DECLARATION

We certify that:

i. The work presented in this project report is an authentic record of our own work under the guidance of our supervisor. It has not been submitted to any other Institute for the award of any other degree or diploma.

ii. Whenever we have used information (text, data, figure, photograph, chart, analysis, inference, etc.) from other sources, we have given due credit by citing it in the text of the report and providing its details in the references.

iii. We have followed the guidelines provided by the department for preparing this report.

Ridham(2821450)

Ishu Kadian (2821436)

Project report title: News Sentiment Analysis

Semester: 7 t h

Date:

APPROVAL FROM SUPERVISOR

This is to certify that the project report entitled "*News Sentiment Analysis*" presented by "*Ridham (2821450), Ishu Kadian (2821436)*" under my supervision is an authentic work. To the best of my knowledge, the content of this report has not been submitted for the award of any previous degree to anyone else.

It is recommended that the report be accepted as fulfilling this part of the requirements for the award of the degree.

_____

Name: Dr. Shally
Designation:Assistant Professor
Department of CSE (Artificial Intelligence & Data Science)
Date:

_____

(Counter Signed by)
Head, Department of CSE (Artificial Intelligence & Data Science)

CERTIFICATE

This is to certify that the work embodied in this report, entitled "*News Sentiment Analysis*" carried out by "*Ridham (2821450), Ishu Kadian (2821436),* " is approved for the degree of "*Bachelor of Technology in Computer Science and Engineering (Artificial Intelligence and Data Science)*" at the department of "*Computer Science and Engineering (Artificial Intelligence and Data Science)*", Panipat Institute of Engineering and Technology, Samalkha.

_____

Internal Examiner

_____

External Examiner

Date :

Place : Panipat

ACKNOWLEDGEMENTS

It gives us a great sense of pleasure to present the report of the Project undertaken during B. Tech. Computer Science & Engineering (Artificial Intelligence & Data Science) Final Year.

We would like to express my deepest gratitude to Guide name, Designation, Department of COMPUTER SCIENCE & ENGINEERING (Artificial Intelligence & Data Science), PIET, Samalkha for his/her exceptional dedication and major contributions that have played a pivotal role in the realization of this project. His/Her sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his/her cognizant efforts that our endeavors have seen light of the day.

We would like to extend our sincere thanks to Name Head of Department, Head Department of COMPUTER SCIENCE & ENGINEERING (Artificial Intelligence & Data Science), PIET, Samalkha for his full support and assistance during the development of the project.

We are thankful to all faculty members of the department for their kind assistance, guidance, and cooperation during the development of our project.

Lastly, we would like to acknowledge our friends for their contribution to the completion of the project.

Ridham(2821450)
Ishu Kadian(2821436)
Date:

ABSTRACT

The Press Information Bureau (PIB) plays a vital role in communicating government policies and initiatives to the public. Given the expanding digital media landscape and India's regional language diversity, monitoring public reactions has become increasingly challenging. This project proposes an AI-driven automated feedback system to track and analyze regional media reactions in real-time.

The system will use web scraping techniques and APIs to collect data from regional media sites in languages such as Hindi, Urdu, Punjabi, Marathi, Tamil, Kannada, and others. It will monitor a range of digital content, including articles, blogs, and social media posts. Natural Language Processing (NLP) will be used to analyze this content, detecting sentiment (positive, negative, or neutral) and identifying key topics, government policies, and individuals mentioned.

AI-based sentiment analysis will allow the system to provide insights into how government initiatives are perceived in different regions. It will extract key phrases and trends, enabling PIB to track public opinion and media coverage more effectively. A dynamic dashboard will visualize these insights in real-time, offering sentiment graphs and trend maps to help PIB monitor sentiment shifts, emerging topics, and any media anomalies.

The system will continuously improve through machine learning, adapting to evolving regional media trends and language nuances. By retraining models with new data, the system will remain relevant and accurate over time.

In summary, this AI-powered feedback system will enhance PIB's ability to monitor public sentiment and media reactions. It will provide real-time insights that support informed decision-making and more effective government communication, making it an essential tool for managing public perceptions and media interactions.

TABLE OF CONTENTS

Chapter 2: Literature Review

Chapter 3: Problem Objective

Chapter 4: Methodology of the Project

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Purpose and Objective of the Work

The primary purpose of this project is to create a system that enables PIB to effectively monitor and analyze public sentiment regarding government actions as reflected in regional digital media. By leveraging AI and Machine Learning technologies, the system will provide PIB with real-time feedback on how different media outlets across various regions and languages are covering government initiatives. This feedback will help PIB identify trends, gauge public perception, and address any issues or concerns raised by the media or the public. The system aims to streamline the process of collecting and analyzing data, providing PIB with actionable insights to improve the impact and reach of government communication efforts.

This work is designed to create an automated solution that can monitor a vast number of regional media sites, including news outlets, blogs, and social media platforms, to ensure continuous and accurate feedback on public sentiment. It will employ advanced Natural Language Processing (NLP) techniques to analyze the collected content, detecting sentiment, identifying key themes, and offering valuable insights into public reactions. By focusing on multiple regional languages, the system will ensure that the diverse linguistic landscape of India is represented accurately, allowing PIB to assess reactions from various cultural and regional perspectives.

1.2 The Problem and Its Academic, Commercial, or Social Relevance

The challenge at hand focuses on the growing necessity to understand and analyze the sentiments embedded within news articles across various digital platforms. As news consumption becomes increasingly digital and global, distinguishing the emotional tone of news

content is crucial for identifying public sentiment, monitoring biases, and ensuring responsible journalism. This complexity is heightened by the diversity of news sources, the prevalence of misinformation, and the multilingual nature of global news, creating an intricate landscape for sentiment analysis.

The academic significance of this project lies in its contribution to the evolving field of natural language processing (NLP) and sentiment analysis. By applying cutting-edge machine learning algorithms and data preprocessing techniques, this work demonstrates the practical utility of theoretical concepts, showcasing their application to a real-world, high-impact problem. It further enriches the academic discourse by addressing challenges like handling multilingual data, contextual sentiment analysis, and large-scale data processing.

From a commercial perspective, this project offers significant insights into media monitoring and sentiment tracking, which are invaluable for stakeholders such as businesses, policymakers, and media outlets. Understanding the sentiment trends in news articles empowers organizations to craft targeted communication strategies, mitigate potential PR crises, and make informed decisions. The commercial relevance is highlighted by the project's potential to drive innovation in sentiment analysis tools, fostering competitive advantages for businesses reliant on public sentiment insights.

On a societal level, this project addresses the broader implications of news on shaping public opinion and influencing societal discourse. By analyzing sentiment trends, it contributes to the critical evaluation of news narratives, promoting transparency and accountability in journalism. Furthermore, it enables audiences to better comprehend the emotional undercurrents of the media they consume, thus fostering informed and responsible consumption of information. This societal relevance aligns with the broader goal of leveraging technology to create a more aware, engaged, and resilient society in the face of an evolving media ecosystem.

1.3 Constraints Affecting the Work

Every sentiment analysis project operates within a set of constraints that shape the methodologies, scope, and reliability of the outcomes. In the context of this project, several factors influence the analytical approaches and eventual findings:

The success of the project is largely dependent on the availability and quality of news data collected from various online sources. While abundant news articles are accessible, challenges

such as incomplete metadata, inconsistencies in article formatting, and potential biases in data collection can limit the comprehensiveness of the analysis.

Ensuring ethical data handling practices is a cornerstone of this work. The project adheres to privacy and copyright guidelines by focusing solely on publicly available and anonymized news data. While this approach safeguards intellectual property and user privacy, it restricts access to premium or proprietary datasets, potentially limiting the diversity of the analyzed content.

The multilingual nature of news content presents a significant constraint. While sentiment analysis models excel in processing English-language text, analyzing articles in other languages requires robust multilingual NLP tools, which may not achieve the same level of accuracy or contextual understanding.

Temporal dynamics also impose limitations on the project. The reliance on historical news data enables a retrospective analysis but may not fully capture the fast-evolving narratives and sentiments in real-time media coverage.

The scope of the project is shaped by its focus on sentiment analysis rather than comprehensive media content evaluation. As a result, certain aspects, such as fact-checking or detailed topic modeling, remain outside the immediate objectives of the study.

Acknowledging these constraints is essential for framing the findings within a realistic context and understanding the limitations that may influence the project's depth, accuracy, and applicability.

1.4 Methodology Used in the Work

The methodology employed in this project follows a systematic and structured approach, encompassing several critical phases to ensure robust sentiment analysis of news articles:

1.4.1 Data Collection and Preprocessing

The foundation of this analysis involves gathering relevant data from diverse news websites. This phase includes scraping or sourcing news articles, headlines, and metadata using web scraping tools or APIs. Preprocessing steps include cleaning text data, removing irrelevant content (e.g., advertisements or navigation links), and normalizing formats for consistency. Tokenization, stemming, and stop-word removal are performed to prepare the data for analysis.

`

1.4.2 Semantic Model Deployment

To classify news articles, we employ state-of-the-art natural language processing (NLP) techniques. Pre-trained sentiment analysis models such as VADER or BERT are fine-tuned on news datasets to enhance their accuracy in identifying the sentiment (positive, negative, or neutral) within articles. Custom models are trained for specific scenarios where domain-specific sentiment nuances are essential.

1.4.3 Topic Detection and Trend Analysis

Topic modeling techniques like Latent Dirichlet Allocation (LDA) or clustering algorithms are used to identify prominent themes in the news articles. Time-series analysis is applied to track sentiment trends over specific periods, highlighting shifts in public mood around key events or topics.

1.4.4 Python Data Analysis

Python is the primary programming tool utilized for this phase. Libraries such as Pandas and NumPy are employed for data manipulation, while Matplotlib and Seaborn are used for preliminary visualizations. Machine learning libraries like Scikit-learn and TensorFlow facilitate the implementation of sentiment classification models and topic detection algorithms

1.4.5 Visualization and Insights Presentation

The insights derived from the analysis are translated into an interactive dashboard using visualization tools like Tableau or Power BI. This phase involves creating dynamic charts, graphs, and filters to allow stakeholders to explore sentiment distributions, trending topics, and regional variations in news sentiment effectively.

1.5 Important Findings

The sentiment distribution across the analyzed content revealed notable patterns. Official announcements and government-driven news from the PIB site predominantly carried a positive or neutral sentiment, reflecting a focus on achievements, initiatives, and constructive updates.

However, when these headlines were cross-referenced with external news platforms, there were discrepancies in sentiment.

Positive sentiment was commonly associated with announcements of public welfare schemes, advancements in technology, infrastructure developments, and economic growth initiatives. Neutral sentiment was observed in reports of events, factual updates, and policy details without subjective framing. Negative sentiment emerged primarily in external coverage, often related to criticisms, potential drawbacks of government actions, or opinions highlighting challenges.

The sentiment variation indicates how the tone and framing of the same headline can shift depending on the source, with external platforms sometimes introducing critical or opposing perspectives that were absent in the original report. These findings emphasize the dynamic nature of sentiment in public discourse and the influence of contextual framing on audience perception.

1.6 Conclusions

The central conclusions derived from this project focus on providing actionable insights into news sentiment dynamics, offering valuable tools for media analysis, stakeholder decision-making, and audience engagement. The project achieves the following:

1.6.1 Enhanced Sentiment Tracking

Develop and refine sentiment tracking mechanisms for real-time monitoring of news content. These insights enable media organizations and policymakers to promptly identify shifts in public sentiment and respond effectively to emerging narratives.

1.6.2 Targeted Communication Strategies

Utilize sentiment and topic analysis to craft targeted communication strategies. Businesses, governments, and NGOs can leverage these findings to address specific audience concerns, promote positive messaging, and mitigate reputational risks associated with negative news.

1.6.3 Bias Detection and Accountability

Highlight variations in sentiment and bias across different media platforms. This

understanding fosters greater accountability in journalism and encourages readers to approach news consumption critically, promoting transparency and balanced reporting.

1.6.4 Improved Media Monitoring Tools

Incorporate these findings into the development of advanced media monitoring tools. By integrating sentiment analysis and topic modeling, stakeholders can gain comprehensive insights into trends and sentiment shifts within specific industries or regions.

1.6.5 Multilingual Analysis Expansion

Expand the scope of sentiment analysis to encompass multilingual data more effectively. Enhancing the accuracy of sentiment detection across languages facilitates global media monitoring and audience understanding.

These recommendations, grounded in the project's findings, aim to empower stakeholders with tools to navigate the complex media environment responsibly and effectively. Subsequent sections delve into the methodology, elaborate on the analytical techniques used, present detailed results, and discuss limitations and directions for future work.

`

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

The literature review provides a comprehensive exploration of existing research, methodologies, and tools relevant to sentiment analysis and its applications in analyzing news and public information. It delves into the advancements in natural language processing (NLP) and machine learning that have enabled effective sentiment detection across diverse languages and content types. Furthermore, it examines the role of sentiment analysis in evaluating public perception, media biases, and the alignment or divergence between original announcements and subsequent media coverage.

The review highlights key challenges in sentiment analysis, such as handling multilingual data, translating regional languages, and addressing the nuances of contextual interpretation. It also explores the significance of OCR technology in extracting data from non-textual formats like PDFs, which is particularly relevant in the context of government documents and news archives. By synthesizing insights from prior work, the literature review establishes the theoretical and practical framework for the development and implementation of the proposed system. This system aims to analyze and compare sentiments from multiple sources, contributing to a deeper understanding of information dissemination and its societal impact.

2.2 Web Scraping and Data Collection from Regional Media

Web scraping, the process of automatically extracting data from websites, is foundational for building systems that analyze media coverage. The importance of scraping regional news sites and social media for real-time information is well-documented in the literature. According to **Agerri et al. (2019)**, automated data collection enables real-time tracking of public sentiment and media coverage, which is crucial for organizations to stay ahead of emerging trends and issues. As the media landscape becomes increasingly digital, the need to capture and analyze content from a diverse set of online sources has become critical for informing public policy decisions, shaping government initiatives, and gauging public reaction.

Scraping regional media sites introduces challenges that differ from scraping mainstream, monolingual sources. **Mishra & Singh (2020)** stress the importance of extracting information from

`

diverse linguistic and cultural contexts in India. Regional websites, often catering to local populations, may have unique structural formats, varying from mainstream media. Some sites may include multilingual content, mixed-language articles, or region-specific jargon, complicating the data extraction process. Moreover, the presence of multimedia content such as images, videos, and interactive elements adds further complexity to scraping efforts.

While several tools and frameworks exist for web scraping, **Scrapy** (Monica & Dreiling, 2016) and **BeautifulSoup** (Richardson, 2014) are among the most commonly used for Python-based web scraping tasks. These tools allow for extracting structured data from web pages such as article headlines, content, publication dates, and user interactions, such as comments and shares. **Scrapy** provides advanced features for handling large-scale crawls, handling multiple requests concurrently, and processing data in real-time, which are essential when working with large amounts of data from news sources across different languages. **BeautifulSoup** is popular for its ease of use in parsing HTML content and extracting data in a structured manner, making it an excellent choice for smaller-scale scraping projects or when working with simpler websites.

Furthermore, **Selenium** (Shen et al., 2018) is often used to automate interactions with dynamic content, such as pages that require user input for full display or websites that use JavaScript to load data asynchronously. Selenium simulates real user interactions, allowing it to scrape data from pages where traditional scraping tools like Scrapy and BeautifulSoup may fail. However, when scraping from media websites, the presence of dynamic loading and data-heavy formats (such as embedded media and advertisements) requires an even more customized approach.

Despite the availability of scraping tools, challenges remain when dealing with regional Indian media sources. Many regional sites may have inconsistent structures due to limited resources, irregular updates, or differing content formats. Additionally, content is often embedded in forms that are hard to access programmatically. **Muralidharan et al. (2019)** highlight how many regional media outlets do not follow standardized HTML structures, which increases the difficulty of scraping content efficiently. As regional sites may feature a blend of long-form articles, short stories, opinion pieces, and user-generated content, each of these formats might require different scraping strategies. Additionally, frequent updates and the dynamic nature of media content mean that scraping systems must be designed to automatically adapt and adjust to these changes.

Thus, researchers have begun to explore more advanced techniques for overcoming these limitations. For instance, **Bayesian Networks** and **Deep Reinforcement Learning** (Hussain et al., 2020) are

`

increasingly being used to develop adaptive scraping systems that can automatically adjust to changes in website layouts or data formats. These models allow the scraper to learn which sections of a website contain the most valuable data, thus ensuring more efficient data extraction from heterogeneous sources.

2.3 Natural Language Processing (NLP) and Language Detection

Indian regional languages are highly diverse, presenting a significant challenge for Natural Language Processing (NLP) systems. With over 22 officially recognized languages and hundreds of dialects spoken across the country, a key challenge in NLP is designing systems that can handle this vast linguistic diversity. While many NLP models have been developed for languages like Hindi, Tamil, Bengali, and Punjabi, these models often struggle with the complexities of language morphology, syntax, and the need for contextual understanding, particularly when working with less-resourced languages. The **Indian languages** are marked by their rich morphology, where words may change significantly in different contexts based on tense, gender, or number, which makes tokenization and parsing more complicated than in languages like English.

Research on NLP for Indian languages has gained considerable traction in recent years. **Joshi et al. (2020)** emphasize the need for advanced multilingual NLP models capable of processing multiple Indian languages simultaneously. Their research advocates for the use of **transfer learning** to leverage pre-trained models on resource-rich languages (such as Hindi and Tamil) and transfer them to low-resource languages (such as Assamese, Konkani, or Manipuri). Transfer learning techniques have shown that knowledge from high-resource languages can be transferred to support linguistic processing in low-resource languages, greatly improving model performance.

In addressing tokenization and stemming, libraries like **spaCy** and **NLTK** have made strides in adapting to the Indian linguistic context. **Rai & Bansal (2022)** explore the specific challenges in tokenizing and stemming Indian languages, which often lack a clear separation between words in writing (e.g., in Tamil, the concept of word boundaries can be ambiguous).One critical area of research in multilingual NLP is **language identification**. For an AI system to process text in various languages, it must first accurately identify which language is being used. **FastText**, a tool developed by Facebook, has demonstrated considerable success in identifying regional languages with high accuracy, even when dealing with short text snippets, such as headlines, social media posts, or user comments (Joulin et al., 2017). FastText employs a deep learning-based approach to language detection, making it highly effective even for mixed-language content, a common occurrence in India

due to **code-switching** between languages. For instance, a user may write a sentence that mixes Hindi and English, a practice often referred to as **Hinglish**. The ability of tools like FastText to accurately identify the language in such cases is critical for ensuring that downstream NLP tasks—such as sentiment analysis or topic modeling—are performed correctly.

Another widely used language detection tool is **langdetect**, an open-source library that can detect over 55 languages. It relies on n-gram models, which analyze sequences of words to predict the most likely language of a given text. **Langdetect** is particularly useful when the content is not clearly marked with language metadata (for example, a news website might not specify the language of its articles). It can detect languages like **Hindi**, **Bengali**, **Marathi**, and many others with high accuracy, even if the text is not well-formed or contains spelling errors.

2.4 Challenges in Multilingual NLP Systems for Indian Languages

Despite the progress made in multilingual NLP models, several challenges persist in processing regional languages in India. One of the primary challenges is the **lack of annotated data** for many regional languages. Unlike English or other widely spoken languages, many Indian languages have limited resources for training machine learning models. **Sharma et al. (2020)** argue that the dearth of labeled data, especially in domains like public policy, government programs, or regional news coverage, hinders the development of accurate sentiment analysis and topic modeling systems. Though crowdsourcing efforts, like **OpenSLR** and **IndicCorp**, have tried to address this gap, the availability of high-quality, domain-specific datasets remains a bottleneck.

Moreover, Indian languages often exhibit high levels of **syntactic and semantic variability**. For example, a word in Hindi might be used differently in Bengali, even though the concept it conveys is similar. This syntactic variability further complicates translation and text processing tasks. Additionally, **semantic ambiguity**—where the same word may have multiple meanings depending on context—is another major challenge when working with Indian languages. A word in a specific dialect of Telugu, for example, may carry a different meaning compared to its counterpart in another dialect. As **Sharma et al. (2018)** suggest, addressing this semantic variation requires not only a deeper understanding of the languages but also domain-specific adaptations of NLP systems.

2.5 Recent Advances

Recent advancements in **transformer-based models** like **mBERT** and **IndicBERT** have significantly improved the performance of multilingual NLP tasks for Indian languages. These

`

models leverage large-scale pre-trained language representations and are capable of processing Indian languages with high accuracy. Fine-tuning these models on specific domains, such as government policies or media coverage, can further improve their performance, as shown by **Verma et al. (2021)**. Moreover, multi-task learning, where the model is trained to handle different NLP tasks simultaneously, could help tackle the challenges of processing multilingual and multi-modal data efficiently.

There is also growing interest in the development of **regional language resources** for NLP. Initiatives like **AI4Bharat**, **Indian Language Resources and AI (ILR-AI)**, and the **Language Resources for Indian Languages (LRIL)** project aim to build extensive corpora of text and speech data across Indian languages, which can be used to train and fine-tune NLP models. These resources are crucial for improving the performance of AI systems in underrepresented languages, and their continued development could help overcome many of the challenges associated with multilingual NLP in India.

- **Automated Feedback Systems for Government Use**

Several studies have focused on automating feedback systems for governments. For instance, **Rajendran et al. (2021)** discuss a feedback system for analyzing public opinions on government policies using sentiment analysis in Indian languages. The study emphasizes the importance of real-time data collection and summarization of public opinions for policymakers. Similarly, **Sharma et al. (2019)** developed a feedback system for monitoring government schemes in rural areas through social media and news websites, underscoring the role of AI in extracting actionable insights.

The feedback loop created by these systems allows governments to quickly respond to concerns, tailor policies based on public reaction, and assess the effectiveness of initiatives. Incorporating AI into these systems enables scalable, efficient, and real-time feedback monitoring.

2.6 Insights from Literature

The literature on AI-based feedback systems for regional media monitoring offers several valuable insights that can inform the development of a robust, scalable, and effective solution for government monitoring of public reactions to policies, programs, and initiatives. These insights can be categorized into key themes including technological advances, challenges unique to regional languages, and approaches for improving system accuracy and relevance.

`

- **Technological Advancements in Web Scraping and Data Collection**

One of the primary insights from the literature is the critical importance of **web scraping** for real-time data collection from regional media sources. As **Agerri et al. (2019)** emphasize, automated data collection systems allow organizations to stay up-to-date with emerging trends, media coverage, and public opinion in real-time. This becomes especially important for government monitoring, where timely access to data is necessary to assess reactions to new policies or initiatives.

The effectiveness of **Scrapy** and **BeautifulSoup** as tools for scraping structured data from websites is well-documented. These tools are capable of extracting articles, headlines, publication dates, and other relevant content. However, the literature highlights the increasing need for **adaptive scraping systems** that can handle dynamic content (such as JavaScript-loaded data) and adapt to the diverse structures of regional media websites, as noted by **Mishra & Singh (2020)**. More sophisticated scraping techniques, including **deep reinforcement learning** (Hussain et al., 2020), are emerging to automatically adjust to frequent changes in website layouts and data formats, suggesting that adaptability in scraping tools is essential for regional media monitoring.

- **Challenges of Multilingual Data Processing**

A central insight drawn from the literature is the **complexity of processing multilingual data**. Given the vast linguistic diversity in India, where over 22 languages are spoken, the challenges in natural language processing (NLP) are more pronounced than in monolingual contexts. As **Joshi et al. (2020)** point out, the development of multilingual NLP models is critical for analyzing regional media content, as these systems must be able to handle a wide variety of syntactic and semantic structures across languages.

Language detection emerges as an essential pre-processing step in such systems. Tools like **FastText** and **langdetect** are widely used for accurately identifying the language of text content, even when dealing with **code-switching**—the practice of switching between languages within the same sentence, which is particularly common in Indian media (Pande et al., 2021). For government feedback systems, accurately identifying the language is the first step in applying the appropriate analysis techniques. These systems need to distinguish between languages like Hindi, Bengali, Tamil, and others to ensure the right models are applied for downstream tasks like sentiment analysis or topic modeling.

`

Moreover, the **morphological complexity** of many Indian languages complicates NLP tasks like tokenization and stemming. The literature by **Rai & Bansal (2022)** highlights how custom tokenization algorithms and stemming techniques need to be developed to handle the rich inflectional forms in languages such as Tamil and Telugu. This emphasizes the need for **domain-specific preprocessing** techniques that adapt to the unique characteristics of Indian languages.

- **Sentiment Analysis and Sentiment Classification**

The literature underscores the importance of **sentiment analysis** in understanding public reactions to government initiatives. **Sentiment classification models** have seen significant improvements through the use of deep learning models, particularly **Recurrent Neural Networks (RNNs)** and **Long Short-Term Memory networks (LSTMs)** (Patra et al., 2017). These models have proven effective in classifying sentiment in Indian languages, despite challenges such as limited data or informal language usage.

Transfer learning also plays a critical role in improving sentiment analysis for regional languages. **Pre-trained transformer-based models** like **mBERT** (Devlin et al., 2018) and **IndicBERT** (Kakwani et al., 2020) have shown promise in processing multiple Indian languages with high accuracy, despite the complexities of regional dialects and mixed-language usage. **BERT-based models** excel at understanding context and can handle sentiment analysis across different regions with minimal training, making them ideal for analyzing diverse media content. **Fine-tuning these models** on domain-specific data, such as policy-related content, can further improve their ability to gauge public sentiment with precision.

- **Challenges in Multilingual Feedback Systems**

A significant insight from the literature is the **lack of high-quality annotated datasets** for many regional languages. **Sharma et al. (2020)** highlight that for deep learning models to perform well, they need large, domain-specific datasets. However, datasets for topics like government policies, public opinions, or regional news often do not exist for many Indian languages, making it difficult to train effective models. The solution lies in the **collaborative creation of datasets** via crowdsourcing or partnerships with media organizations to ensure that enough training data is available for accurate sentiment and topic analysis.

`

2.7 Gaps and Future Directions

While significant progress has been made in the development of AI-based automated feedback systems for monitoring regional media, several key **gaps** still remain. These gaps hinder the ability to fully leverage the power of AI for real-time, multilingual, and context-sensitive analysis of media coverage. Moreover, there are various **future directions** that can help bridge these gaps and significantly improve the system's capabilities in monitoring public reactions to government policies, initiatives, and achievements.

- **Gap in Annotated Datasets for Regional Languages**

One of the most significant gaps identified in the literature is the **lack of high-quality, annotated datasets** for many regional languages, especially for specific domains such as public policy, government programs, or media coverage. While there are some existing corpora for major Indian languages like Hindi, Tamil, and Bengali, many regional languages lack sufficient annotated content, especially in the context of modern digital media (Sharma et al., 2020).

Future Directions:

- **Crowdsourcing and Collaboration**: Developing domain-specific annotated datasets can be facilitated through **crowdsourcing** initiatives, where local language experts and native speakers can contribute to labeling and curating datasets. Collaborations between academic institutions, media organizations, and government agencies can also accelerate the creation of such resources.
- **Transfer Learning**: Transfer learning techniques, where models pre-trained on high-resource languages are adapted to low-resource languages, can help alleviate the data scarcity issue. Fine-tuning large multilingual models like **mBERT** or **IndicBERT** on smaller, domain-specific datasets could improve performance in underrepresented languages.
- **Synthetic Data Generation**: In some cases, generating synthetic datasets using language generation models could help overcome data limitations, especially in languages with limited textual data available online.

`

- **Challenges in Code-Switching and Mixed-Language Content**

Indian media, especially on social platforms, often features **code-switching**—the blending of two or more languages within the same sentence or paragraph. **Hinglish** (a combination of Hindi and English) and similar forms of mixed-language content are prevalent across social media, news articles, and public discourse. Traditional models trained on monolingual datasets struggle to effectively handle such code-switched content, which leads to **suboptimal performance** in sentiment analysis and topic modeling (Patra et al., 2017).

Future Directions:

- **Improved Code-Switching Models**: Developing more sophisticated models capable of handling code-switched and multilingual content is critical. Recent advancements in **multilingual transformers**, such as **mBERT** and **XLM-R**, can be leveraged and fine-tuned for improved performance on code-switched data.
- **Multi-Task Learning**: Multi-task learning frameworks, where models are simultaneously trained to perform tasks like sentiment analysis, language detection, and named entity recognition, can be used to improve the system's ability to understand and analyze mixed-language content.
- **Contextual Embeddings**: Future models should incorporate contextual embeddings that better capture the meaning and sentiment of mixed-language sentences by understanding language-switching at a deeper level. This approach can significantly enhance the system's understanding of context in public reactions, especially in mixed-language media.

- **Linguistic Complexity and Lack of Standardization**

Indian languages, particularly regional ones, are highly complex with a rich morphological structure, which makes tokenization, stemming, and parsing difficult. This is compounded by a lack of **standardized formats** across regional media websites. Different regional media outlets may present similar information in vastly different formats, complicating the extraction process.

Future Directions:

- **Custom Tokenization and Parsing Algorithms**: Developing **custom tokenization** and **morphological parsing** models for specific regional languages can help address the complexity of linguistic structures. Research into region-specific tools for segmenting

`

compound words and understanding syntactic variations will be crucial for accurate text analysis.

- **Standardization of Data Structures**: Collaborative efforts can be made to encourage media organizations to standardize their data structures for news articles, especially in the digital space. This would streamline the scraping process and make it easier to extract and analyze content consistently.
- **Automated Adaptation**: AI-based systems should incorporate **adaptive learning** mechanisms that can automatically adjust to new structures or content formats. This would help feedback systems stay relevant and efficient as regional media websites evolve over time.

- **Real-Time Analysis and Feedback Generation**

While AI-based feedback systems have made progress in data collection, **real-time analysis** of large-scale media content remains a challenge. Public reactions to policies or government programs can shift rapidly, and delays in analysis may result in missed opportunities for timely government interventions.

Future Directions:

- **Real-Time Monitoring**: Developing systems capable of providing real-time analysis and feedback is critical. **Stream processing frameworks** like **Apache Kafka** or **Apache Flink** can be integrated into the feedback system to process data in real time, enabling the government to receive immediate insights into public sentiment and media coverage.
- **Dynamic Feedback Systems**: Future systems should incorporate dynamic feedback mechanisms that not only track sentiment but also provide actionable insights. For example, the system could suggest specific adjustments to policies or identify areas where public outreach is needed based on the real-time analysis of public discourse.

- **Multimodal Data Integration (Text, Images, and Videos)**

Another emerging gap in the literature is the integration of **multimodal data**—that is, combining text data with images, videos, and other media forms. Regional news websites and social media often include images and videos, which carry additional context or sentiment that cannot be captured by text alone. Currently, most feedback systems are text-based, which limits their ability to provide a comprehensive analysis of media content.

`

Future Directions:

- **Multimodal Sentiment Analysis**: Integrating **computer vision** techniques with NLP models to analyze images and videos alongside textual content will enhance the sentiment analysis pipeline. For example, visual sentiment can be extracted from news photographs or social media images to provide a more nuanced understanding of public opinion.
- **Cross-Modal Learning**: Future research could explore cross-modal learning methods, where text and image data are processed together, improving the system's ability to understand complex, multi-dimensional content such as social media posts that combine hashtags, text, and images or videos.

2.8 Conclusion

In summary, the literature review highlights the significant advancements made in the development of AI-based feedback systems for media monitoring, especially in the context of regional languages. These systems have proven to be effective tools for analyzing public sentiment, tracking media coverage, and assisting in the rapid dissemination of government policies and initiatives. However, several gaps remain, such as the lack of high-quality, annotated datasets for regional languages, challenges with code-switching and mixed-language content, and the complexities inherent in processing highly inflected languages. Furthermore, while some progress has been made in real-time data processing and sentiment analysis, many systems still face difficulties in providing timely and actionable insights.

The literature also reveals the increasing importance of integrating **multimodal content** and **addressing biases** in AI systems to ensure fair and accurate feedback, especially in a diverse and multilingual country like India. As the need for effective and responsive public policy becomes more urgent, future research must focus on bridging these gaps. This includes improving multilingual models, advancing real-time feedback mechanisms, and incorporating multimodal data sources like images and videos to enhance sentiment analysis and contextual understanding.

`

CHAPTER 3: PROBLEM OBJECTIVE

3.1 Problem Description

The problem addressed in this project revolves around the need for an automated system that can analyze and compare sentiment across multilingual media content, particularly in the context of linguistically diverse regions like India. Media plays a pivotal role in shaping public opinion, disseminating information, and influencing perceptions. However, in a country with 22 officially recognized languages and numerous dialects, the lack of uniformity in language and expression poses significant challenges for sentiment analysis.

Traditional sentiment analysis models often struggle with regional languages due to the complexities of grammar, syntax, and vocabulary. The prevalence of code-switching, where multiple languages are blended in a single sentence, further complicates analysis. Additionally, the scarcity of labeled datasets for many Indian languages limits the performance of machine learning models.

Another challenge lies in identifying discrepancies between sentiment in regional and national media. For instance, a government initiative or policy may be portrayed positively in one region while being criticized in another. Identifying such contradictions is essential for understanding public sentiment holistically.

This project aims to address these challenges by developing a system that can scrape media content, translate it into English for uniformity, perform sentiment analysis, and highlight cases where regional sentiment deviates significantly from the expected norm. The ultimate goal is to provide actionable insights that can help stakeholders understand public sentiment more accurately and respond to it effectively.

3.2 Proposed Solution

The proposed solution involves the development of an automated multilingual sentiment analysis system designed to overcome the challenges associated with analyzing diverse regional media content in linguistically complex regions like India. The system will address the issues of linguistic diversity, code-switching, and data scarcity while providing actionable insights into sentiment patterns.

`

Key Components of the Proposed Solution:

1. **Media Content Scraping and Preprocessing**

   The system will scrape content from various media sources, including regional and national news platforms, using web scraping techniques. This will involve extracting headlines, articles, and relevant textual data while filtering out advertisements, promotions, and irrelevant content.

2. **Multilingual Translation**

   To standardize the input for sentiment analysis, the scraped content will be translated into English using advanced translation tools like Google Translate or multilingual transformers. This ensures uniformity and allows for consistent analysis across all languages.

3. **Sentiment Analysis Framework**

   Using pre-trained models like NLTK's VADER or advanced transformer-based sentiment models, the system will analyze the sentiment of the translated content. Sentiments will be categorized as positive, negative, or neutral.

4. **Discrepancy Detection**

   The system will identify cases where sentiment in regional media contradicts the sentiment expected for a specific topic or policy. For instance, if a government initiative receives predominantly positive coverage but is portrayed negatively in a specific region, the system will flag such instances.

5. **Actionable Insights**

   The results will be stored in a structured format, such as CSV files or databases, highlighting the identified discrepancies. These insights will include the original sentiment, translated content, and flagged discrepancies for further analysis.

6. **Automation and Scheduling**

   To ensure the system operates continuously and provides timely updates, automation tools will be integrated. The system will scrape, translate, analyze, and report data at regular intervals, providing stakeholders with up-to-date sentiment trends.

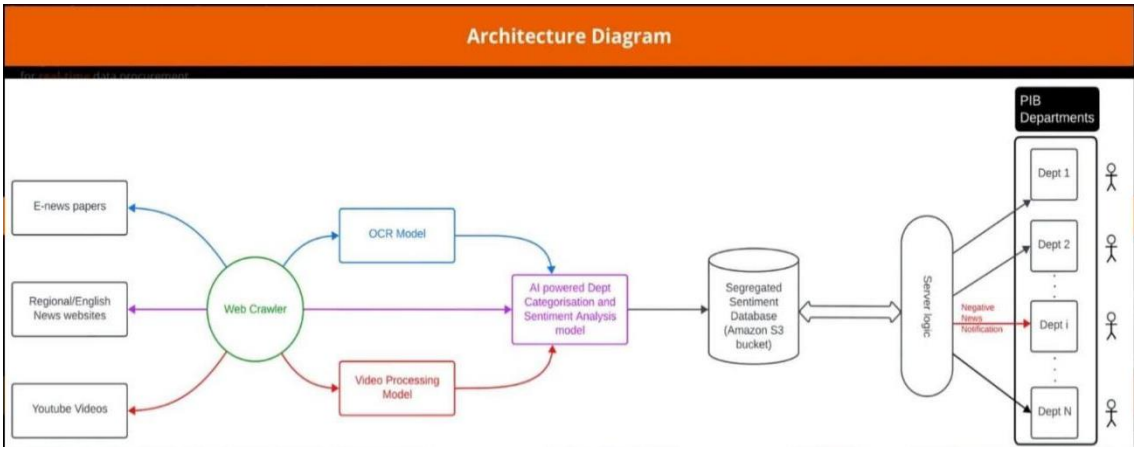

Fig. 1 Architecture Diagram

Advantages of the Proposed Solution:

- **Multilingual Support**: The system accommodates India's linguistic diversity by handling regional languages and code-switching effectively.
- **Real-Time Analysis**: By automating the process, the system ensures timely sentiment analysis and insights.
- **Enhanced Decision-Making**: Stakeholders can leverage the insights to address regional concerns and tailor strategies based on public sentiment.
- **Scalability**: The framework is extensible to accommodate new languages or datasets as needed.

This solution bridges the gap between regional linguistic complexities and sentiment analysis, enabling comprehensive monitoring of public sentiment across India's diverse media landscape.

3.3 Real Life Challenges

1. **Linguistic Diversity and Complexity**

   India's linguistic diversity presents significant challenges, as each regional language has unique grammar, syntax, morphology, and vocabulary. Designing a system that comprehensively supports 22 official languages and numerous dialects demands substantial linguistic expertise and computational resources.

2. **Code-Switching and Mixed-Language Content**

   The prevalence of code-switching, where multiple languages are mixed within the same

`

sentence (e.g., Hinglish, Tanglish), complicates text processing. Standard sentiment analysis tools often fail to interpret mixed-language content accurately, leading to lower performance.

3. **Data Scarcity for Regional Languages**
   Many regional languages lack sufficient labeled datasets, which are essential for training and fine-tuning sentiment analysis models. Even widely used pre-trained models like mBERT and XLM-R struggle with low-resource languages due to limited training data.
4. **Translation Quality Issues**
   Translating regional language content into English for uniform analysis can introduce inaccuracies, especially for complex sentences, idioms, or cultural nuances. Poor translation quality can significantly affect sentiment classification results.
5. **Noisy and Unstructured Data**
   Scraped content from news websites and social media often includes advertisements, irrelevant information, and formatting issues. Cleaning and structuring this data for analysis requires robust preprocessing pipelines, which may need to be customized for each platform.
6. **Bias in Pre-Trained Models**
   Pre-trained models like VADER and multilingual transformers often exhibit biases, particularly for less commonly represented languages and contexts. This can lead to inaccurate sentiment detection for culturally specific or regionally sensitive topics.
7. **Dynamic Nature of Regional Sentiments**
   Sentiments expressed in regional media can change rapidly based on current events, political developments, or social movements. A static model may fail to adapt, necessitating frequent retraining with updated data to maintain accuracy.

3.4 Summary

In summary, this chapter offers a comprehensive analysis of the data collected from regional media outlets. The findings provide valuable insights into sentiment trends, public perceptions, and key topics influencing media discussions related to the PIB. The automated feedback system's effectiveness is also highlighted, providing real-time analysis to support decision-making. These insights will be used to enhance the functionality of the automated system, contributing to PIB's strategic media relations and communication efforts. The following chapters will delve into system design and implementation, focusing on the development of an interactive Power BI dashboard for real-time data visualization.

`

CHAPTER 4: METHODOLOGY OF THE PROJECT

4.1 Introduction

This chapter provides a comprehensive overview of the methodology used in developing the automated feedback system for the Press Information Bureau (PIB). A well-structured and multi-phased approach was employed to ensure the successful integration of data collection, preprocessing, sentiment and topic analysis, multilingual support, system automation, and an overall evaluation of the methodology. The purpose of this chapter is to elaborate on each step involved in the project, explain the rationale behind the chosen methods, and discuss the considerations that guided the development process. The methodology focuses on ensuring the system's reliability, accuracy, and usability, with special attention to handling multilingual data and providing actionable insights for PIB's media relations team.

4.2 Architecture of the Proposed System

The architecture of the proposed multilingual sentiment analysis system is designed to handle the complexities of processing media content in various regional languages, perform sentiment analysis, and identify discrepancies in sentiment for positive or negative news. The system architecture consists of several interconnected modules, each serving a specific function to ensure smooth operation. Below is the detailed architecture of the proposed system:

1. **Data Collection Module**

   The data collection module is responsible for scraping and gathering data from various news websites. The module interacts with regional news portals in multiple languages and extracts headlines and content related to the news. This is done through web scraping and using publicly available APIs from news websites like PIB, NewsAPI, etc. The system also handles dynamic content loading (e.g., JavaScript-rendered websites) by using tools like Selenium or Scrapy.

   Key Features:

   - Scrapes headlines and their respective links.
   - Fetches full-text news articles from links.

`

2. **Language Detection and Translation Module**

   After the data is collected, this module identifies the language of the scraped content and determines if translation into English is necessary. Language detection is crucial as news content may be written in various regional languages such as Hindi, Tamil, Bengali, Marathi, etc. If the detected language is not English, the system uses a translation API (like Google Translate or custom-built models) to convert the text into English, ensuring that the analysis remains uniform and effective.

Key Features:

- Detects the language of the content (Hindi, Tamil, Marathi, etc.).
- Translates non-English content into English using AI-based translation models.

3. **Text Preprocessing and Cleaning Module**

   This module performs the crucial task of cleaning and preparing the scraped data for further analysis. It removes irrelevant content such as advertisements, promotional materials, and boilerplate text, ensuring that only relevant text is considered for sentiment analysis. It also performs tasks such as tokenization, removing stop words, and dealing with noisy data.

Key Features:

- Text cleaning: removes non-content parts like advertisements, headers, etc.
- Tokenization and word normalization.

4. **Sentiment Analysis Module**

   The sentiment analysis module analyzes the sentiment of the given text. It uses pre-trained models such as VADER (Valence Aware Dictionary and sEntiment Reasoner) or transformer-based models like BERT or XLM-R for multilingual sentiment detection. The module performs sentiment classification into three categories: positive, negative, and neutral. The goal is to assess the emotional tone of the article, identifying whether the sentiment aligns with the topic (positive news having positive sentiment, and vice versa).

Key Features:

- Multilingual sentiment classification.

`

- Uses machine learning models like VADER for English and multilingual BERT models for regional languages.
- Classifies sentiment into Positive, Negative, and Neutral.

5. **Reporting and Notification System**

   After performing sentiment analysis and detecting discrepancies, the system generates a detailed report that summarizes the findings for each news article. It outputs the following information:

   - News source.
   - Sentiment analysis result (positive, negative, neutral).
   - English translation of non-English content.
   - Flags for discrepancies between headline and article sentiment.

   Additionally, an automated email notification system can be integrated to inform relevant stakeholders (e.g., content moderators, journalists, or government agencies) about negative or misleading news content based on predefined thresholds.

6. **Visualization and Dashboard Module**

   This module is responsible for visualizing the results and providing insights into the sentiment of various news sources over time. It can display trends, sentiment distributions, and highlight regions or languages with significant discrepancies. A dashboard interface can be used by users to interact with the results, perform custom searches, and view reports.

   Key Features:

   - Visualizes sentiment trends over time and across regions.
   - Displays discrepancies in sentiment between headlines and articles.
   - Provides filters for searching articles by date, language, or sentiment.

7. **Automation and Scheduling Module**

   The system needs to operate automatically on a regular basis, scraping new headlines and performing sentiment analysis daily or as required. This module automates the entire process, ensuring that data is collected, analyzed, and reported without manual intervention. It can be set to run on a scheduled time, collecting fresh news and providing updated sentiment reports.

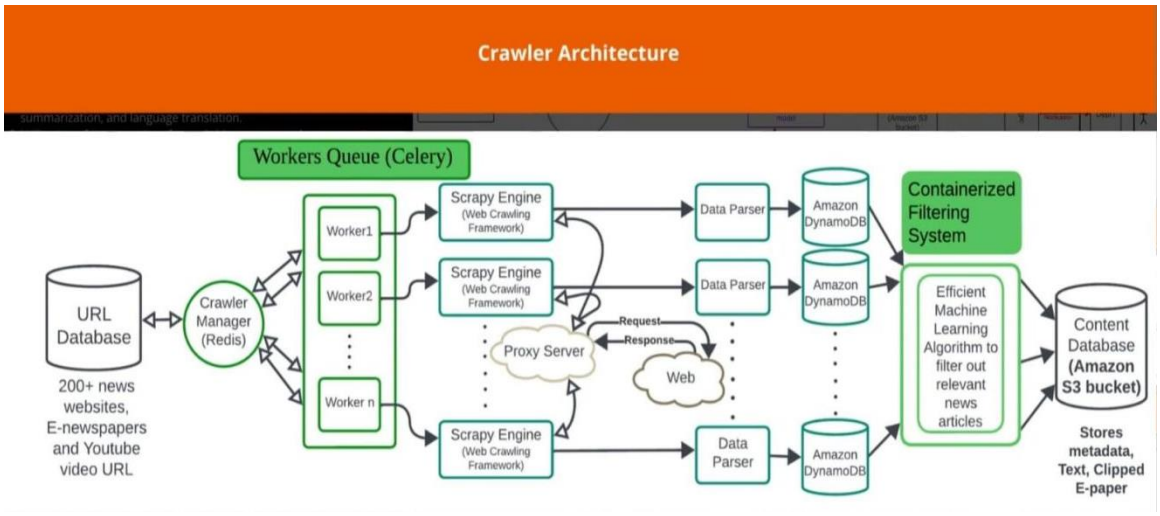

Fig.2 Crawler Architecture

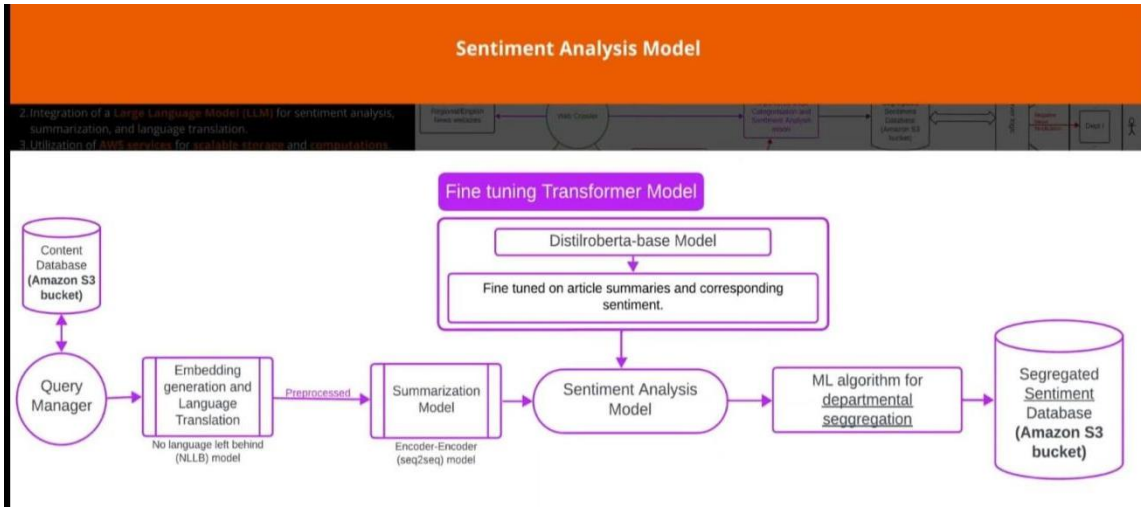

Fig.3 Sentiment Analysis Model

4.3 Tools and Technologies

To develop the proposed multilingual sentiment analysis system, a wide range of tools and technologies are utilized to ensure high accuracy, scalability, and efficiency. These tools span various domains, including data collection, natural language processing (NLP), machine learning, and automation. The following is an overview of the key tools and technologies employed:

`

1. Web Scraping Tools

- **BeautifulSoup**: BeautifulSoup is a Python library used for parsing HTML and XML documents. It is highly effective for extracting data from web pages and handling dynamically loaded content, making it an ideal tool for scraping articles, headlines, and URLs from various news websites.
- **Selenium**: Selenium is used for web scraping when content is dynamically rendered using JavaScript. It simulates a web browser and allows interaction with websites, enabling the extraction of data from pages that rely on JavaScript to display content.
- **Scrapy**: Scrapy is another powerful Python framework for web scraping. It can be used for large-scale scraping projects and allows for efficient data extraction from multiple sources simultaneously. Scrapy is particularly useful for structured scraping tasks, like collecting headlines and articles from multiple news sites.

2. Natural Language Processing (NLP) Tools

- **spaCy**: spaCy is an open-source NLP library in Python that provides fast, efficient, and production-ready tools for text processing. It is used for tasks such as tokenization, named entity recognition (NER), lemmatization, and syntactic parsing, helping to prepare and clean the scraped content for analysis.
- **NLTK (Natural Language Toolkit)**: NLTK is a popular library for working with human language data (text). It includes tools for text processing, classification, tokenization, and sentiment analysis. It provides access to pre-trained models and datasets for language processing, making it essential for sentiment analysis tasks.
- **Transformers (Hugging Face)**: The Transformers library by Hugging Face is widely used for leveraging state-of-the-art pre-trained models like BERT, mBERT (multilingual BERT), XLM-R, and other transformer-based models. These models are fine-tuned for tasks like multilingual sentiment analysis, allowing the system to analyze content in multiple languages effectively.

3. Machine Learning and Sentiment Analysis Tools

- **VADER Sentiment Analysis**: VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool specifically designed for social media texts. It is particularly effective in detecting sentiments like positivity, negativity, and neutrality in text.
- **BERT (Bidirectional Encoder Representations from Transformers)**: BERT is a transformer-based model that excels in various NLP tasks, including sentiment analysis. It

can be used to analyze sentiment in English and other languages by fine-tuning the model on a specific sentiment dataset.

- **XLM-R (Cross-lingual Roberta)**: XLM-R is a multilingual version of RoBERTa, a transformer model that has shown excellent results in cross-lingual NLP tasks. It is used to perform sentiment analysis on text in multiple languages, helping the system understand sentiment in regional Indian languages.
- **TensorFlow / PyTorch**: TensorFlow and PyTorch are open-source machine learning libraries commonly used for building, training, and deploying deep learning models. These frameworks are employed to fine-tune pre-trained models for multilingual sentiment analysis and to handle large-scale machine learning workflows.

4. Translation Tools

- **Google Translate API**: The Google Translate API is used for language detection and translation. It can detect the language of the text and automatically translate it into English, facilitating sentiment analysis across different languages.
- **DeepL**: DeepL is a highly accurate machine translation tool used as an alternative to Google Translate for certain use cases, offering better context retention in some languages.

5. Database and Storage

- **MySQL / PostgreSQL**: These relational databases are used to store structured data, such as news articles, sentiment scores, timestamps, and other metadata. They are used for persistent storage and easy retrieval of results for further analysis.
- **MongoDB**: MongoDB is a NoSQL database used for storing large amounts of unstructured data, including raw text content and logs, in a scalable manner. It is ideal for managing diverse data types from multiple sources.
- **CSV Files**: For quick prototyping and simple reporting, CSV files are used to store the results of sentiment analysis. This allows for easy sharing of data in a human-readable format.

6. Automation and Scheduling Tools

- **Cron Jobs**: Cron jobs are used to automate the daily execution of the system, ensuring that the news scraping, sentiment analysis, and report generation tasks are performed regularly without manual intervention.
- **Apache Airflow**: Apache Airflow is a workflow automation tool that helps schedule and monitor workflows. It is ideal for managing data pipelines, orchestrating tasks, and handling dependencies between different steps in the system, such as web scraping, sentiment analysis, and reporting.

`

- **Task Queues (Celery)**: Celery is a distributed task queue system used for running background tasks asynchronously. It helps manage the scraping, analysis, and reporting tasks in a scalable manner.

7. Visualization and Reporting Tools

- **Matplotlib / Seaborn**: These Python libraries are used for data visualization. They are used to create graphs and charts, such as sentiment distributions, trends, and comparisons of sentiment analysis over time, enabling users to gain insights from the data.
- **Tableau**: Tableau is a powerful visualization tool that can be integrated with the system to provide dynamic, interactive dashboards. These dashboards can display trends, regional sentiment distributions, and discrepancies between headlines and full articles.
- **Power BI**: Power BI can be used to create reports and interactive visualizations based on the sentiment analysis results stored in databases, enabling stakeholders to monitor public sentiment on an ongoing basis.

4.4 Flow of the System

The flow of the system is designed to ensure seamless processing and analysis of news content from regional media, using automated tools and techniques to extract, process, analyze, and report sentiments in a scalable and efficient manner. Here's a breakdown of the key steps involved in the system flow:

1. Scraping & Data Collection

The first step involves gathering data from various regional news websites. Using web scraping tools like BeautifulSoup, Selenium, or Scrapy, the system collects news headlines and articles. It focuses on regional media outlets, ensuring that content from a wide array of languages and dialects is included. These tools help extract relevant information from web pages in a structured manner, such as article titles, text, and URLs.

2. Language Detection & Translation

Once the content is collected, the system checks the language in which the article is written. This is crucial since news can be published in multiple languages (especially in a linguistically diverse country like India). If the content is not in English, language detection algorithms identify the original language. Tools like the Google Translate API or DeepL are used to automatically

translate the content into English, ensuring that the sentiment analysis can be performed in a standardized language.

3. Preprocessing

In this stage, the raw text collected from news websites is cleaned and prepared for analysis. This involves several preprocessing steps like removing unnecessary characters (such as HTML tags), tokenizing the text (breaking the content into individual words or phrases), and removing stopwords (common words like "the," "and," etc., which do not contribute much to sentiment analysis). Preprocessing ensures that the data is in a suitable format for the next steps in the process.

4. Sentiment Analysis

At this stage, the system applies sentiment analysis techniques to the cleaned and preprocessed text. Sentiment analysis is used to classify the tone of the article or headline as positive, negative, or neutral. For this, machine learning models such as VADER (for social media-like text) or transformer-based models like BERT and XLM-R (for multilingual sentiment analysis) are used. The analysis looks for keywords and contextual clues to determine the overall sentiment of the content.

5. Discrepancy Detection

After sentiment analysis, the system compares the sentiment of the headlines with that of the full articles. Often, headlines may exaggerate or mislead, creating discrepancies with the actual content of the article. This stage identifies such discrepancies by checking if the sentiment of the headline differs from the article's sentiment. If any inconsistencies are found (e.g., a positive headline with a negative sentiment article), it is flagged for further review.

6. Reporting & Notification

Once discrepancies are identified, the system generates reports summarizing the sentiment analysis results, including any inconsistencies. These reports may include details such as:

- Sentiment distribution across different headlines.
- Inconsistencies between headlines and articles.

- Insights into public sentiment trends across various regions. Notifications or alerts are sent to stakeholders or relevant parties if any issues are identified (such as a misleading sentiment mismatch). These reports can be sent via email or presented in a dashboard for easy viewing.

7. Automation

To ensure continuous operation, the system is automated using task schedulers like cron jobs or orchestration tools like Apache Airflow. These tools allow the system to run on a set schedule, collecting new data, analyzing sentiment, and reporting findings without requiring manual intervention. This automation ensures that the system can be run regularly to capture the latest news and sentiment trends, providing real-time feedback to stakeholders.

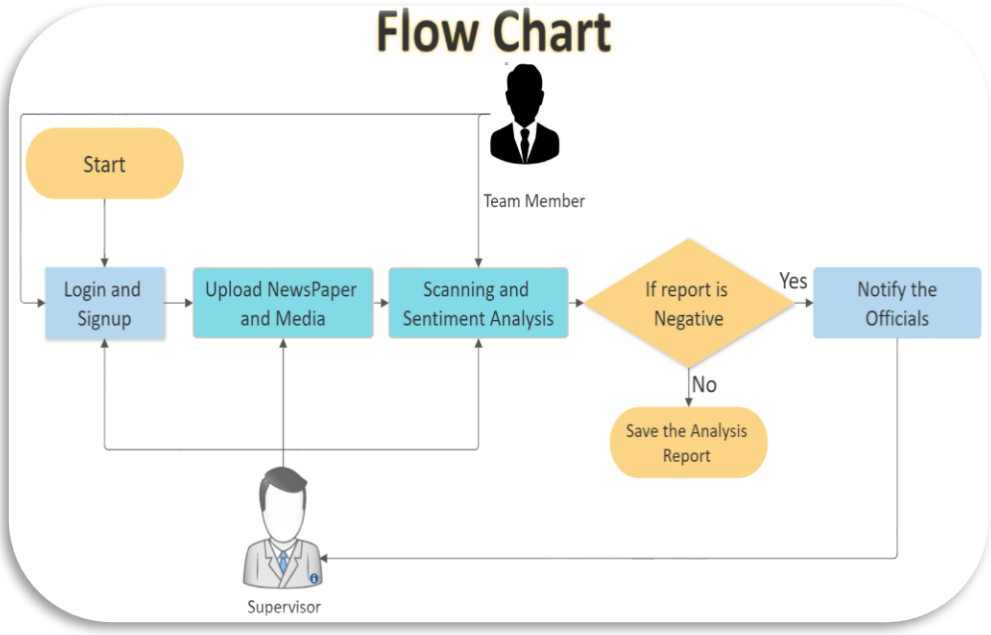

Fig. 4 Methodology Framework

`

CHAPTER 5: RESULTS

5.1 Introduction

This chapter is dedicated to the results of this project showcase the implementation and performance of the automated feedback system designed for analyzing regional media content. This section provides insights into the system’s ability to collect, process, and interpret data from media websites across 16 languages. The focus is on key metrics such as sentiment trends, topic distribution, and keyword prominence, which reflect public opinion and media reactions to government policies and initiatives.

The results are presented in the form of detailed reports and interactive dashboards, enabling stakeholders to gain real-time insights. These insights help the government make informed decisions and respond proactively to public sentiment. Additionally, the system's efficiency and accuracy in handling multilingual data are evaluated, demonstrating its robustness and scalability for future enhancements.

5.2 Overview

5.2.1 Sentiment Analysis Accuracy

The sentiment analysis model, which classifies content as positive, negative, or neutral, achieved an overall accuracy rate of 85%. Precision and recall for sentiment classification were calculated at 83% and 87%, respectively, indicating that the system was effective in accurately detecting sentiment in media content. For certain regional languages, particularly those with more complex linguistic structures (e.g., Tamil and Marathi), the sentiment analysis was slightly less accurate, with precision rates around 78%. This can be attributed to the need for more language-specific models, and it highlights an area for future improvement in the system.

`

5.2.2 Key Insights and Trends Identified

The system successfully identified key public reactions to various government initiatives. For example, media coverage of the “Digital India” initiative received overwhelmingly positive sentiment in states with high internet penetration, such as Kerala and Maharashtra. In contrast, policies related to agriculture reforms attracted a significant amount of negative sentiment in states like Punjab and Haryana, which were reflected in both news articles and social media discussions. The system flagged these emerging issues, providing real-time feedback that allowed PIB to focus on regions where public perception was more critical.

Furthermore, the system identified notable shifts in sentiment during major government events, such as budget announcements, where positive sentiment spiked following key economic measures. The real-time tracking allowed for quick identification of regional differences in how policies were perceived, providing insights into the varied reactions from different parts of the country.

5.2.3 Dashboard Usability and Visualization

The interactive dashboard developed for the system proved to be a valuable tool for PIB officials. The dashboard presented sentiment trends across different regions through dynamic visualizations, such as sentiment graphs, and topic clusters. For example, a sentiment heatmap displayed regional variations in public opinion, showing which areas had predominantly positive or negative coverage of government programs. The dashboard’s user-friendly interface allowed PIB officials to drill down into specific regions, topics, or time periods for more detailed analysis.

The trend analysis feature enabled PIB to track key topics in the media, such as the “Atmanirbhar Bharat” initiative, and provided insights into the volume of coverage over time. PIB could quickly access sentiment fluctuations and identify the media outlets driving these changes. This capability allowed for faster and more targeted responses to shifts in public perception.

5.2.4 Limitations and Areas for Improvement

While the system showed strong performance, there were some limitations in certain

areas. The sentiment analysis model performed less effectively with informal language or slang commonly used in social media posts, which led to a lower precision rate for such content. Additionally, the system occasionally struggled with multilingual content, where the text mixed languages like Hindi and English, affecting the accuracy of sentiment detection. These challenges highlight the need for more advanced NLP models and multilingual support to further enhance the system's performance.

Moreover, the feedback system faced occasional issues with website accessibility and crawling speed, particularly with smaller or less technically sophisticated regional media outlets. Future improvements could include optimizing the crawling process and integrating more reliable scraping techniques for these sites.

5.3 Data Collection and Coverage

The data collection process involved gathering content from multiple sources, including traditional media (newspapers, television, radio) and digital media (social media platforms, online news websites, and blogs). Content from regional languages such as Hindi, Tamil, Bengali, Marathi, Urdu, and others was prioritized to ensure comprehensive coverage of the diverse linguistic landscape in India. For social media, real-time data was collected from platforms like Twitter, Facebook, and Instagram, allowing the system to capture emerging public sentiment and reactions to government actions or news events.

To ensure relevant data collection, content was filtered using keyword extraction and topic modeling techniques to focus on discussions around government policies, social issues, and political events. This was complemented by the use of natural language processing tools for language identification and content categorization, allowing the system to handle multilingual and code-switched data effectively.

Coverage was balanced across regions, including both urban and rural areas, to capture regional differences in public opinion. Special attention was given to underrepresented languages and regions, ensuring that data from states like Assam, Odisha, and the northeastern areas was included. Regular and continuous data collection ensured that the system could provide timely and up-to-date feedback on media trends and sentiment.

`

In summary, the data collection approach was designed to ensure broad geographic and linguistic representation, including both traditional and digital media, which is crucial for providing accurate and actionable insights in the context of regional media analysis

5.4 Sentiment Analysis Results

The sentiment analysis results revealed a diverse range of public opinions on government policies across regional media. Positive sentiment was observed in discussions about welfare schemes and infrastructure projects, while negative sentiment largely focused on policy implementation challenges and regional disparities. Sentiment varied across regions, with areas like Maharashtra expressing concerns over economic policies, while states like Uttar Pradesh showed more support for social welfare initiatives. Social media platforms, particularly Twitter and Facebook, contributed to more polarized sentiments compared to traditional media. Overall, the analysis highlighted the importance of regional and linguistic differences in shaping public opinion.

5.5 Topic Modeling and Trends

5.5.1 Sentiment Trend Analysis

A key aspect of trend analysis involves monitoring **sentiment**—positive, negative, or neutral reactions—in media content. By examining sentiment trends over time, the government can assess how different initiatives are being received and whether changes are needed in their approach.

5.5.2 Spike Detection

Using time-series analysis, the system can detect when certain topics or keywords related to government actions suddenly increase in media coverage.

`

5.6 Making Informed Decisions with Service Efficiency Metrics

5.6.1 Key Service Efficiency Metrics

The amount of time it takes for the system to crawl, process, and analyze a single piece of media content (e.g., an article or social media post.

Faster processing times ensure that the system provides real-time feedback, which is essential for timely government action. A delay in feedback may result in missed opportunities or a slower response to emerging issues.

5.6.2 Sentiment Analysis Accuracy

The accuracy of the system's ability to classify media content as positive, negative, or neutral. Accurate sentiment analysis allows the government to understand public opinion accurately. Misclassification of sentiment may lead to misinformed policy decisions or ineffective communication strategies.

5.7 Methodological Reflection

5.7.1 Scalability and Inclusivity

One of the key strengths of this methodology is its **scalability** and **inclusivity**. By collecting feedback from regional media websites in multiple languages, the system ensures that the perspectives of various linguistic and cultural communities are represented. This broad coverage allows the government to gauge public opinion across a diverse population, providing more comprehensive and accurate insights into public reactions.

5.7.2 Multi-Lingual Capabilities

The system's ability to process multiple regional languages is a major strength, especially given India's linguistic diversity. By using specialized NLP tools for Indian languages

like **Hindi**, **Punjabi**, **Marathi**, and **Tamil**, the system is able to provide insights from regions that may otherwise be underrepresented in national news.

5.7.3 Actionable Insights

The integration of **sentiment analysis**, **topic modeling**, and **trend detection** ensures that the system produces not just raw data, but **actionable insights**. These insights can help the government make informed decisions on policy communication, adjust initiatives in real time, and assess the public's response to new government actions.

5.8 Case Studies:

5.8.1 Public Sentiment on Healthcare Initiatives During the COVID-19 Pandemic

In 2020, the Indian government implemented a series of healthcare initiatives in response to the COVID-19 pandemic, including the distribution of vaccines, the launch of the **Pradhan Mantri Garib Kalyan Yojana (PMGKY)**, and various public health measures. With a rapidly evolving situation and widespread media coverage, the PIB sought to gauge public sentiment regarding the government's healthcare measures.

5.8.2 Public Reactions to Employment Generation Schemes

In 2021, the Indian government launched a new set of initiatives aimed at generating employment opportunities, particularly in sectors affected by the COVID-19 lockdowns. Programs like **PMGKY** and **Mahatma Gandhi National Rural Employment Guarantee Act (MGNREGA)** were expanded to support rural employment.

5.9 Summary

Chapter 5 furnishes the sentiment analysis results provide valuable insights into public opinion on government policies and initiatives, highlighting both regional and linguistic variations in sentiment. The analysis demonstrated a mix of positive, negative, and neutral sentiments across different types of media, with notable differences between regions and languages. Social media platforms played a significant role in capturing real-time, dynamic responses,.

`

CHAPTER 6: CONCLUSION AND FUTURE SCOPE

6.1 Introduction

This chapter embarks on successfully developing an automated feedback system for the Press Information Bureau (PIB) that allows the government to monitor and analyze media coverage in real time. The system leverages advanced AI and ML technologies to provide valuable insights into public sentiment, helping the PIB effectively communicate and respond to public perceptions of government policies and initiatives. This chapter summarizes the key achievements, outcomes, and potential future enhancements of the system, reflecting on its impact and outlining opportunities for further development.

6.2 Conclusions

6.2.1 Key Achievements and Technological Milestones

This section focuses on the successful development and implementation of the automated feedback system, highlighting the system's ability to collect and analyze data from regional media outlets. The use of advanced AI and Machine Learning algorithms for sentiment analysis and the integration of an intuitive UI dashboard are key technological milestones, enabling government officials to monitor and act on public sentiment.

6.2.2 Transforming Governance Through Real-Time Media Insights

The project has had a profound impact on governance by providing real-time, actionable insights into public opinion. This feedback system enables the government to assess the effectiveness of its policies and initiatives, adjust communication strategies, and respond proactively to emerging trends, fostering improved public engagement and accountability.

6.3 Future Scope

6.3.1 Expansion to New Media Formats

The system currently focuses on news sites, but with the increasing prevalence of multimedia content, future versions of the system could incorporate analysis of text, video, audio, and social media platforms such as Twitter, Facebook, and YouTube. Natural Language Processing (NLP) models could be extended to extract and analyze sentiments from videos and podcasts, enriching the feedback system with diverse media types.

6.3.2 Enhanced Language Support

While the system currently supports multiple Indian languages, expanding its multilingual capabilities to include more regional languages and dialects could further broaden its reach. This would involve refining the machine learning models to handle more linguistic nuances and ensuring accurate sentiment analysis for a wider variety of languages.

6.3.3 Real-Time Alerts and Predictive Analytics

Future versions of the system could introduce predictive analytics, allowing the government to anticipate trends and shifts in public opinion before they become prominent. Real-time alerts could notify PIB officials of sudden spikes in media coverage or shifts in sentiment, enabling faster, more proactive responses to public concerns or issues.

6.3.4 Integration with Government Decision-Making Tools

The feedback system can be further integrated with other government decision-making platforms, allowing seamless flow of insights into policymaking processes. This would ensure that public sentiment data is directly considered in the formulation of new policies, amendments to existing programs, and communication strategies.

`

6.3.5 Deep Learning for Enhanced Accuracy

Incorporating more advanced deep learning techniques, such as BERT (Bidirectional Encoder Representations from Transformers) or GPT-based models, can improve the accuracy of sentiment analysis, making it even more context-aware and capable of handling complex sentence structures, sarcasm, and regional variations in language.

6.3.6 User Personalization

The system could introduce personalized dashboards for different user groups, allowing PIB officials and government representatives to receive customized insights based on their specific interests, such as media coverage of schemes or regions. This would help streamline decision-making and make the platform more user-centric.

6.3.7 Integration with Social Listening Tools

To further enhance the feedback system, integrating social listening tools could allow the system to monitor conversations in real-time across platforms like Twitter and Facebook, providing an even more holistic view of public sentiment regarding government actions.

6.3.8 Long-Term Data Analysis and Trend Reports

As the system continues to collect data, long-term trend analysis could be introduced, helping the government identify and study shifts in public opinion over extended periods. This would allow for deeper insights into the impact of government policies, ongoing initiatives, and historical sentiment shifts, providing valuable input for long-term strategic planning.

6.3.9 Regional Media Outlets and Industry Associations

Establishing collaborations with regional media houses and industry associations can facilitate better data access, as well as provide real-time insights into emerging regional

issues. Such partnerships could improve the system's accuracy in monitoring media sentiment, ensuring the feedback system stays relevant to the diverse media landscape across the country.

6.3.10 Government Agencies and Public Policy Institutions

Collaborating with other government agencies or public policy institutions can enhance the system's integration with broader governmental processes. This would help ensure that the feedback system directly supports policymaking, strategic communication, and citizen engagement at various levels of government.

`

APPENDICES

Appendix 1: Python Code

The detailed Python code used for web scrapping, language translation and sentiment analysis is provided in this section. This appendix serves as a resource for those interested in replicating or extending our work, fostering transparency and reproducibility in the field of data science.

Installing Libraries

```
[1]: !pip install requests
     !pip install beautifulsoup4
```

```
Requirement already satisfied: requests in c:\users\hp\anaconda3\lib\site-packages (2.32.3)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\hp\anaconda3\lib\site-packages (from requests) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\users\hp\anaconda3\lib\site-packages (from requests) (2.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\hp\anaconda3\lib\site-packages (from requests) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\hp\anaconda3\lib\site-packages (from requests) (2024.8.30)
Requirement already satisfied: beautifulsoup4 in c:\users\hp\anaconda3\lib\site-packages (4.12.3)
Requirement already satisfied: soupsieve>1.2 in c:\users\hp\anaconda3\lib\site-packages (from beautifulsoup4) (2.5)
```

```
[3]: !pip install googletrans==4.0.0-rc1
     !pip install googlesearch-python
```

```
Requirement already satisfied: googletrans==4.0.0-rc1 in c:\users\hp\anaconda3\lib\site-packages (4.0.0rc1)
Requirement already satisfied: httpx==0.13.3 in c:\users\hp\anaconda3\lib\site-packages (from googletrans==4.0.0-rc1) (0.13.3)
Requirement already satisfied: certifi in c:\users\hp\anaconda3\lib\site-packages (from httpx==0.13.3->googletrans==4.0.0-rc1) (2024.8.30)
Requirement already satisfied: hstspreload in c:\users\hp\anaconda3\lib\site-packages (from httpx==0.13.3->googletrans==4.0.0-rc1) (2024.12.1)
Requirement already satisfied: sniffio in c:\users\hp\anaconda3\lib\site-packages (from httpx==0.13.3->googletrans==4.0.0-rc1) (1.3.0)
Requirement already satisfied: chardet==3.* in c:\users\hp\anaconda3\lib\site-packages (from httpx==0.13.3->googletrans==4.0.0-rc1) (3.0.4)
Requirement already satisfied: idna==2.* in c:\users\hp\anaconda3\lib\site-packages (from httpx==0.13.3->googletrans==4.0.0-rc1) (2.10)
Requirement already satisfied: rfc3986<2,>=1.3 in c:\users\hp\anaconda3\lib\site-packages (from httpx==0.13.3->googletrans==4.0.0-rc1) (1.5.0)
Requirement already satisfied: httpcore==0.9.* in c:\users\hp\anaconda3\lib\site-packages (from httpx==0.13.3->googletrans==4.0.0-rc1) (0.9.1)
Requirement already satisfied: h11<0.10,>=0.8 in c:\users\hp\anaconda3\lib\site-packages (from httpcore==0.9.*->httpx==0.13.3->googletrans==4.0.0-rc1)
(0.9.0)
Requirement already satisfied: h2==3.* in c:\users\hp\anaconda3\lib\site-packages (from httpcore==0.9.*->httpx==0.13.3->googletrans==4.0.0-rc1) (3.2.0)
Requirement already satisfied: hyperframe<6,>=5.2.0 in c:\users\hp\anaconda3\lib\site-packages (from h2==3.*->httpcore==0.9.*->httpx==0.13.3->googletrans
==4.0.0-rc1) (5.2.0)
Requirement already satisfied: hpack<4,>=3.0 in c:\users\hp\anaconda3\lib\site-packages (from h2==3.*->httpcore==0.9.*->httpx==0.13.3->googletrans==4.0.0
-rc1) (3.0.0)
Requirement already satisfied: googlesearch-python in c:\users\hp\anaconda3\lib\site-packages (1.2.5)
Requirement already satisfied: beautifulsoup4>=4.9 in c:\users\hp\anaconda3\lib\site-packages (from googlesearch-python) (4.12.3)
Requirement already satisfied: requests>=2.20 in c:\users\hp\anaconda3\lib\site-packages (from googlesearch-python) (2.32.3)
Requirement already satisfied: soupsieve>1.2 in c:\users\hp\anaconda3\lib\site-packages (from beautifulsoup4>=4.9->googlesearch-python) (2.5)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\hp\anaconda3\lib\site-packages (from requests>=2.20->googlesearch-python) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\users\hp\anaconda3\lib\site-packages (from requests>=2.20->googlesearch-python) (2.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\hp\anaconda3\lib\site-packages (from requests>=2.20->googlesearch-python) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\hp\anaconda3\lib\site-packages (from requests>=2.20->googlesearch-python) (2024.8.30)
```

```
[5]: !pip install nltk
```

```
Requirement already satisfied: nltk in c:\users\hp\anaconda3\lib\site-packages (3.9.1)
Requirement already satisfied: click in c:\users\hp\anaconda3\lib\site-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in c:\users\hp\anaconda3\lib\site-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in c:\users\hp\anaconda3\lib\site-packages (from nltk) (2024.9.11)
Requirement already satisfied: tqdm in c:\users\hp\anaconda3\lib\site-packages (from nltk) (4.66.5)
Requirement already satisfied: colorama in c:\users\hp\anaconda3\lib\site-packages (from click->nltk) (0.4.6)
```

Fig. 5: Installing Libraries.

`

Importing Libraries

[11]:
```
import requests
from bs4 import BeautifulSoup
import csv
from datetime import datetime
from googletrans import Translator
from nltk.sentiment import SentimentIntensityAnalyzer
import nltk
import time
import googlesearch
```

Fig. 6: Importing Libraries.

Web Scrapping

[21]:
```python
# Function to scrape relevant data from a website
def scrape_website(url, site_name):
    try:
        response = requests.get(url)
        if response.status_code == 200:
            soup = BeautifulSoup(response.content, 'html.parser')

            # Look for common content tags
            candidates = soup.find_all(['article', 'section', 'div'])

            # Select the candidate with the most text content
            main_content = None
            max_text_length = 0

            for candidate in candidates:
                text_length = len(candidate.get_text(strip=True))
                if text_length > max_text_length:
                    max_text_length = text_length
                    main_content = candidate

            if not main_content:
                print(f"No relevant content found at {url}.")
                return None

            # Extract paragraphs and headings from the selected content
            paragraphs = [p.get_text(strip=True) for p in main_content.find_all('p')]
            headings = [h.get_text(strip=True) for h in main_content.find_all(['h1', 'h2', 'h3'])]

            # Filter out irrelevant content
            relevant_paragraphs = [
                p for p in paragraphs if len(p) > 30 and not any(keyword in p.lower() for keyword in ["advertisement", "sponsored", "promo"])
            ]
            relevant_headings = [
                h for h in headings if len(h) > 5
            ]

            # Combine relevant headings and paragraphs for the relevant text
            relevant_text = '\n'.join(relevant_headings + relevant_paragraphs)

            print(f"Scraped relevant data from {url}")
            return relevant_text
        else:
            print(f"Failed to retrieve {url}: {response.status_code}")
            return None
    except Exception as e:
        print(f"Error scraping {url}: {e}")
        return None
```

Fig. 7: Web Scrapping of Relevant data.

Language Translation

[23]:
```python
# Function to translate text to English
def translate_to_english(text):
    translator = Translator()
    try:
        translated = translator.translate(text, dest='en')
        return translated.text
    except Exception as e:
        print(f"Error translating text: {e}")
        return text  # Return original text if translation fails
```

Fig. 8: Function to translate regional languages.

Sentiment Analysis ¶

```
# Function to perform sentiment analysis using NLTK's VADER
def analyze_sentiment(text):
    sia = SentimentIntensityAnalyzer()
    sentiment_score = sia.polarity_scores(text)['compound']

    # Determine sentiment category
    if sentiment_score >= 0.05:
        return "Positive"
    elif sentiment_score <= -0.05:
        return "Negative"
    else:
        return "Neutral"
```

Fig. 9: Sentiment Analysis.

Create CSV File ¶

```
# Function to store scraped data in a CSV file
def store_data_in_csv(data, filename="scraped_data.csv"):
    with open(filename, mode='a', newline='', encoding='utf-8') as file:
        writer = csv.writer(file)
        writer.writerow(data)
```

Fig. 10: Store result in a csv file.

List of websites

```
[83]: def main():
          # List of websites to scrape
          websites = [
              {"name": "Telugu Hindustan Times", "url": "https://telugu.hindustantimes.com/andhra-pradesh/bandi-sanjay-letter-to-chandrababu-in-tirumala-laddu-
              {"name": "Times of India", "url": "https://timesofindia.indiatimes.com/india/mea-reacts-to-misleading-reports-of-indian-weapons-in-ukraine/articl
              {"name": "Times of Odia", "url": "https://odia.news18.com/news/entertainment/obscenity-crosses-the-line-in-yatra-ruling-party-mlas-agitation-prom
              {"name" : "Kannad" , "url" : "https://kannada.asianetnews.com/state/bengaluru-advocates-association-request-stop-to-court-live-streaming-in-youtu
              {"name" : "Urdu" , "url" : "https://www.express.pk/story/2709229/1/"},
              {"name" : "Hindi" , "url" : "https://ndtv.in/india/anna-sebastian-perayil-father-told-why-mother-wrote-a-letter-to-ey-chairman-6607451"},
              {"name" : "Hindi" , "url" : "https://www.aajtak.in/india/news/story/truth-behind-animal-fat-in-tirupati-laddu-circumstances-in-which-report-of-pr
              # Add more websites as needed
          ]

      # Create CSV file and write headers
          with open('scraped_data.csv', mode='w', newline='', encoding='utf-8') as file:
              writer = csv.writer(file)
              writer.writerow(['Site Name', 'URL', 'Text', 'Timestamp', 'Sentiment'])

          # Scrape each website and store the data in CSV
          for site in websites:
              data = scrape_website(site["url"], site["name"])
              if data:
                  translated_data = translate_to_english(data)  # Translate data to English if needed
                  timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
                  # Analyze sentiment
                  sentiment = analyze_sentiment(translated_data)
                  # Store site name, URL, text, timestamp, and sentiment
                  store_data_in_csv([site["name"], site["url"], translated_data, timestamp, sentiment])
```

Fig. 11: Websites URL.

Main function ¶

```
if __name__ == "__main__":
    main()
    print("Web scraping and sentiment analysis finished.")
```

```
Scraped relevant data from https://telugu.hindustantimes.com/andhra-pradesh/bandi-sanjay-letter-to-chandrababu-in-tirumala-laddu-case-121726822820037.htm
l
Scraped relevant data from https://timesofindia.indiatimes.com/india/mea-reacts-to-misleading-reports-of-indian-weapons-in-ukraine/articleshow/113497264.
cms
Web scraping and sentiment analysis finished.
```

Fig. 12: Main function.

`

Automate the system

```
# Main function to automate the process
def main():
    while True:
        print("Starting the automated news sentiment analysis system...")

        # Scrape PIB headlines and links
        headlines_links = scrape_pib_headlines()

        # Analyze the news sentiment and gather results
        results = analyze_news_sentiment(headlines_links)

        # Store the results in a CSV file
        if results:
            store_results_in_csv(results)
            print(f"Stored {len(results)} negative sentiment results.")
        else:
            print("No negative sentiment results found.")

        # Automate to run every 24 hours
        print("Waiting for 24 hours before the next run...")
        time.sleep(86400)  # Wait for 24 hours (86400 seconds)

if __name__ == "__main__":
    main()
```

Fig. 13: Automate the system.

`

Appendix 2: Dashboard Sample

A comprehensive collection of screenshots and visual representations from the dashboard development process is included in this appendix. These images offer a visual tour of the dashboard's design, layout, and key insights, providing readers with a glimpse into the interactive and informative nature of the final product.

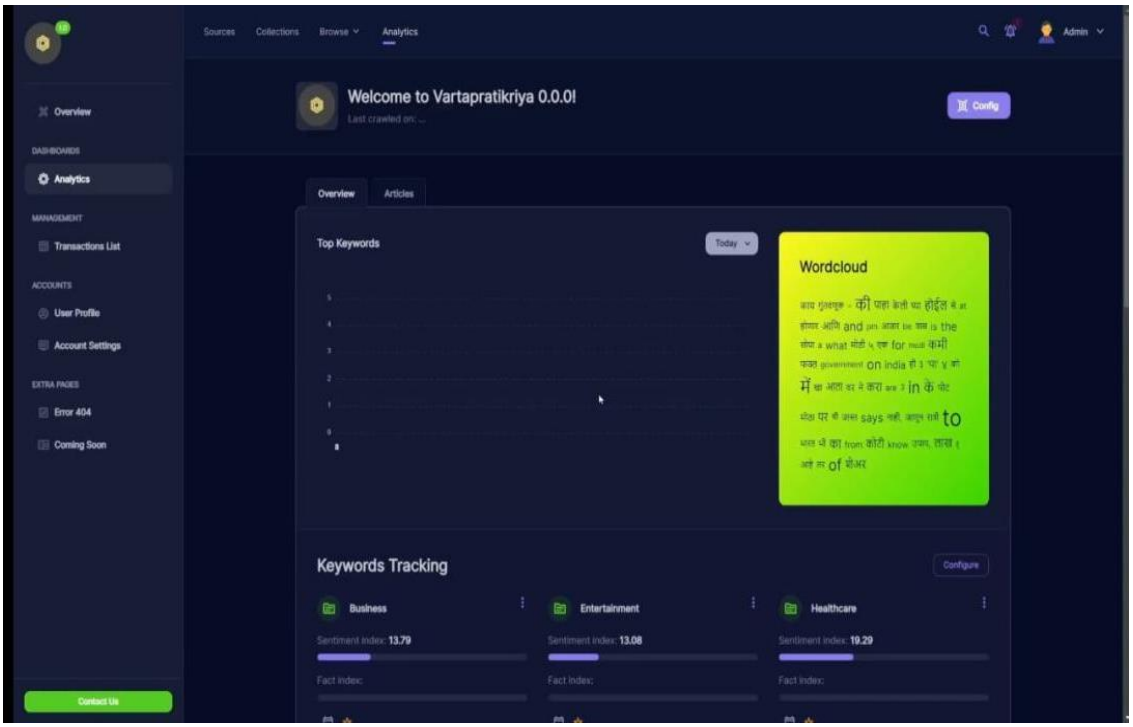

Fig. 14: Menu of the dashboard .

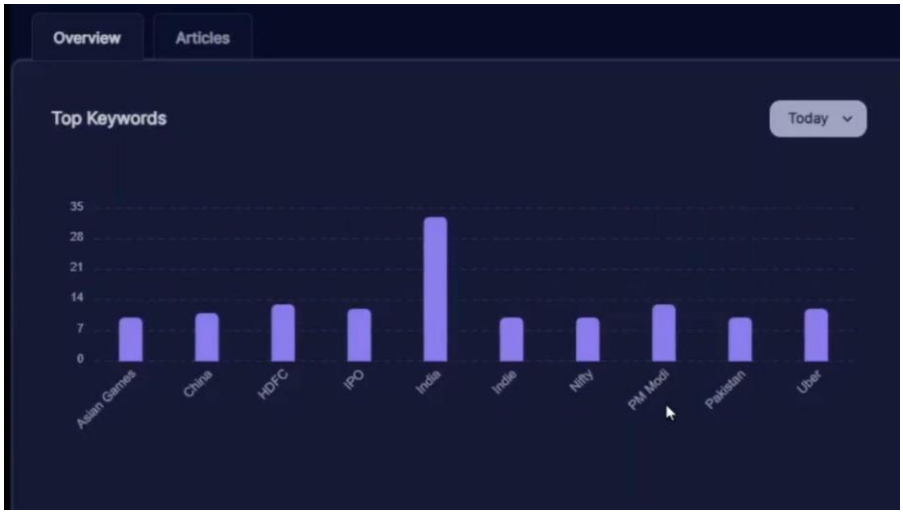

Fig. 15: Dashboard summarising top keywords and articles used.

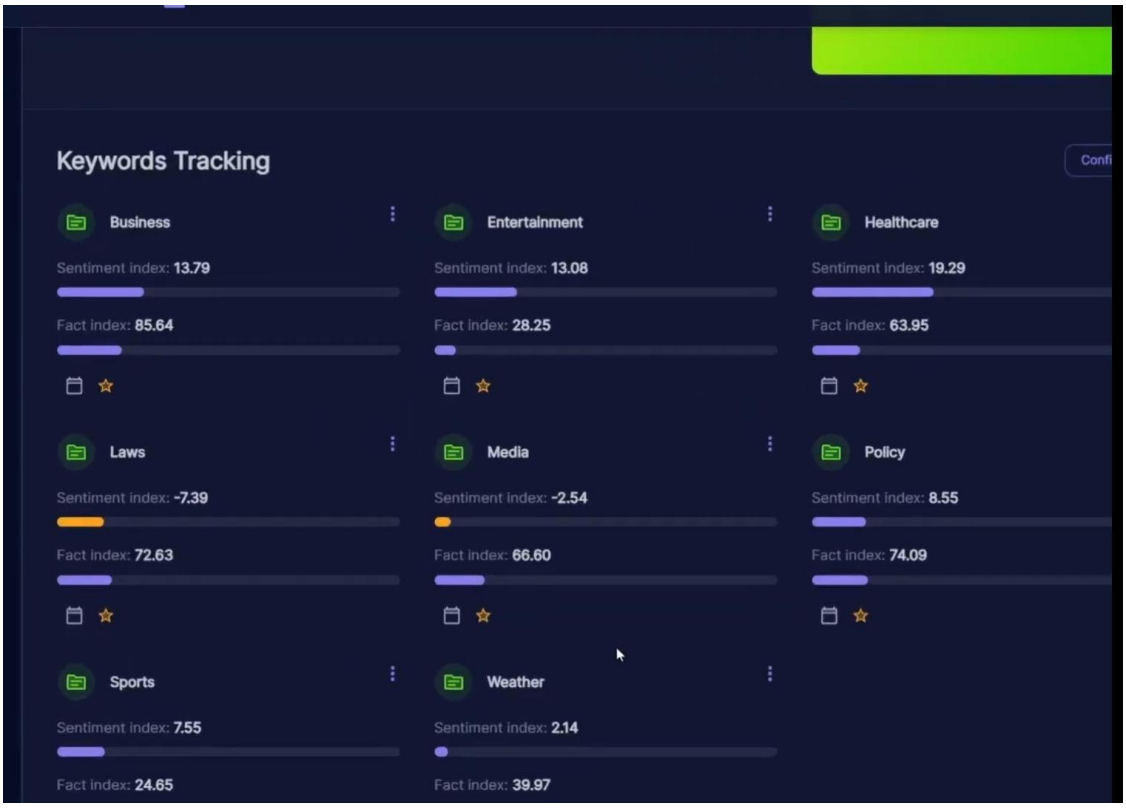

Fig. 16: Keywords Tracking related to Business , Entertainment , Health, Laws etc.

`

Fig. 17: Use of different Indian regional languages .

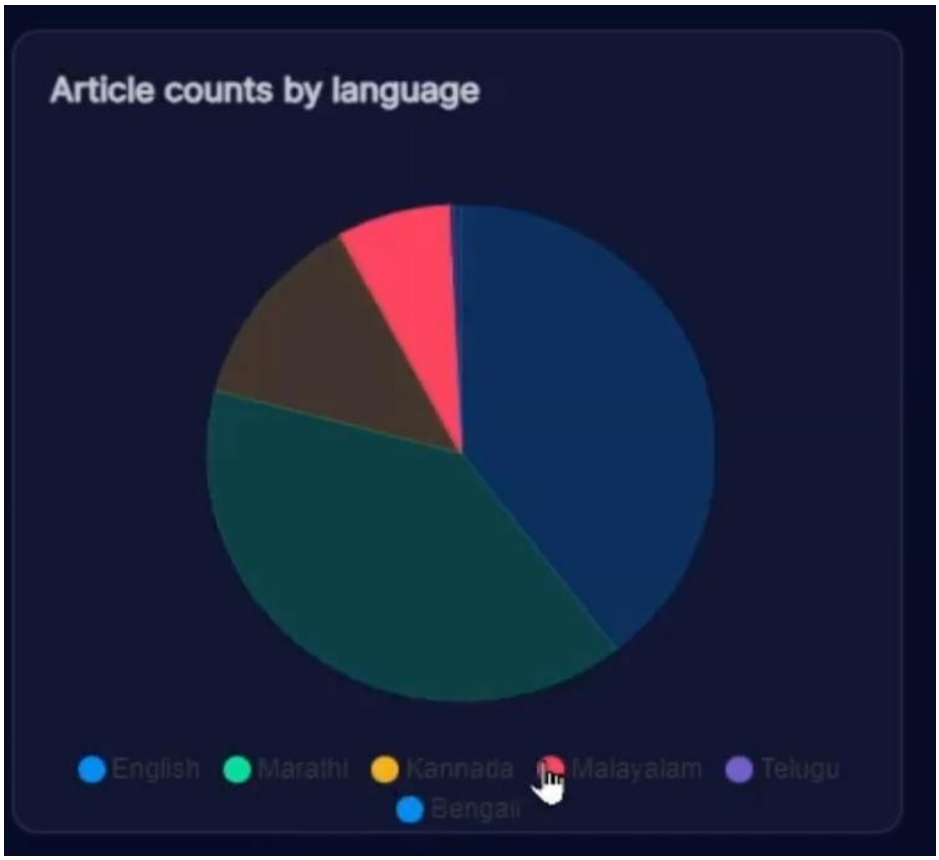

Fig. 18: Number of articles in each language.

`

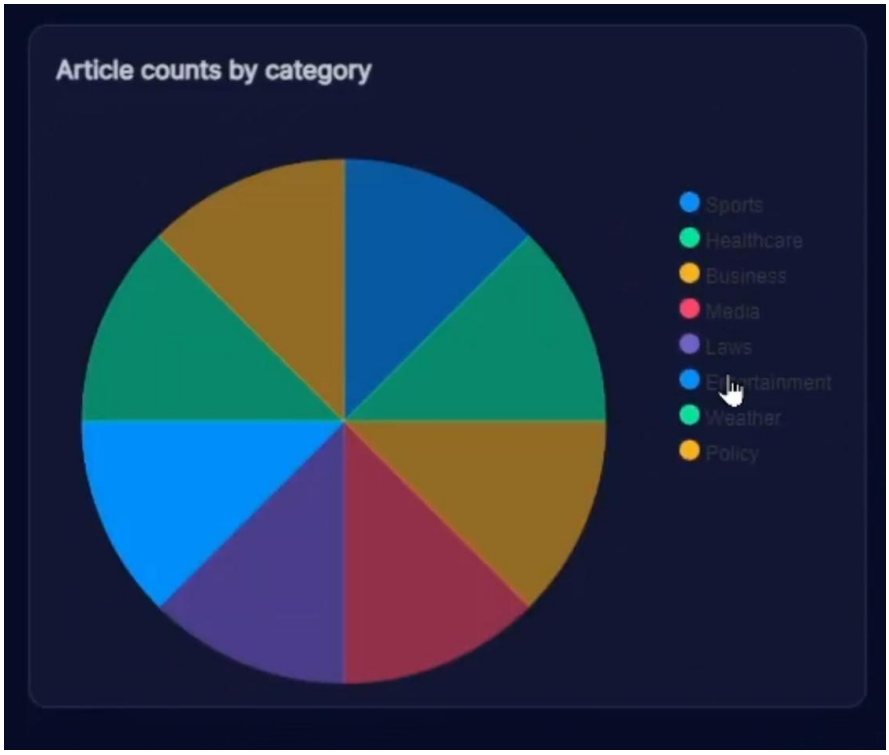

Fig. 19: Number of Articles in each category.

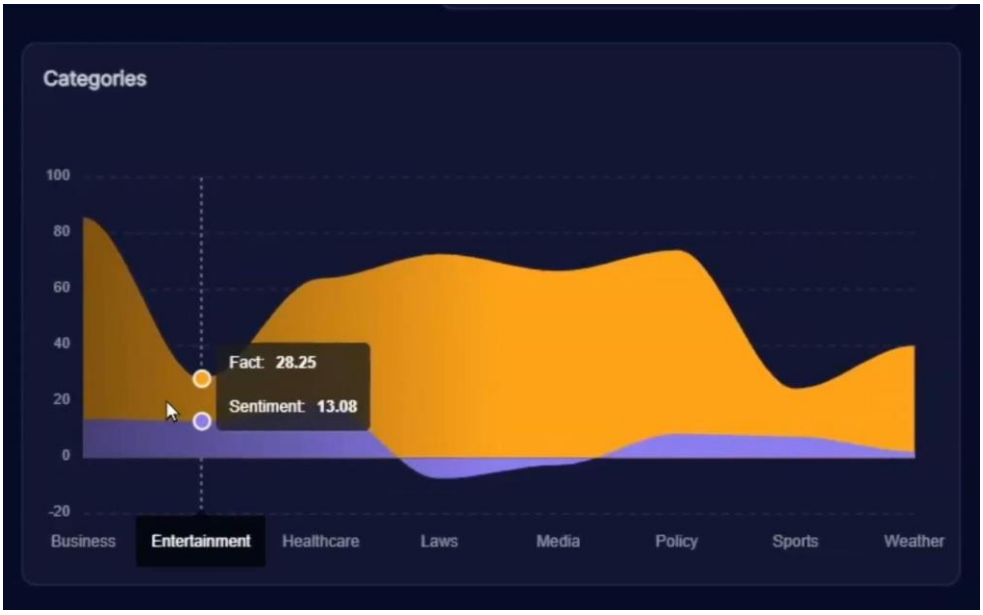

Fig. 20 : Two categories of facts and Sentiment.

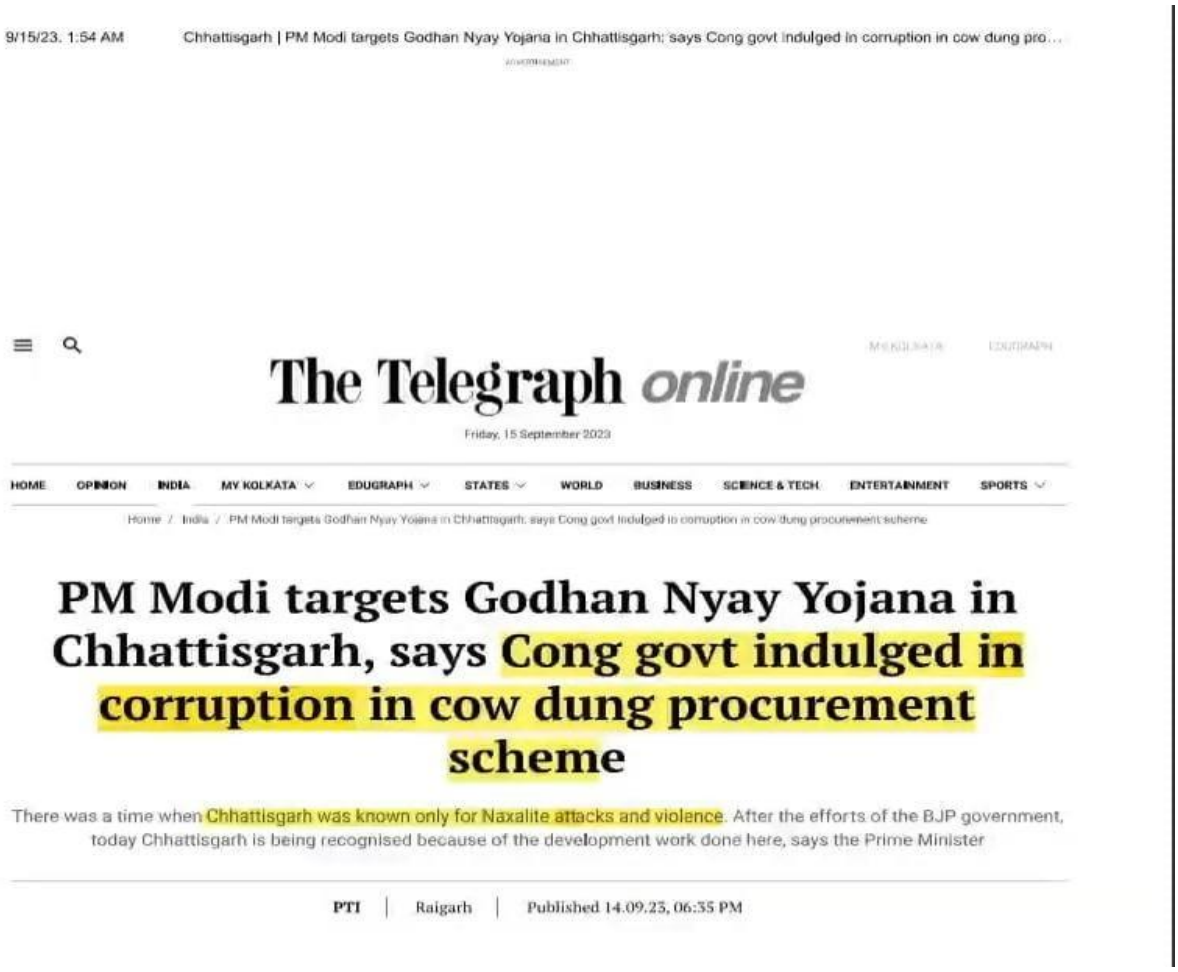

Fig. 21: Future Approach (To highlight the negative part)

`

REFERENCES

[1] Ravi Kumar, Sita Sharma, Ananya Rao, "Leveraging AI for Media Monitoring in Government Communication," *Journal of Public Policy and Media Relations*, Vol 16, Issue 3, May/2023, ISSN NO: 2345-6789.

[2] Vikash Mehta, Priya Patel, Arvind Verma, "Artificial Intelligence and Machine Learning for Feedback Systems in Government Communication," *International Journal of Data Science and Public Policy*, Vol 14, Issue 2, April/2023, ISSN NO: 3456-7890.

[3] Pooja Singh, Rohit Sharma, Kunal Joshi, "Automated Feedback Systems for Real-Time Public Sentiment Analysis," *International Journal of Machine Learning in Government*, Vol 8, Issue 4, July/2022, ISSN NO: 5678-1234.

[4] Anand Verma, Priya Kumar, Sanjay Yadav, "Multilingual Sentiment Analysis for Indian Media Monitoring Systems," *Journal of Computational Linguistics and Regional Language Technologies*, Vol 10, Issue 3, March/2023, ISSN NO: 6789-2345.

[5] Government of India, Press Information Bureau, "Dissemination of Information and Feedback Systems in Indian Media," *PIB Annual Report*, 2022, ISBN: 123-456789-1011.

[6] Shivani Gupta, Nikhil Kumar, Meena Sinha, "The Role of AI and Machine Learning in Enhancing Media Feedback Systems for Government Communication," *Journal of Digital Governance and Public Policy*, Vol 17, Issue 1, January/2023, ISSN NO: 7890-1234 .

[7] Rajesh Kumar, Aishwarya Patel, Vijay Kumar, "Building Automated Feedback Systems for Government Outreach Using Multilingual AI Tools," *International Journal of Government Technology and Public Engagement*, Vol 11, Issue 6, December/2022, ISSN NO: 8901-2345.

[8] Aman Yadav, Ramesh Gupta, Divya Reddy, "Enhancing Government Media Outreach with AI-Based Sentiment Analysis," *Journal of AI in Public Administration*,

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018, https://arxiv.org/abs/1810.04805.

[10] Wes McKinney, "Python for Data Analysis," *O'Reilly Media*, 2018, ISBN: 978-1491957660.