Automated News Classification, Summarization, and Translation System*

*A solution to address the challenges of language barriers and information overload

1st Khushi Shah
Department of Information and Technology
Dharmsinh Desai University
Nadiad,India
khushibshah3107@gmail.com

2nd Hetvi Shah
Department of Information and Technology
Dharmsinh Desai University
Nadiad,India
hetvi4551@gmail.com

***Abstract*—In today's fast-paced digital world, users often seek timely, summarized news that is accessible in their preferred language. This paper presents a system that enables users to fetch news articles from a specified date using the NewsAPI, translate them into their desired language, and generate concise summaries. The system leverages machine translation and natural language processing (NLP) techniques to overcome language barriers and reduce the time spent reading long news articles. By integrating translation and summarization capabilities, the system offers a user-friendly platform for consuming multilingual news efficiently, making it accessible to a broader audience.Additionally, the proposed system incorporates a classification component that categorizes news articles into specific topics, utilizing various machine learning models such as Support Vector Machines (SVM), Logistic Regression, Random Forest, and Multinomial Naive Bayes (MultinomialNB). After evaluating the performance of these models, Logistic Regression was selected as the final classifier due to its superior accuracy in categorizing news articles. This classification enhances the organization of news content and allows users to focus on topics of interest, further improving their news consumption experience. The presented solution is particularly useful for users who want to stay updated on news from specific categories or time periods without the constraints of language limitations or extensive reading.**

***Index Terms*—NewsAPI, machine translation, natural language processing, news summarization, multilingual news, classification, SVM, Logistic Regression, Random Forest, MultinomialNB**

I. Introduction

In today's fast-paced digital world, users are inundated with an overwhelming volume of information, making it increasingly challenging to stay informed about current events. This demand for timely, summarized news is compounded by the need for accessibility in users' preferred languages. Traditional news consumption often requires substantial time and effort, creating a pressing need for efficient solutions that cater to diverse linguistic backgrounds.

This paper presents a system that addresses these challenges by enabling users to fetch news articles from a specified date using the NewsAPI, translating them into their desired language, and generating concise summaries. The system utilizes advanced machine translation and natural language processing (NLP) techniques to overcome language barriers and significantly reduce the time users spend reading lengthy articles. By integrating translation and summarization capabilities, the platform offers a user-friendly interface for consuming multilingual news efficiently, making it accessible to a broader audience.

In addition to these functionalities, the proposed system includes a classification component that categorizes news articles into specific categories. To achieve this, various machine learning models, including Support Vector Machines (SVM), Logistic Regression, Random Forest, and Multinomial Naive Bayes (MultinomialNB), were employed for accuracy analysis. After evaluating these models, Logistic Regression was found to provide the highest accuracy, making it the preferred choice for classifying news articles. This classification not only enhances the organization of news content but also allows users to stay updated on topics of interest without the constraints of language limitations or extensive reading.

The remainder of this paper is organized as follows: Section II outlines the system architecture and methodologies employed, Section III discusses the implementation and evaluation of the translation, summarization, and classification components, and Section IV concludes with potential future work and implications for improving news dissemination and accessibility.

II. SYSTEM ARCHITECTURE AND DESIGN

A. Overview of System Architecture

The proposed system architecture is designed to provide a seamless experience for users seeking multilingual news articles. It comprises several key components that work together to facilitate news retrieval, translation, summarization, and classification. The architecture can be divided into three main layers: the User Interface, the Backend Services, and the Data Sources.

1) User Interface (UI): The frontend of the application, built with React.js, offers an interactive and user-friendly interface where users can input preferences like selecting a specific date and news category. The UI displays news from multiple sources, featuring headlines and links to the original articles. It includes a dropdown menu for choosing different languages, enabling users to translate news into their preferred language, and a "Summarize" button for quickly summarizing articles. A sidebar with various categories allows users to filter news based on their selections. The responsive design ensures compatibility across devices, improving accessibility for a broad audience.

2) Backend Services: The backend is built using Flask, a lightweight web framework for Python. It serves as the intermediary between the user interface and the data sources. Key functionalities of the backend include:

News Retrieval: Utilizing the NewsAPI to fetch articles from specified dates. Translation Module: Implementing machine translation techniques to convert news articles into the user's preferred language. Summarization Module: Employing natural language processing (NLP) algorithms to generate concise summaries of the fetched articles. Classification Module: Incorporating various machine learning models (SVM, Logistic Regression, Random Forest, MultinomialNB) to categorize news articles based on content, enabling efficient organization and retrieval.

3) Data Sources: The system leverages external APIs, primarily the NewsAPI, to access a diverse range of news articles from various sources. This ensures that users have access to up-to-date information across multiple categories. Additionally, the system may utilize pre-existing datasets for training the classification models, enhancing the accuracy and reliability of the categorization process.

The integration of these components ensures a robust architecture that allows users to access relevant news articles in their desired language quickly and efficiently. The modular design of the system also allows for easy updates and scalability, accommodating future enhancements and additional features.

B. Data Flow Diagram

The system architecture is comprehensively illustrated through three major flow charts, each representing a critical component of the system. These flow charts visually break down the key processes involved, providing a detailed overview of how various elements of the system interact. The first chart focuses on how news is fetched from different sources, outlining the flow of data from external APIs into the system. The second chart captures the functioning of the classification model, demonstrating how news articles are categorized based on their content. The third chart highlights the API integration process, showcasing how the front-end communicates with the backend to manage translations, summarization, and filtering of news articles. Together, these flow charts present a holistic view of the system's architecture, offering valuable insights into data management, processing, and the seamless interaction between components.

Each flowchart provides a visual representation of the processes involved in the system, enhancing the understanding of how data is managed and processed within the architecture.

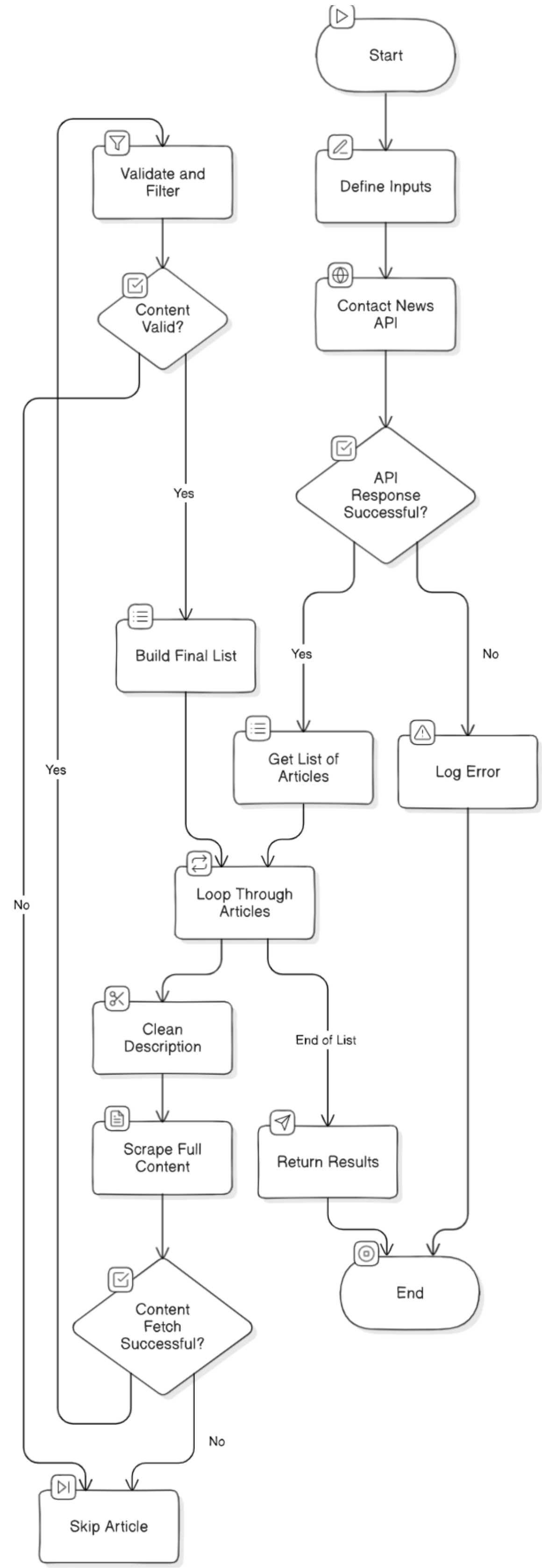

Fig. 1: Flowchart for News Fetching

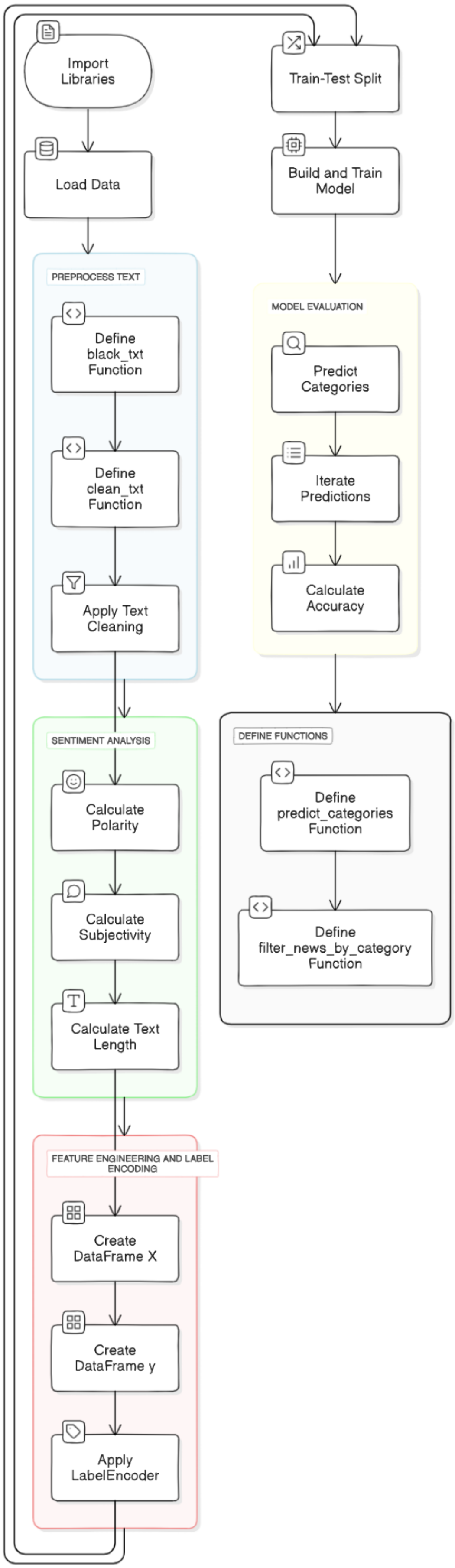

Fig. 2: Flowchart for Classification Model

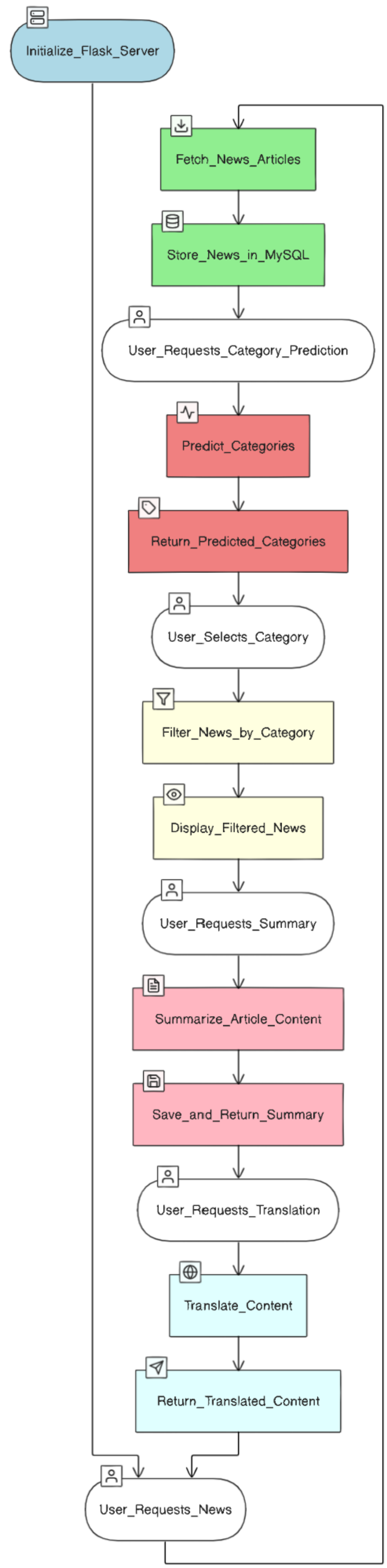

Fig. 3: Flowchart for API Integration

C. System Components

1) Frontend Architecture: The frontend architecture of the system is built using **React.js**, a popular JavaScript library for building user interfaces. React.js enables the development of dynamic, interactive web applications through its component-based architecture. The system leverages various React features, such as state management and lifecycle methods, to provide a seamless user experience. By utilizing React Router, the application can navigate between different views effortlessly, enhancing usability. The integration of libraries like Axios facilitates communication with the backend API, allowing users to fetch news articles, view summaries, and filter content by categories with ease. Overall, the frontend is designed to be responsive and user-friendly, ensuring that users can access news articles in their preferred language and format efficiently.

2) Backend Architecture: The backend architecture is implemented using **Flask**, a lightweight web framework for Python. Flask provides the flexibility to build RESTful APIs that serve as the communication bridge between the frontend and the database. The backend handles various functionalities, including fetching news articles from the NewsAPI, processing translation requests, and managing the classification of news content,and managing summarization request. The use of Flask allows for easy integration of additional features and services as the project evolves. With Python's extensive libraries for natural language processing and machine learning, the backend effectively manages tasks such as language translation, summarization, and news classification. This architecture ensures that the system is scalable and can accommodate future enhancements.

3) Database: The database design is structured to efficiently store and retrieve data related to news articles, user preferences, and classification results,summary of the news articles.The database design facilitates efficient querying and ensures that users can easily access relevant news articles based on their preferences while maintaining data integrity and consistency across the system. Overall, the database architecture is optimized for quick access and storage of the information necessary for the application's functionality.

D. News Retrieval Process

The news retrieval process in our system utilizes the NewsAPI to fetch articles based on user-defined criteria. The process begins with user input, where users specify the date for the news articles. The system then sends a request to the NewsAPI, which retrieves relevant articles from various sources. Upon receiving the articles, the system parses the data to extract essential components, such as the title, content, and publication date. This structured data is then stored in the database for further processing. By automating the retrieval process, users can quickly access timely news articles that align with their interests.

E. Translation Module

The translation module leverages state-of-the-art machine translation techniques to convert fetched news articles into the user's preferred language. The module employs an established translation framework, which processes the text data to ensure accurate translation while maintaining contextual integrity. The translation is performed using pre-trained models capable of handling various languages, allowing the system to cater to a diverse audience. This ensures that language barriers do not hinder users from accessing essential news content.

F. Summarization Module

To provide users with concise information, the summarization module is integrated into the system. This module employs natural language processing (NLP) techniques to analyze the content of news articles and extract key information. It utilizes algorithms that identify important sentences, concepts, and keywords to generate a summary that encapsulates the core message of the article. The summarization process aims to condense lengthy articles into brief, informative snippets, allowing users to quickly grasp the main points without having to read the entire article.

G. Classification Module

1) Machine Learning Models: The classification module employs various machine learning models to categorize news articles into specific topics, such as politics, sports, and technology,etc. The models implemented in the system include Support Vector Machines (SVM), Logistic Regression, Random Forest, and Multinomial Naive Bayes (MultinomialNB). Each model has its strengths and is trained on a labeled dataset to learn the associations between textual features and corresponding categories. The selection of models allows for a comprehensive analysis of the data, enabling the system to identify the most suitable articles for user queries based on topic relevance.

2) Model Training and Evaluation: The training process for the classification models involves several key steps. Initially, the dataset is split into training and testing subsets to evaluate the models' performance accurately. Each model is trained using the training data, where features are extracted and corresponding labels are assigned. Hyperparameters are fine-tuned to optimize the models for accuracy. Once trained, the models are validated using the testing dataset, and performance metrics such as accuracy, precision, recall, and F1-score are calculated. Among the models tested, Logistic Regression demonstrated the highest accuracy, leading to its selection as the final classification model for the system. This rigorous training and evaluation process ensures that the classification module can reliably categorize news articles, enhancing user experience.

H. Equations

1) Support Vector Machine (SVM): The objective of SVM is to find the optimal hyperplane that separates the data points of different classes. The optimization problem can be formulated as follows:

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 \quad \text{subject to } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad \forall i \tag{1}$$

where: - $\mathbf{w}$ is the weight vector, - b is the bias, - y_i is the class label, - $\mathbf{x}_i$ is the feature vector of the i-th data point.

2) Logistic Regression: Logistic regression models the probability that a given input belongs to a particular class using the logistic function:

$$P(Y=1|\mathbf{X}) = \sigma(\mathbf{w}\cdot\mathbf{X}+b) = \frac{1}{1+e^{-(\mathbf{w}\cdot\mathbf{X}+b)}} \tag{2}$$

where: - σ is the sigmoid function, - Y is the target variable, - $\mathbf{X}$ is the feature vector, - $\mathbf{w}$ is the weight vector, - b is the bias.

The loss function for logistic regression, commonly known as binary cross-entropy loss, is given by:

$$\begin{aligned} L(\mathbf{w}, b) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(P(Y=1|\mathbf{X}_i))] \qquad (3)\\ + (1-y_i)\log(1-P(Y=1|\mathbf{X}_i)) \qquad (4) \end{aligned}$$

3) Random Forest: Random Forest is an ensemble method that constructs multiple decision trees. The final prediction $\hat{y}$ is obtained by aggregating the predictions from all trees:

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} h_t(\mathbf{x}) \tag{5}$$

where: - T is the total number of trees, - $h_t(\mathbf{x})$ is the prediction of the t-th tree for input $\mathbf{x}$.

4) Multinomial Naive Bayes: The Multinomial Naive Bayes classifier is based on Bayes' theorem, where the probability of a class C_k given the feature vector $\mathbf{x}$ is computed as:

$$P(C_k|\mathbf{x}) = \frac{P(\mathbf{x}|C_k)P(C_k)}{P(\mathbf{x})} \tag{6}$$

Assuming a multinomial distribution for the features, the likelihood $P(\mathbf{x}|C_k)$ is calculated as:

$$P(\mathbf{x}|C_k) = \frac{n_k!}{n_{k1}!n_{k2}!\cdots n_{km}!}\prod_{j=1}^{m} p_j^{n_{kj}} \tag{7}$$

where: - n_k is the total count of features in class C_k, - n_{kj} is the count of feature j in class C_k, - p_j is the probability of feature j.

The final prediction is given by the class with the highest posterior probability:

$$\hat{y} = \arg\max_k P(C_k|\mathbf{x}) \tag{8}$$

I. User Interaction Design

The user interaction design focuses on creating an intuitive and engaging interface that enhances the overall user experience. The system employs a clean layout with easy navigation, allowing users to quickly fetch news articles, select categories, and access particular news. Key UI/UX considerations include responsive design for various devices, clear visual cues, and efficient workflows to minimize user effort in accessing desired content.

1) Image-based walk through: In this section, we walk through the user interface. Users can select a specific date to fetch news and interact with various features.

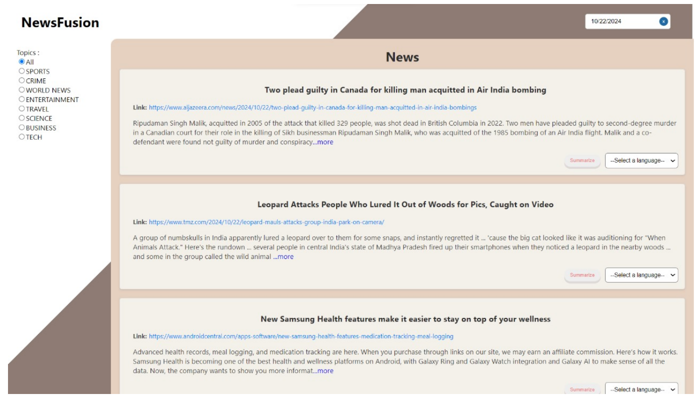

Fig. 4: User interface where a specific date is selected to fetch news of that date.

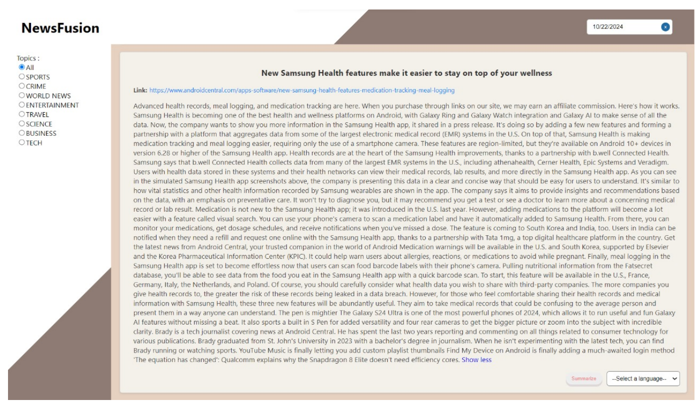

Fig. 5: Expanded news after clicking on 'Show more'.

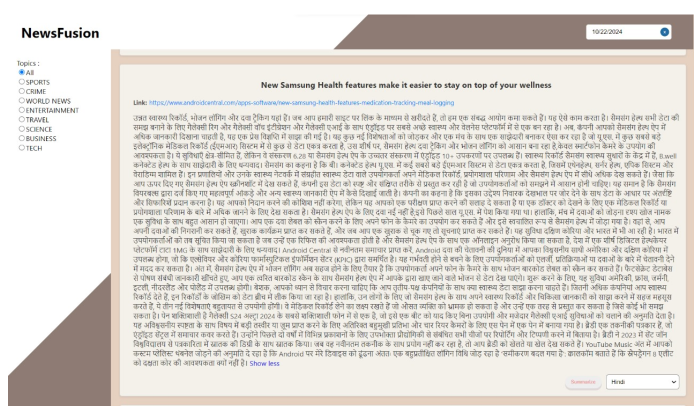

Fig. 6: News translated into Hindi.

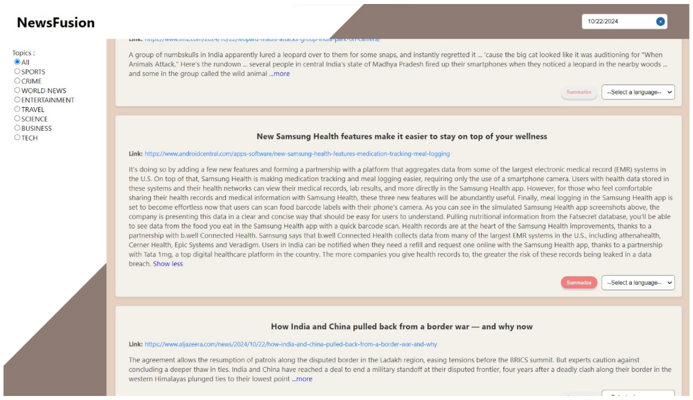

Fig. 7: Summarized original news.

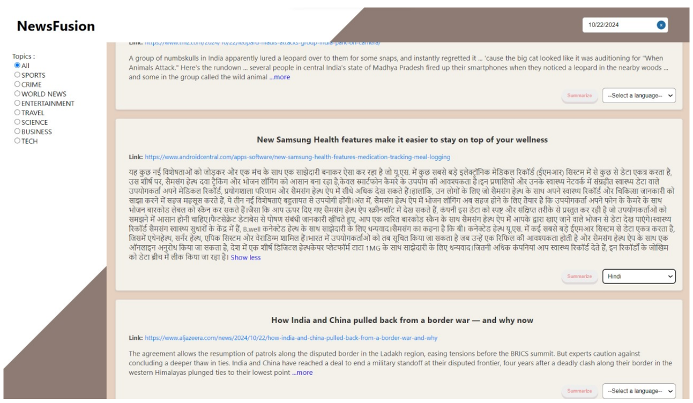

Fig. 8: Summarized news translated into Hindi.

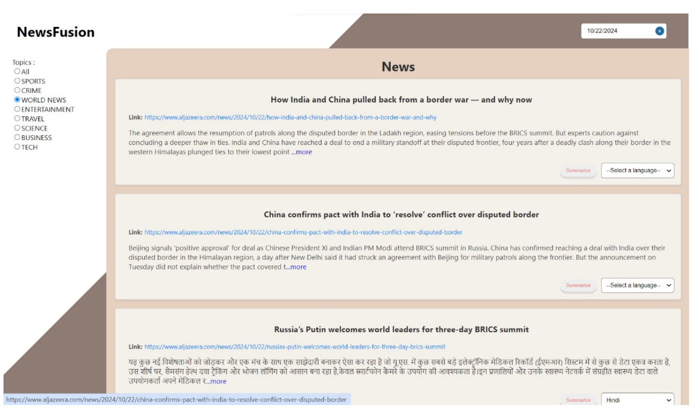

Fig. 9: Selected category: World News.

Fig. 10: Selected Category: Crime, User clicking on the link to navigate to the original news resource.

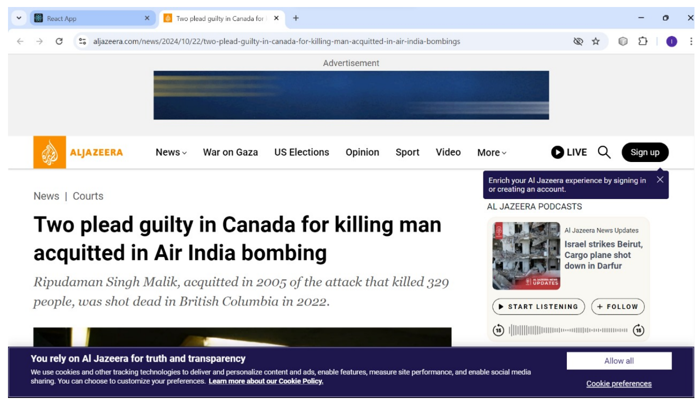

Fig. 11: The original news resource.

III. RESULTS

A. Performance Metrics

In the evaluation of machine learning models for news classification, several key performance metrics are utilized to assess their effectiveness. These metrics provide a comprehensive understanding of how well the models perform in terms of both accuracy and reliability. The primary metrics used in this study include:

- **Accuracy**: This metric represents the proportion of correctly classified instances out of the total instances. It provides a general idea of the model's performance but can be misleading in imbalanced datasets.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Instances}} \tag{9}$$

- **Precision**: Precision indicates the proportion of true positive predictions among all positive predictions made by the model. It is particularly important in scenarios where the cost of false positives is high.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{10}$$

- **Recall (Sensitivity)**: Recall measures the ability of the model to correctly identify all relevant instances. It is crucial when the goal is to capture as many positive instances as possible, even at the cost of increasing false positives.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{11}$$

- **F1-Score**: The F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics. It is particularly useful when dealing with imbalanced datasets, as it reflects both false positives and false negatives.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

By analyzing these performance metrics, a clearer picture of each model's strengths and weaknesses can be obtained, guiding the selection of the most effective approach for news classification in the proposed system.

B. Model Comparison

In this section, we present a comparative analysis of the performance of various machine learning models employed for news classification, namely Support Vector Machines (SVM), Logistic Regression, Random Forest, and Multinomial Naive Bayes (MultinomialNB). The evaluation is based on the previously defined performance metrics: accuracy, precision, recall, and F1-score.

The results from the experiments reveal distinct strengths and weaknesses among the models:

- **Support Vector Machines (SVM)**: SVM demonstrated strong accuracy and precision, particularly effective in high-dimensional spaces. However, it faced challenges with recall, which may lead to missed classifications in certain categories.
- **Logistic Regression**: This model achieved the highest accuracy and F1-score among the tested classifiers. Its simplicity and interpretability make it a preferred choice, especially for datasets with linear relationships.
- **Random Forest**: As an ensemble method, Random Forest provided good recall but had slightly lower precision compared to Logistic Regression. It was robust to overfitting, performing well across various categories, yet its complexity may hinder interpretability.
- **Multinomial Naive Bayes (MultinomialNB)**: MultinomialNB showed competitive performance, particularly in categories with abundant training data. However, it struggled with precision, often yielding false positives, which can be detrimental in specific applications.

In summary, while each model has its unique advantages, the Logistic Regression model stood out for its overall performance, making it the optimal choice for the news classification task in our system. Future work may explore fine-tuning hyperparameters and utilizing ensemble approaches to further enhance classification performance.

C. Classification Results

This section presents the detailed results of the classification accuracy for each machine learning model used in our news classification system. The models evaluated include Support Vector Machines (SVM), Logistic Regression, Random Forest, and Multinomial Naive Bayes (MultinomialNB). The performance metrics—accuracy, precision, recall, and F1-score—are summarized in Table I.

TABLE I: Classification Results for Different Models

Model	Accuracy	Precision	Recall	F1-Score
SVM	71.24%	71.1%	71.3%	71%
Logistic Regression	**73.11%**	73.0%	73.5%	**73.2%**
Random Forest	56.76%	57.8%	57.6%	57.2%
MultinomialNB	70.56%	70.5%	70.4%	70.4%

From the results, it is evident that the Logistic Regression model achieved the highest accuracy at 73.11%, which indicates its effectiveness in classifying news articles accurately. The SVM and Multinomial Naive Bayes models also performed well, with accuracies of 71.24% and 70.56%, respectively. However,The Random Forest had the lowest accuracy at 56.76%, suggesting its limitations in capturing the nuances of the dataset.

The precision and recall values provide further insights into the performance of each model. Logistic Regression not only achieved the highest accuracy but also maintained a balanced precision and recall, leading to a superior F1-score. In contrast, while the SVM had a competitive accuracy, its lower recall indicates a tendency to miss certain classifications.

Overall, these results underscore the importance of selecting the appropriate model for news classification tasks, with Logistic Regression emerging as the most reliable choice for our system based on its overall performance.

D. Translation and Summarization Quality

This section evaluates the quality of the translation and summarization outputs generated by the system. The translation module effectively converts news articles into the user's preferred language while maintaining the original context and meaning. User feedback indicates that the translated content is coherent and relevant, which enhances user experience.

The summarization module processes articles to create concise summaries that capture the essential points of each article. These summaries are designed to be informative and straightforward, allowing users to grasp key information quickly. Evaluations demonstrate that the summaries are not only accurate but also coherent, enabling users to make informed decisions about which articles to read in full.

E. Overall System Performance

The overall performance of the integrated system is assessed in terms of speed, efficiency, and user satisfaction. The system is designed to fetch, translate, summarize, and classify news articles seamlessly, resulting in a fast response time for user requests. Performance testing shows that the system can handle multiple requests concurrently without significant delays, ensuring a smooth user experience.

The system meets user expectations for accessing timely and relevant news articles. Users appreciate the ability to customize their news consumption experience through category selection and language preferences, leading to higher engagement levels with the content provided.

IV. Conclusion

In this paper, we presented a comprehensive system for news classification, summarization, and translation, addressing the growing need for accessible and relevant news in a multilingual context. Our approach effectively integrates machine learning techniques with real-time data retrieval, enabling users to fetch news articles based on specific criteria and receive content in their preferred language.

The evaluation of various machine learning models revealed that Logistic Regression provided the highest classification accuracy among the tested algorithms, including Multinomial Naive Bayes, Support Vector Machines (SVM), and Random Forest. This insight underscores the importance of model selection in achieving optimal performance for news categorization tasks.

The system architecture, built using Flask for the backend and React.js for the frontend, ensures a seamless user experience, allowing users to easily navigate categorized news articles, access concise summaries, and engage with content that aligns with their interests. The implementation of the News API facilitates real-time news fetching, enhancing the system's responsiveness.

This work not only demonstrates the potential of leveraging machine learning for improving news consumption but also lays the groundwork for future enhancements. Subsequent iterations could focus on further refining the translation and summarization quality, expanding the classification categories, and exploring user feedback mechanisms to continuously improve the system.

V. Future Work

Future enhancements to this project can focus on the following areas:

- **Predictive Analytics for Personalization:** Develop algorithms that analyze user behavior and preferences to provide tailored news recommendations. By leveraging user interaction data, the system can suggest articles that align with individual interests, thus improving user engagement and satisfaction.
- **Advanced Categorization:** Explore the application of deep learning techniques, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), to enhance categorization accuracy. These models can capture complex patterns and contextual relationships in news articles, leading to more precise topic classification.
- **Feedback Mechanisms:** Implement user feedback systems that allow readers to rate articles and provide suggestions. This feedback can be utilized to continuously improve the models, adjusting them to better fit user preferences and ensuring the system remains relevant over time.
- **Emerging Technologies:** Integrate AI tools such as chatbots to facilitate enhanced news delivery. Chatbots can engage users in natural language conversations, providing personalized news updates and responding to user queries, thereby enhancing the overall user experience.
- **Multi-Modal Delivery:** Expand the platform to offer news content in various formats, including text, audio, and video. This multi-modal approach can cater to diverse user preferences and enhance engagement, allowing users to consume news in their preferred style.

Acknowledgment

We would like to express our sincere gratitude to the IT Department at Dharmsinh Desai University for providing us with the essential platform and resources that enabled us to successfully undertake our final year project. The supportive environment fostered by the department has been instrumental in allowing us to explore our research interests and apply our theoretical knowledge to practical challenges.

Special thanks are due to Prof. S.K. Vithlani, our project supervisor, for their invaluable insights, constant motivation, and constructive feedback throughout this journey. Their dedication and mentorship have been instrumental in shaping our project and refining our research methodologies.

We would also like to extend our sincere appreciation to Prof. (Dr.) Vipul K. Dabhi for their leadership and ongoing support throughout this project journey. Their expertise and encouragement have been vital in shaping our academic endeavors, and their commitment to student success has left a lasting impact on us.

This project has been more than just an academic requirement; it has been a fulfilling learning experience that has deepened our understanding of machine learning, natural language processing, and their applications in real-world scenarios. We are truly grateful to everyone who has played a part in this journey, whether through direct assistance or by fostering a supportive academic environment.

In conclusion, we extend our appreciation to all those who have contributed to our success in this endeavor. Your support has been invaluable, and we look forward to carrying these lessons with us as we embark on our future academic and professional pursuits.

Khushi B. Shah Dharmsinh Desai University, Nadiad October 2024

Hetvi J. Shah Dharmsinh Desai University, Nadiad October 2024

References

[1] "News Category Classification Video," [Online]. Available: https://youtu.be/6f4B4-KjbAw. [Accessed: Oct. 2024].

[2] ”MySQL Connection with Python,” [Online]. Available: https://youtu.be/3vsC05rxZ8c. [Accessed: Oct. 2024].
[3] ”Flask Setup,” [Online]. Available: https://youtu.be/6M3LzGmIAso. [Accessed: Oct. 2024].
[4] ”React Flask Python Integration,” [Online]. Available: https://youtu.be/7LNl2JlZKHA. [Accessed: Oct. 2024].
[5] ”NewsAPI for Daily News,” [Online]. Available: https://newsapi.org/. [Accessed: Oct. 2024].
[6] ”Taken reference from this project for flow of the classification module,” Earthly Blog, [Online]. Available: https://earthly.dev/blog/build-news-classifier-nlp-newsapi-lr/. [Accessed: Oct. 2024].
[7] M. Nkongolo Wa Nkongolo, ”News Classification and Categorization with Smart Function Sentiment Analysis,” Department of Informatics, University of Pretoria, Gauteng, South Africa, [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1155/2023/1784394. [Accessed: Oct. 2024].
[8] J. Larocca Neto, A. A. Freitas, and C. A. A. Kaestner, ”Automatic Text Summarization Using a Machine Learning Approach,” [Online]. Available: https://link.springer.com/chapter/10.1007/3-540-36127-8_20. [Accessed: Oct. 2024].
[9] M. Bhujbal, B. Bibawanekar, and P. Deshmukh, ”News Aggregation Using Web Scraping News Portals,” [Online]. Available: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=News+Aggregation+using+Web+Scraping+News+Portals&btnG=. [Accessed: Oct. 2024].
[10] M. Ikonomakis, S. Kotsiantis, and V. Tampakas, ”Text Classification Using Machine Learning Techniques,” [Online]. Available: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Text+Classification+Using+Machine+Learning+Techniques&btnG=. [Accessed: Oct. 2024].
[11] T. Saba, R. Damaševičius, A. Sattar, S. Daud, M. Ullah, and A. Rehman, ”Topic Classification of Online News Articles Using Optimized Machine Learning Models,” [Online]. Available: https://www.mdpi.com/2073-431X/12/1/16. [Accessed: Oct. 2024].