: Personalization algorithms might amplify biases inherent in the training data or user behavior inadvertently.

These challenges and limitations point to possible venues for improvement and iterative development to make the system more robust and accessible.

**Conclusion:**

The 'Personalized News Aggregator with Sentiment Analysis' embraces this new trend by fulfilling growing needs for personalized, emotionally responsive news delivery in this information-rich era. Advanced sentiment analysis, along with AI-driven recommendation systems, was used to present a light-hearted experience to the user in this platform.

The experiment showed how a mix of content aggregation, user profiling, and machine learning can promote improved news consumption efficiency. Some of the most salient features, including sentiment tagging, category filtering, and relevance-based recommendations, improved user satisfaction and usability.

Challenges: Maintenance of data quality, real-time performance, and ability to avoid recommendation biases were a few of the impediments, but modular architecture and flexibility will help create future improvements.

In conclusion, the platform is a huge step forward in personalized content delivery and exemplifies the potential for change brought about by AI in media use. With some refinement, it can develop into a useful tool for informed and emotionally astute news consumption, with a sharp move towards new standards for intelligent news aggregation systems.

**References:**

[1] Chan Wei Chang and Owen Noel Newton Fernando, "Twittener: An Aggregated News Platform," Department of Computer Science, Nanyang Technological University, Singapore.

[2] Spyridon D. Kaftanis and Dimitrios A. Koutsomitropoulos, "Sentiment Text Analysis for Providing Individualized Feed of Recommendations Using Reinforcement Learning," University of Patras, Computer Engineering and Informatics Department, Patras, Greece.

[3] G. Sudha Sadasivam, Saranya K. G, and E. M. Praveen, "Personalisation of News Recommendation Using Genetic Algorithm," PSG College of Technology, Coimbatore, India.

[4] Hiroaki Takatsu; Mayu Okuda; Yoichi Matsuyama; Hiroshi Honda; Shinya Fujie; Tetsunori Kobayashi; 2021 IEEE Spoken Language Technology Workshop (SLT)

[5] Li Gao;Yinghui Liu;2023 International Conference on Computer Science and Automation Technology (CSAT)

[6] J. Wilson, S. E. Hosseini and S. Pervez, "Identification of Fake News in Social Media Using Sentimental Analysis," 2023 IEEE Industrial Electronics and Applications Conference (IEACon), Penang, Malaysia, 2023, pp. 220-224, doi: 10.1109/IEACon57683.2023.10370300.

[7] S. Nabil, J. Elbouhdidi and M. Yassin Chkouri, "Recommendation System Based on Data Analysis-Application on Tweets Sentiment Analysis," 2018 IEEE 5th International Congress on Information Science and Technology (CiSt), Marrakech, Morocco, 2018, pp. 155-160, doi: 10.1109/CIST.2018.8596418.

[8] R. Dhanvardini, P. Martina, R. Vijay, R. Amirtharajan and P. Pravinkumar, "Development and integration of dApp with blockchain smart contract Truffle Framework for user interactive applications," 2023 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2023, pp. 1-6, doi: 10.1109/ICCCI56745.2023.10128406.

[9] K. G. Saranya, G. Sudha Sadasivam, “A Personalized Online News Recommendation System,” International Journal of Computer Applications, vol. 57, no. 18, 2012.

[10] K. G. Saranya, G. Sudha Sadasivam, “A Personalized Online News Recommendation System,” International Journal of Computer Applications, vol. 57, no. 18, 2012.B. Faltings and F. Garcin, “PEN recsys: a Personalized News Recommender Systems Framework,” in News Recommender Systems Workshop (NRS), October 13, 2013.

[11] K. Cotter, J. Cho, and E. Rader, “Explaining the News Feed Algorithm: An Analysis of the 'News Feed FYI' blog,” in Proc. of the 2017 CHI conference extended abstracts on human factors in computing systems, 2017, pp. 1553-1560.

[12] D. T. Nguyen, S. Joty, M. Imran, H. Sajjad and P. Mitra, “Applications of Online Deep Learning for Crisis Response Using Social Media Information,” arXiv preprint arXiv:1610.01030, 2016.

[13] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender system. A survey of the state-of-the-art and possible extensions,” IEEE Trans Knowl Data Eng; 17(6):734 – 49.

[14] M Barthel, “Newspapers Fact Sheet. Pew Research Center,” Retrieved from http://www.journalism.org/fact-sheet/newspapers/

[15] J. Beel, B. Gipp, S. Langer, and C. Breitinger, “Research-paper recommender systems: a literature survey,” International Journal on Digital Libraries, 17(4): 305-338.

[16] D. Billsus and M. J. Pazzani, “User Modeling for Adaptive News Access,” User Modeling and User-Adapted Interaction, 10(2-3): 147-180.

[17] M. Broersma and T. Graham, “SOCIAL MEDIA AS BEAT,” Journalism Practice, 6(3): 403-419.