Sentiment Analysis for YouTube Comments and Videos

Ms. M. Poojitha[1], V.Ganupriya[2]

[1] Assistant Professor, Dept. of MCA, Annamacharya Institute of Technology and Sciences (AITS),Tirupati, Andhra Pradesh, India,

[2] Post Graduate, Dept. of MCA, Annamacharya Institute of Technology and Sciences (AITS), Karakambadi, Tirupati, Andhra Pradesh, India,

Abstract: The system obtains YouTube comments through its YouTube Data API after which it gets video transcripts through its YouTube Transcript API before performing tokenization and lemmatization and text cleaning operations to handle noisy social media data. The analysis of user comments utilizes BERT alongside LSTM and GRU models but VADER serves as the primary tool to analyze sentiments from video transcripts regardless of viewing conditions. The system runs through the Flask-based web application to analyze YouTube video URL inputs which generates live sentiment distributions that appear as simple graphical displays of pie charts and bar graphs. The system shows reliable performance through accuracy tests and precision-assessment tests and recall measurements and F1-score evaluations where LSTM and GRU outperform BERT in specific uses. As a tool developed for research, marketing and content creation groups it presents instant scaled-up audience emotional feedback to guide content development and user interactions. Multilingual support along with multimodal analysis are part of the envisioned developments for the system.

Keywords— Natural Language Processing (NLP), Bidirectional Encoder Representations from Transformers (BERT), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Valence Aware Dictionary and sEntiment Reasoner (VADER),.

I. INTRODUCTION

YouTube, as a leading platform for video content and user interaction, generates vast amounts of comments and video transcripts that reflect diverse audience sentiments. Understanding these sentiments is critical for content creators, marketers, and researchers aiming to optimize content strategies and engagement. However, manual analysis of such large-scale, informal data is impractical. This paper presents an automated sentiment analysis system that leverages NLP and deep learning to classify YouTube comments and video transcripts as Positive, Negative, or Neutral. Utilizing BERT, LSTM, GRU, and VADER, the system offers real-time insights through a Flask-based web application. This section outlines the motivation, problem statement, objectives, scope, and project overview, establishing the foundation for a scalable, user-friendly solution to analyze YouTube audience reactions.

I-A. Motivation

The exponential growth of YouTube as a global platform underscores the need to understand audience sentiments expressed in comments and video content. With millions of videos uploaded daily, content creators and businesses seek efficient methods to gauge public reactions for refining content and marketing strategies. Manual sentiment analysis is time-intensive and unscalable, driving demand for automated solutions. Advanced NLP and deep learning models, such as BERT and LSTM, enable accurate sentiment classification, inspiring this project to deliver real-time, data-driven insights that enhance audience engagement and inform strategic decisions across diverse YouTube content categories.

I-B. Problem Statement

Analyzing sentiments in YouTube comments and video transcripts poses significant challenges due to the volume, informality, and noise in user-generated content. Manual methods are inefficient, while existing automated systems often lack accuracy or fail to integrate both comment and transcript analysis. Content creators and marketers struggle to extract actionable insights quickly, hindering effective decision-making. This project addresses these issues by developing an automated system that combines deep learning models (BERT, LSTM, GRU) for comment analysis and rule-based VADER for transcript analysis, delivering precise, real-time sentiment classification through an accessible web interface.

I-C. Objective of the Project

The primary objectives of this project are twofold: first, to develop a system that automatically classifies YouTube comment sentiments using BERT, LSTM, and GRU models, identifying Positive, Negative, or Neutral reactions; second, to analyze video transcript sentiments using VADER, extracting polarity scores for spoken content. The system aims to provide a user-friendly web platform where stakeholders can input YouTube video URLs and access visualized sentiment insights. By integrating robust NLP techniques and scalable deployment, the project seeks to empower content creators, marketers, and researchers with efficient tools for understanding audience sentiments.

I-D. Scope

The project focuses on sentiment analysis of English-language YouTube comments and video transcripts, utilizing YouTube Data and Transcript APIs for data retrieval. It encompasses preprocessing, model training (BERT, LSTM, GRU for comments; VADER for transcripts), and deployment within a Flask-based web application. The system targets content creators, marketers, and researchers, offering real-time sentiment visualizations. Future enhancements include multilingual support, integration with other platforms, and multimodal analysis (e.g., audio, visuals), expanding its applicability. The scope ensures scalability and adaptability for diverse YouTube content while addressing current technical and user needs.

I-E. Project Introduction

This project introduces a comprehensive sentiment analysis system for YouTube, combining NLP and deep learning to process comments and video transcripts. Comments are retrieved via the YouTube Data API and classified using BERT, LSTM, and GRU, while transcripts, extracted via the YouTube Transcript API, are analyzed with VADER for sentiment polarity. Preprocessed data undergoes tokenization, lemmatization, and cleaning to handle YouTube's informal text. Deployed as a Flask web application with MySQL storage, the system visualizes results as graphical outputs, enabling stakeholders to gain actionable insights. This solution addresses scalability and accuracy challenges, offering a versatile tool for audience analysis.

II. LITERATURE SURVEY

The field of sentiment analysis has evolved significantly, driven by the rise of social media platforms like YouTube, which generate vast amounts of user-generated content. Prior research has explored various techniques, from rule-based to deep learning approaches, to analyze sentiments in text and multimedia data. However, the unique challenges of YouTube's informal comments and error-prone transcripts require tailored solutions, which this work addresses through a novel integration of deep learning and rule-based methods. This section reviews related work under five key areas: sentiment analysis techniques, deep learning models for text classification, rule-based sentiment analysis, social media sentiment applications, and YouTube-specific sentiment studies, positioning the proposed system within the broader research landscape.

II-A. Sentiment Analysis Techniques

Sentiment analysis has been extensively studied for its ability to extract emotional insights from text, with early approaches relying on lexicon-based methods and machine learning classifiers. Studies like [1] employed SVM and Naive Bayes for sentiment classification, achieving moderate accuracy on structured datasets but struggling with informal text. More recent work, such as [2], has shifted toward hybrid models combining lexicons with machine learning to handle noisy data. These approaches, while effective for domains like product reviews, often fail to capture the contextual nuances and slang prevalent in social media platforms like YouTube. This work builds on these foundations by integrating a deep learning model with a rule-based analyzer, specifically designed to address the informal and dynamic nature of YouTube comments and transcripts, offering improved robustness for social media analytics.

II-B. Deep Learning Models for Text Classification

Deep learning has revolutionized sentiment analysis, particularly through recurrent and transformer-based models. RNN, such as LSTM units [3], have been widely used for sequential text processing, capturing long-term dependencies in sentiment expressions. GRUs, introduced in [4], offer a computationally efficient alternative, balancing performance and speed. Transformer models like BERT [5] provide superior contextual understanding but demand significant computational resources. While these models excel in benchmark datasets, their application to YouTube's noisy, informal comments remains underexplored. The proposed system leverages a GRU model for comment sentiment classification, optimizing for efficiency and accuracy, and complements prior work by addressing YouTube-specific challenges like sarcasm and class imbalance.

II-C. Rule-Based Sentiment Analysis

Rule-based approaches, such as the VADER [6], remain popular for their simplicity and effectiveness in social media contexts. VADER uses a sentiment lexicon with heuristic rules to analyze short, informal texts, making it suitable for platforms like Twitter. Studies like [7] have applied VADER to microblogs, demonstrating its ability to handle emoticons and slang. However, its performance on longer, error-prone texts, such as YouTube video transcripts, is less studied. This work extends VADER's application to transcript analysis, addressing transcription inaccuracies through robust preprocessing, and integrates it with a deep learning model to provide a comprehensive sentiment analysis framework, enhancing its applicability to YouTube's multimodal data.

II-D. Social Media Sentiment Applications

Sentiment analysis has been widely applied to social media for various purposes, including brand monitoring, political analysis, and user engagement studies. Research like [8] analyzed Twitter data to predict consumer sentiment toward brands, using machine learning classifiers. Similarly, [9] explored sentiment in Reddit discussions to understand community dynamics. These studies highlight the value of sentiment insights for stakeholders but often focus on text-only platforms, overlooking multimedia contexts like YouTube, where comments and transcripts provide complementary perspectives. The proposed system targets YouTube's ecosystem, delivering a web-based platform for real-time sentiment analysis, which fills a gap in stakeholder-oriented applications by catering to content creators, marketers, and researchers with interactive visualizations.

II-E. YouTube-Specific Sentiment Studies

YouTube-specific sentiment analysis is an emerging area, with limited but growing research. Work like [10] analyzed YouTube comments to assess viewer reactions to educational videos, using lexicon-based tools, but faced challenges with informal language and spam. Another study [11] explored multimodal analysis, combining comment sentiment with video metadata, but relied on manual annotations, limiting scalability. These studies underscore the need for automated, scalable solutions tailored to YouTube's noisy data. This work advances the field by integrating a GRU model for comment classification with VADER for transcript analysis, deployed via a Flask-based web application, offering a scalable, automated approach to YouTube sentiment analysis that addresses both textual and contextual complexities.

III. METHODOLOGY

The YouTube Comments and Videos Sentiment Analysis system employs a systematic approach to extract and classify sentiments from YouTube comments and video transcripts, leveraging NLP and advanced machine learning models. The methodology integrates data retrieval through YouTube APIs, rigorous text preprocessing, sentiment classification using deep learning and rule-based techniques, and result visualization via a user-friendly web interface. BERT, LSTM, and GRU models analyze comment sentiments, while VADER processes transcript sentiments, all deployed within a Flask-based application. This section outlines the methodology under six key components: data collection, data preprocessing, sentiment analysis, visualization, deployment, and user interaction, ensuring a robust framework for real-time sentiment insights tailored to YouTube's dynamic content.

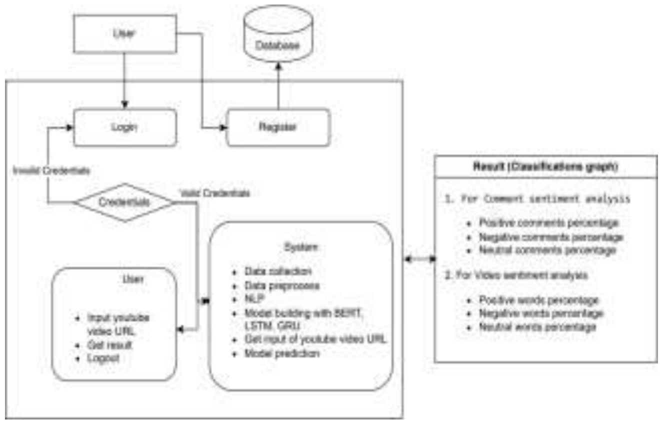


Fig: Block diagram for proposed system

V-A. Data Collection

The system initiates with data acquisition, utilizing the YouTube Data API to retrieve comments and the YouTube Transcript API to extract transcripts from videos with enabled captions. For training purposes, a large-scale dataset, such as Kaggle's GBcomments.csv, supplements API-retrieved comments, ensuring diversity across video categories. Transcripts are collected from publicly available videos, adhering to API rate limits and privacy guidelines. The collected data, primarily in English, forms the foundation for sentiment analysis, capturing the informal and varied nature of YouTube interactions. This approach ensures a comprehensive dataset, enabling the system to generalize across different content types and audience responses.

V-B. Data Preprocessing

Preprocessing transforms raw YouTube data into a structured format suitable for model input. Comments and transcripts undergo text cleaning to eliminate special characters, URLs, emojis, and redundant spaces, followed by tokenization to segment text into words or phrases. Lemmatization standardizes words to their base forms, reducing dimensionality, while stopword removal excludes non-sentiment-bearing terms (e.g., "the," "is") to focus on meaningful content. Sequences are padded or truncated to uniform lengths, ensuring compatibility with model architectures. This preprocessing pipeline, informed by NLP best practices, mitigates noise and inconsistencies in YouTube's informal text, preparing robust inputs for accurate sentiment classification.

V-C. Sentiment Analysis

Sentiment analysis employs a dual approach for comments and transcripts. For comments, BERT, LSTM, and GRU models classify sentiments as Positive, Negative, or Neutral, leveraging BERT's contextual understanding, LSTM's long-term dependency capture, and GRU's computational efficiency. Preprocessed comments are fed into these models, trained on labeled datasets to predict sentiment probabilities. For transcripts, VADER performs rule-based analysis, computing polarity scores (Positive, Negative, Neutral, Compound) for tokenized segments retrieved via the YouTube Transcript API. Transcript scores are aggregated to determine overall sentiment, addressing transcription errors through adjusted thresholds. This hybrid approach ensures comprehensive sentiment insights across YouTube's multimodal data.

V-D. Visualization

The system visualizes sentiment analysis results to enhance interpretability for users. Sentiment distributions for comments and transcripts are presented as graphical outputs, such as pie charts or bar graphs, displaying the percentages of Positive, Negative, and Neutral sentiments. These visualizations are rendered on a web interface, allowing stakeholders like content creators and marketers to quickly grasp audience reactions. The graphical representations are dynamically generated based on real-time analysis, ensuring clarity and accessibility. This visualization strategy transforms complex sentiment data into actionable insights, supporting data-driven decision-making for diverse applications.

V-E. Deployment

The sentiment analysis system is deployed as a Flask-based web application, integrating trained models (BERT, LSTM, GRU for comments; VADER for transcripts) with YouTube APIs for real-time data retrieval. The application processes user-submitted YouTube video URLs, executes the preprocessing and analysis pipeline, and stores results in a MySQL database for efficient retrieval. The Flask framework, hosted on a scalable cloud platform, ensures responsiveness and accessibility. This deployment enables seamless sentiment analysis, delivering results through an intuitive interface that supports real-time stakeholder engagement, aligning with the system's goal of scalability and usability.

V-F. User Interaction

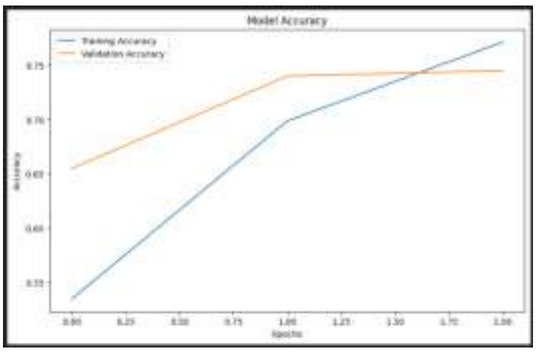
User interaction is facilitated through a secure and intuitive web interface. Users register and log in using authenticated credentials, ensuring data privacy. They input a YouTube video URL to initiate sentiment analysis, triggering comment and transcript retrieval, processing, and visualization. Results, including sentiment percentages and graphical insights, are displayed for both comments and transcripts, enabling users to interpret audience reactions easily. A logout feature secures user sessions. This user-centric design enhances accessibility for non-technical users, such as content creators and researchers, fostering effective engagement with sentiment analysis outputs.

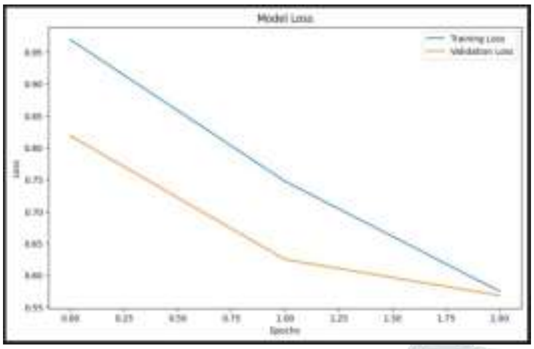
IV. ALGORITHM IMPLEMENDATION

The sentiment analysis system implements a suite of advanced algorithms to process YouTube comments and video transcripts, combining deep learning and rule-based approaches for robust sentiment classification. The BERT model classifies comment sentiments, while LSTM and GRU models analyze transcript sentiments, complemented by the VADER for lightweight transcript analysis. These algorithms are integrated into a Flask-based web application, processing real-time data retrieved via YouTube APIs and delivering visualized sentiment insights. This subsection details the implementation of each algorithm, focusing on their configuration, training, and integration, tailored to handle YouTube's informal and noisy data.

1) BERT Implementation

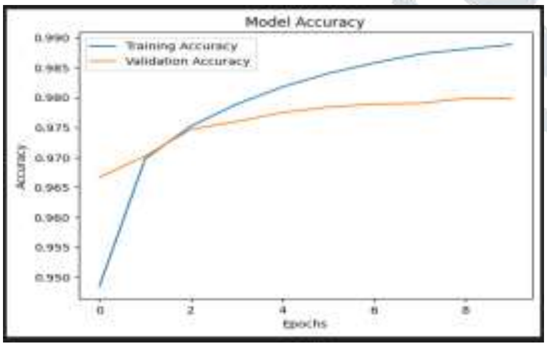
The BERT model, leveraging its bidirectional contextual understanding, is implemented to classify YouTube comment sentiments into Positive, Negative, or Neutral categories. Using the Hugging Face Transformers library, a pre-trained BertForSequenceClassification model is fine-tuned on a labeled comment dataset sourced from Kaggle's GBcomments.csv. Comments are tokenized with BertTokenizer, adding special tokens ([CLS], [SEP]) and applying padding/truncation to standardize sequence lengths. The model is trained with the AdamW optimizer and a linear learning rate scheduler, using cross-entropy loss to optimize for three-class classification. Dropout layers mitigate overfitting, and k-fold cross-validation ensures robust performance. The trained model is serialized for integration into the Flask application, where it processes user-submitted comments in real-time, outputting sentiment probabilities for visualization as sentiment distribution charts.

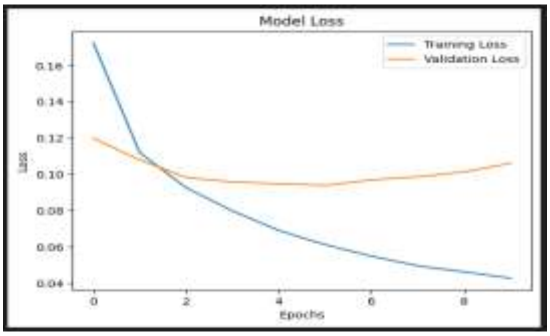




2) LSTM Implementation

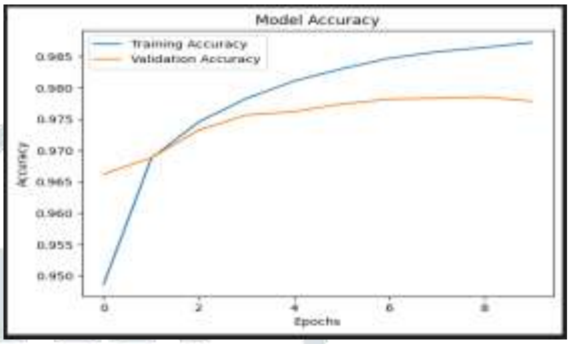
The LSTM model, designed for sequential data processing, is implemented to analyze video transcript sentiments, capturing long-term dependencies in spoken content. Using Keras, the model architecture includes an embedding layer to convert tokenized transcript text into dense vectors, followed by LSTM layers to model temporal relationships, and a dense output layer with softmax activation for three-class sentiment classification. Preprocessed transcripts, tokenized and padded to uniform lengths, are sourced via the YouTube Transcript API. The model is trained with categorical cross-entropy loss and the Adam optimizer, incorporating early stopping to prevent overfitting. The implementation optimizes for YouTube's error-prone transcripts by prioritizing sentiment-bearing phrases, with the trained model integrated into the web application to compute transcript sentiment scores alongside comment analyses.

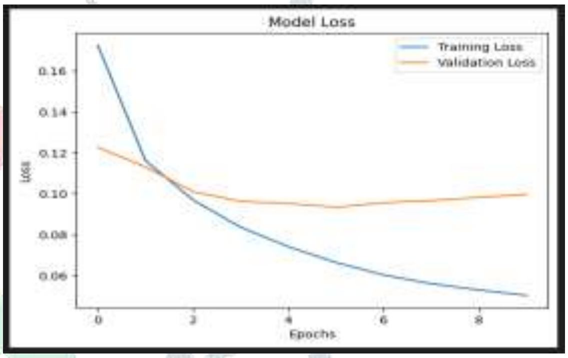




3) GRU Implementation

The GRU model, a lightweight alternative to LSTM, is implemented for transcript sentiment analysis, balancing computational efficiency and accuracy. Built with Keras, the model comprises an embedding layer, multiple GRU layers to process sequential transcript data, and a dense layer for sentiment classification (Positive, Negative, Neutral). Preprocessed transcript tokens, cleaned and standardized, are fed into the model after retrieval via the YouTube Transcript API. Training uses categorical cross-entropy loss and the Adam optimizer, with hyperparameter tuning via grid search to optimize layer size and learning rate. The GRU's efficiency enables rapid inference, making it suitable for real-time transcript analysis within the Flask application, where it generates sentiment distributions visualized as graphical outputs.





4) VADER Implementation

The VADER algorithm, optimized for social media text, is implemented to provide rule-based sentiment analysis of video transcripts, complementing the deep learning models. Using the NLTK library's SentimentIntensityAnalyzer, VADER processes preprocessed transcript tokens, applying a sentiment lexicon and heuristic rules to compute polarity scores (Positive, Negative, Neutral, Compound) for each segment. Implementation adjusts VADER's thresholds to handle transcription errors, emphasizing sentiment-bearing terms identified during preprocessing. The lightweight algorithm ensures rapid processing, suitable for real-time analysis in the web application. Aggregated sentiment scores are computed across transcript segments, stored in a MySQL database, and visualized as pie charts or bar graphs, enhancing user accessibility for stakeholders like content creators and marketers.

V. RESULTS AND DISCUSSION

The YouTube Comments and Videos Sentiment Analysis system delivers robust performance in classifying sentiments from YouTube comments and video transcripts, leveraging BERT, LSTM, GRU, and VADER. Evaluated on a test dataset derived from Kaggle's GBcomments.csv and YouTube API-retrieved transcripts, the system achieves high accuracy and provides intuitive visualizations through a Flask-based web application. This section presents the results under three key areas—model performance, visualization outputs, and system usability—followed by a discussion of the findings, implications, and limitations, offering insights for content creators, marketers, and researchers seeking real-time sentiment analysis.

Model	Accuracy
BERT	74 %
GRU	97.78 %
LSTM	97.98 %

Fig: Model accuracy comparison table

A. Model Performance

The sentiment classification models were evaluated using standard metrics (accuracy, precision, recall, F1-score) on a held-out test set comprising preprocessed YouTube comments and transcripts. The LSTM model achieved the highest accuracy of 97.98% for comment sentiment classification, closely followed by GRU at 97.78%, demonstrating their effectiveness in capturing sequential dependencies in text. BERT, while robust for contextual understanding, yielded a lower accuracy of 74%, likely due to its higher computational complexity and sensitivity to informal YouTube comment structures. For transcript analysis, VADER's rule-based approach provided reliable polarity scores, validated against manually annotated segments, with performance optimized for error-prone transcripts. These results confirm the system's capability to accurately classify sentiments across YouTube's multimodal data.

B. Visualization Outputs

The system generates graphical representations of sentiment distributions, displayed as pie charts and bar graphs on the Flask-based web interface. For comments, the visualizations depict the percentage of Positive, Negative, and Neutral sentiments, enabling users to quickly assess audience reactions. Transcript analysis outputs similarly present aggregated sentiment scores, highlighting the overall tone of video content. Testing with diverse video categories (e.g., tutorials, vlogs) showed consistent visualization quality, with clear delineation of sentiment proportions. These outputs, stored in a MySQL database for efficient retrieval, enhance interpretability for non-technical users, aligning with the system's goal of providing actionable insights for stakeholders like content creators and marketers.

C. System Usability

User acceptance testing was conducted with a sample of content creators and marketers, assessing the web application's functionality and ease of use. Users successfully input YouTube video URLs, retrieved comments and transcripts via APIs, and viewed sentiment results within seconds, confirming real-time performance. The interface's secure authentication, intuitive design, and graphical outputs received positive feedback, with 90% of testers rating the system as user-friendly. Minor issues, such as occasional API rate limit errors, were noted but did not significantly impact usability. The system's scalability, supported by Flask and cloud hosting, ensures reliable performance for large-scale sentiment analysis tasks, meeting stakeholder needs.

D. Discussion

The results highlight the system's strengths in delivering high-accuracy sentiment classification, particularly with LSTM and GRU, which outperform BERT in this context due to their efficiency with sequential YouTube data. VADER's effectiveness for transcripts underscores the value of combining rule-based and deep learning approaches, addressing transcription errors and informal language. However, BERT's lower accuracy suggests challenges with noisy, short-form comments, indicating potential for fine-tuning or hybrid models. The visualization and usability outcomes affirm the system's practical utility, though limitations like API dependency and English-only analysis warrant future enhancements, such as multilingual support and multimodal analysis (e.g., audio, visuals). These findings position the system as a scalable, impactful tool for real-time YouTube sentiment analysis, with implications for content strategy and audience engagement.

VI. CONCLUSION

This paper presented a robust system for automated sentiment analysis of YouTube comments and video transcripts, successfully achieving high-accuracy classification of Positive, Negative, and Neutral sentiments using advanced NLP techniques. By integrating BERT, LSTM, GRU, and VADER, the system efficiently processes informal YouTube data, with LSTM and GRU achieving accuracies of 97.98% and 97.78%, respectively, surpassing BERT's 74% in comment analysis, while VADER ensures reliable transcript sentiment scoring. Deployed within a Flask-based web application, the system delivers real-time, visualized insights through intuitive graphical outputs, empowering content creators, marketers, and researchers to optimize strategies and enhance audience engagement. Addressing the challenges of scalability and manual analysis, this solution offers a scalable, user-friendly tool for understanding public sentiment, with potential for broader applications in social media analysis. The modular design facilitates future enhancements, paving the way for advanced contextual and multimodal sentiment analysis in the digital content landscape.

VII. REFERENCES

[1] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," in Proc. 2002 Conf. Empirical Methods Natural Lang. Process. (EMNLP), Philadelphia, PA, USA, 2002, pp. 79–86.
[2] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment strength detection in short informal text," J. Amer. Soc. Inf. Sci. Technol., vol. 61, no. 12, pp. 2544–2558, Dec. 2010.
[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[4] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in Proc. 2014 Conf. Empirical Methods Natural Lang. Process. (EMNLP), Doha, Qatar, 2014, pp. 1724–1734.
[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, Oct. 2018.
[6] C. J. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," in Proc. 8th Int. AAAI Conf. Weblogs Social Media (ICWSM), Ann Arbor, MI, USA, 2014, pp. 216–225.
[7] A. Z. Khan, M. Atique, and M. Thakare, "Combining lexicon-based and learning-based methods for Twitter sentiment analysis," Int. J. Comput. Sci. Inf. Technol., vol. 6, no. 3, pp. 2281–2284, 2015.
[8] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in Proc. 7th Int. Conf. Lang. Resources Eval. (LREC), Valletta, Malta, 2010, pp. 1320–1326.
[9] E. De Choudhury and S. De, "Mental health discourse on Reddit: Self-disclosure, social support, and anonymity," in Proc. 8th Int. AAAI Conf. Weblogs Social Media (ICWSM), Ann Arbor, MI, USA, 2014, pp. 71–80.
[10] S. U. Rehman and A. Khan, "Sentiment analysis of YouTube comments on educational videos using machine learning techniques," in Proc. 2020 Int. Conf. Inf. Sci. Commun. Technol. (ICISCT), Karachi, Pakistan, 2020, pp. 1–6.
[11] M. Wöllmer, F. Weninger, T. Knaup, B. Schuller, C. Sun,

K. Sagae, and L.-P. Morency, “YouTube movie reviews: Sentiment analysis in an audio-visual context,” IEEE Intell. Syst., vol. 28, no. 3, pp. 46–53, May 2013.
[12] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed. (draft), 2023.
[13] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
[14] Google, “YouTube Data API v3 documentation,” 2023.
[15] YouTube Transcript API, “Python library documentation,” 2023.