

ENHANCED SENTIMENT DETECTION WITH XLNET & NATURAL LANGUAGE PROCESSING

Bomideni Nutan Teja¹, Mrs. P. Swarajya Lakshmi², Dr. G. Vishnu Murthy³

¹*Student, Department of CSE, Anurag University, Hyderabad, India Email: nuthann59@gmail.com*

²*Assistant Professor, Department of CSE, Anurag University, Hyderabad, India*

³*Professor and Dean, Department of CSE, Anurag University, Hyderabad, India*

Abstract—With the rapid expansion of digital communication platforms, an overwhelming amount of opinion-based content—ranging from consumer feedback to social media expressions—requires efficient interpretation for decision-making processes. This study introduces a sentiment classification framework that leverages XLNet, an advanced transformer-based model, optimized for multi-class sentiment prediction tasks. XLNet distinguishes itself from prior architectures by utilizing a permutation-driven training paradigm, which enhances its ability to model intricate linguistic patterns, including sarcasm and domain-specific language.

The offered system combines Natural Language Processing (NLP) workflow that includes data cleaning, tokenization, model fine-tuning, and real-time deployment processes which can be reached through a web interface based on Flask framework. Empirical analysis on wide-range datasets demonstrates a high level of accuracy of the model, making a score of about 86.5 percent on validation data. It is particularly noteworthy that the sarcasm handling and multilingual processing make the system effective in various contexts and in many languages. This framework is scalable and flexible and, as such, can be applied to consumer sentiment tracking, social media assessment, and corporate intelligence collection.

Index Terms—Natural Language Processing (NLP), XLNet, Transformer Architecture, Text Classification, Transfer Learning, Fine-Tuning, Sarcasm Detection, Multilingual NLP, Deep Learning, Sentiment Analysis.

I. INTRODUCTION

A. Overview of XLNet for Sentiment Analysis

The development of Natural Language Processing (NLP) has been largely affected by the appearance of state-of-the-art machine learning architectures, especially transformer-based models. These models have introduced significant changes to work related to sentiment, text summarization, and language translation. One of these, XLNet, has become one of the most promising models by improving the performance of the earlier systems such as BERT. The combination of autoregressive learning and bidirectional modeling of context with regard to the permutation-based approach enables XLNet to learn intricate semantic regularities in the text. This renders it extremely applicable to sensitive sentiment classification assignments.

B. Importance of Sentiment Analysis

The computational evaluation of the emotional tone in a written language is often called opinion mining or sentiment analysis. It usually labels input as already pre-determined sets

of emotion as positive, negative or neutral. Applications The application of sentiment analysis cuts across many spaces such as marketing, finance, customer engagement and public health. In the processing of reviews, surveys and online discussions, companies and organizations can retrieve quality information on what people are thinking and the trends in the society. It is common to find that conventional systems with manual rules and lexicons fail to decipher sarcasm, ambiguity, or otherwise context heavy expressions. Newer models such as XLNet address such limitations with increased use of contextual knowledge.

C. Role of Transformer Architectures in NLP

The transformer models have changed the NLP field by solving critical problems in previous recurrent models such as RNNs and LSTMs, which are incapable of parallelization and longer-range connections. Proposed by Vaswani et al. the Transformer framework also uses the self-attention mechanism that gives the model the ability to perform operations on the entire input sequence at once. This invention made great progress in terms of computational cost-effectiveness and quality of prediction. One of its first successors, BERT, used the second advantage in being bidirectional; however, it was limited by the masked token prediction method. This weakness made it less effective when using sequence generation tasks, which are the tasks that draw insights into word ordering and autoregressive dependencies.



Fig. 1. NLP

D. XLNet: A Next-Generation Pre-Trained Model

XLNet is trained by academic developers at Google and enhances its predecessors with a new training strategy. Machine translation systems In contrast to masked language models, XLNet does not obscure tokens at training time. Rather, it is trained on various kinds of permutations of transactions, and hence, a wider understanding of the variations of context is achieved. Such strategy enhances its power to capture long-term dependencies and syntactic subtleties, which happen to be very crucial in deciding sentiment. Self-attention mechanisms are applied to make sure that long-range connections between words used are not lost; this is encapsulated in the inter-pretability of the model and performance. XLNet has already proven successful on top benchmark performance on dataset like GLUE and SST-2.

E. Application of XLNet in Real-World Sentiment Analysis

DeThe use of XLNet in the sentiment prediction is accom- plished in a number of steps, and it begins with loading and also tokenizing the model, then its fine-tuning by means of labeled sentiment course data. Such data frequently contains social media data, reviews text, and comments of customers. Through training, the model adapts to the linguistic style and sentiment patterns of the domain. Once optimized, XLNet can be used for real-time sentiment inference, supporting applica- tions like brand sentiment tracking, user review analysis, and social listening. Tools such as Hugging Face’s Transformers library simplify integration into production systems, allowing developers to deploy sentiment models at scale with minimal overhead.

II. LITERATURE REVIEW

A. Overview of Existing Sentiment Analysis Techniques

Sentiment analysis has evolved from simple rule-based systems to complex deep learning frameworks, driven by the increasing demand for understanding user-generated content. Numerous researchers have contributed to this field through comprehensive studies and comparative analyses of different methods. This section summarizes key contributions that have significantly shaped the landscape of sentiment analysis.

B. Ravi and Ravi: A Survey on Sentiment Analysis Techniques

In their comprehensive review article, Ravi and Ravi look through a variety of sentiment analysis procedures which get use to reviewing products and social feeds. This paper examines a broad range of machine learning algorithms such as the classical ML models, which are Naive Bayes, Support Vector Machines (SVM), and Decision Tree as well as the neural network-based models, like Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN). The au- thors present the problems of sentiment detection includ- ing interpretation of sarcasm, handling of nebulous phrases and handling domain-specific vocabulary. They point to the necessity of hybrid modeling strategies and domain-specific methods of increased accuracy and robustness.

C. Liu: Insights into Opinion Mining

Bing Liu offers a foundational survey that delves into key aspects of sentiment analysis and opinion mining. The study outlines critical tasks such as sentiment classification, aspect- based sentiment analysis, and opinion retrieval. It categorizes existing methodologies into lexicon-based, statistical, and hy- brid approaches, discussing their strengths and limitations. Liu draws attention to practical issues like domain variability, in- formal expressions (e.g., emoticons and slang), and subjective versus objective content. The paper stresses the importance of context awareness and the development of high-quality, annotated datasets for real-world applications.

D. Zhang, Zhao, and LeCun: DL Advancements

This paper presents the change-bringing aspect of deep learning in opinion mining. Zhang and co-authors draw par- allels between classical sentiment analysis models and new architectures Long Short-Term Memory (LSTM), Gated Re- current Units (GRUs), and Transformers networks. The article observes that deep learning models perform much better than the previous models in capturing non-linear properties and longstanding dependencies in text data. Additionally, it covers the input of pre-trained models such as BU and Gano which add the contextuality and generalization to the sentiment pre- diction models. Nonetheless, the review indicates limitations on computation cost and availability of dense, labeled data.

E. Poria et al.: Sentiment Analysis in Social Media Contexts

In their study, Poria and colleagues concentrate on sentiment analysis applied to dynamic platforms like Twitter and Face- book. Given the brevity and informal nature of social media text, the authors examine tailored strategies such as sentiment analysis using hashtags, emojis, and temporal context. They identify unique challenges like noise in user-generated content, mixed emotional expressions, and implicit sentiment. The study emphasizes the value of sentiment analysis in areas like digital marketing, political opinion mining, and online customer support systems.

F. Cambria and Sentiment: Machine Learning Paradigms

The study of machine learning methods in sentiment clas- sification by the writer and co-author has been narrowed down to depth. Their review involves classifiers that include Decision Trees, k-Nearest Neighbors (KNN), and Ensemble. They discuss feature engineering approach such as bag-of- words, n-grams and word embeddings and why they are useful in enhancing the accuracy of text classification. The authors underline issues including ambiguity in language, data scarcity, and domain sensitivity. They advocate for hybrid frameworks that integrate statistical learning with semantic knowledge bases to effectively interpret complex sentiment structures.

III. EXISTING AND PROPOSED SYSTEM

A. Existing System

Traditionally sentiment analysis was based mostly on the lexicon-based methods, its basic principle involved the use of predefined dictionaries to assign the sentiment scores to specific words or phrases. While effective in structured environments, these techniques often failed in interpreting informal expressions, sarcasm, and context-dependent sentiment shifts frequently found in user-generated content.

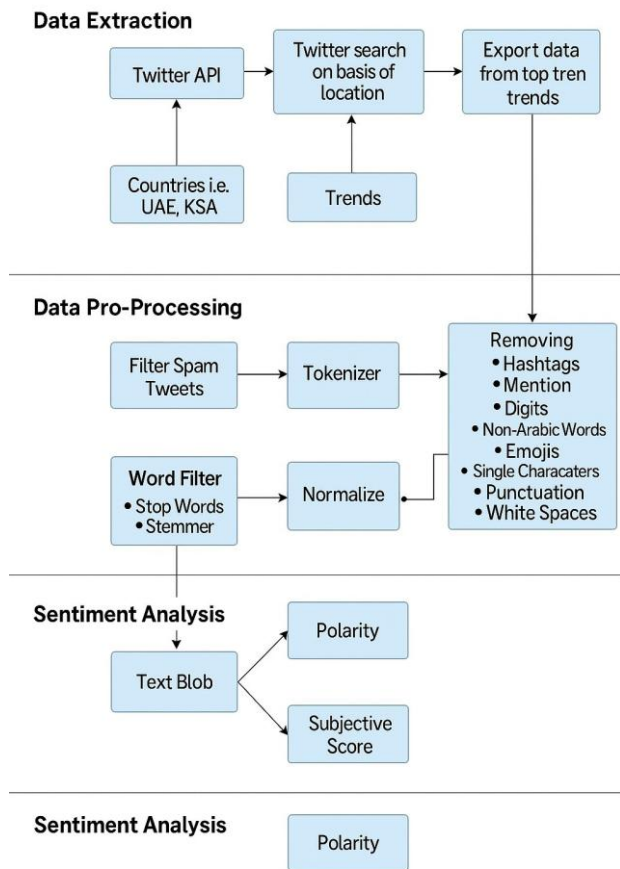


Fig. 2. Block Diagram of Existing Sentiment Analysis System

The emergence of machine learning approaches such as Support Vector Machines (SVM) and Naive Bayes introduced more adaptive sentiment classifiers. These algorithms utilized labeled training datasets to identify sentiment patterns and were more flexible than rule-based systems. However, their ability to understand complex linguistic structures remained limited, particularly in cases involving ambiguity or nuanced emotional tone.

The learning techniques were quite an advancement in the discipline. Architectures like Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks allowed vertical and time series feature extraction of text. Such models have enhanced sentiment classification by ensuring that more complex representations of a text are learned. The

presentation of transformer models such as BERT (Bidirectional Encoder Representations from Transformers) introduced the contextual word vectors and transfer learning possibilities. Despite its success, BERT's reliance on masked language modeling limits its effectiveness in capturing sequential dependencies needed for tasks involving language generation or complete contextual modeling.

B. Proposed System

The proposed sentiment analysis model is built upon XLNet, a state-of-the-art transformer-based architecture developed to address the shortcomings of its predecessors. XLNet introduces a permutation-based autoregressive pretraining strategy, which enhances its capacity to model both left-to-right and right-to-left dependencies in text without using masked tokens. This results in better contextual understanding and more flexible sentence representations.

PROPOSED SYSTEM

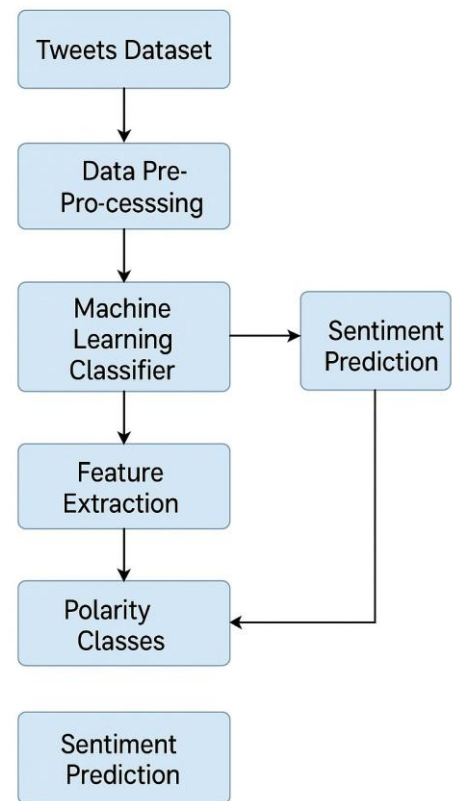


Fig. 3. Block Diagram of Proposed Sentiment Analysis System Using XLNet

The system pipeline begins with standard NLP preprocessing techniques such as noise removal, tokenization, and text vectorization. Once the text is processed, it is passed through a fine-tuned XLNet model trained on domain-specific sentiment datasets. This makes it possible to classify text based on sentiment in the following ways, positive, negative, neutral,

and even mixed, and, at the same time, allows extracting even the slightest details of sentiment as sarcasm and irony.

Architected to scale, it has the capacity to handle large numbers of electronic events in real-time using product reviews, social media feeds, and survey response sources to detect sentiment. In addition, the system is designed to accommodate various languages, and as such, it can be utilized in different languages and cultures. The framework is quite useful in areas like user experience analytics, customer feedback, trend, and analysis in public discourse which have the following features.

IV. SYSTEM ARCHITECTURE

The proposed architectural design of the sentiment analysis system is optimized to operate on real-time textual information using sophisticated deep learning techniques. It is based on the fine-tuned version of the XLNet model that operates in relation to the multi-stage Natural Language Processing (NLP) system and the interactive front-end that has been developed using Flask. This system configuration provides a seamless and efficient pathway through the processing of text, classification of sentiment and presentation of the results.

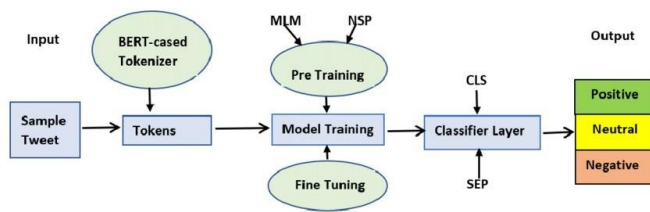


Fig. 4. System Architecture of the Proposed XLNet-Based Sentiment Analysis Model

A. System Components

1) *User Input Interface*: The system starts with a front-end interface that enables users to submit text data either manually or via automated methods. Text may be sourced from platforms like YouTube comments, product reviews, or uploaded datasets in CSV format. This raw data is then forwarded to the preprocessing engine.

2) *Text Preprocessing Module*: During this step, the text entering is cleaned in accordance with the traditional NLP methods. That consists of elimination of elements of noise, including punctuation, emojis, special characters, excess of spaces and stop words. A lowercase conversion is performed on the text to standardize it and then a tokenization is applied to the text to ease the subsequent transformation of its input.

3) *Tokenization and Embedding*: Before tokenization, the text is converted to the XLNet tokenizer (by use of Hugging Face Transformers library). The tokenizer breaks up the data into tokens and converts them into numerical embeddings with both syntactic and semantic context. These embeddings represent the representation of the model as it provides the input representation of a further sentiment inference.

4) *Sentiment Classification via Fine-Tuned XLNet*: The fine-tuned XLNet model is the main part of the architecture. XLNet is trained on sentence-labeled training data, and is able to examine the relationships between the context of the given tokenized input through its permutation-based language modeling strategy. The model outputs predictions of sentiment (Positive, Negative, or Neutral), confidence of the prediction and probability assigned to the assigned class.

5) *Prediction Output and Interpretation*: The model output is mapped onto readable sentiments of humans. Dependent measures like confidence values and probability distribution across sentiment classes are additional measures that are calculated in the system. Such structured outputs may be kept in the databases or presented to the front-end to be displayed in real-time.

6) *Result Visualization via Web Application*: A Flask-based web application is used to present the results interactively. Visualization tools such as bar graphs, pie charts, and word clouds provide a clear and insightful view of sentiment trends. Users can analyze results across multiple data entries, enhancing understanding and supporting decision-making.

7) *Optional Enhancements*: The modularity of the system enables one to easily add some more features. These can range anywhere up to multilingual sentiment support, sarcasm and irony identification, or API-driven dynamic data collection of social sites e.g. Twitter, YouTube. Extensions in this form broker a wide range of functionalities in the application of the system as well as in the areas to which it applies.

V. IMPLEMENTATION

The presented section defines fundamental components of the chosen sentiment analysis framework that will solve the problem of providing accurate and flexible sentiment categorization in diverse linguistic environments.

A. Data Preprocessing Module

The initial stage of the system is data preprocessing. This module converts unstructured text, which is in its raw form, into a clean and standardized form which can be fed into neural network based models. The procedure involves tokenization whereby sentences are slashed into tokens or words, as well as stop-word elimination, which eradicates the most common, vocabulary level words that do not carry any contextual meaning like: the, and, is. In addition, text normalizations are carried out, e.g. lowercasing, removal of punctuation, and simple spelling corrections. All these steps decrease data noise, increase consistency, and lead to better learning of the model during training processes.

B. Sentiment Classification Module

This module is the analytical part of the system. Based on the state-of-the-art XLNet transformer-based model, the system will recognize inputs in text format and will use the classification mechanism into one of four certain sentiments: *Positive*, *Negative*, *Neutral*, or *Mixed*. Unlike the previous transformer models, XLNet has a permutation based training

A. System Components

- 1) **User Input Interface:** A straightforward input interface is provided where users can manually enter text or load data from external platforms such as YouTube comments, product reviews, or CSV files. This raw data proceeds to the preprocessing unit.
- 2) **Text Preprocessing Module:** In this phase, textual data is refined using NLP operations. These include the removal of stopwords, emojis, punctuations, special symbols, and whitespace redundancies. Text is standardized to lowercase and tokenized to prepare it for the model pipeline.
- 3) **Tokenization and Embedding:** The clean text is fed into LSTM which is XLNet’s tokenizer and that breaks the input into meaningful tokens and this produces embeddings. These embeddings capture linguistic patterns and context which is necessary for doing reliable sentiment analysis.
- 4) **Sentiment Classification Using XLNet:** A pre-trained and fine-tuned XLNet model is used as the main part of the system and optimized on sentiment-labeled datasets. The permutation model sequence of XLNet enables it to learn text with a higher understanding of text. The model indicates sentiment predictions (Positive, Neutral or Negative) with the help of confidence levels and the probability of the classes.
- 5) **Prediction Results and Interpretation:** Output sentiments are mapped to user-friendly labels. Additional performance indicators like confidence scores and probability charts are generated. These outputs are either stored for analysis or sent directly to the web interface.
- 6) **Result Visualization and Web Display:** Sentiment results are displayed on a Flask-based web portal using visual tools such as pie charts, bar graphs, and word clouds. These visuals enhance the interpretability of multiple results in real time.
- 7) **Optional Enhancements:** The architecture supports scalability with optional features including multilingual sentiment analysis, sarcasm recognition, and API integration with platforms like Twitter or YouTube, making the system adaptable to varied use cases.

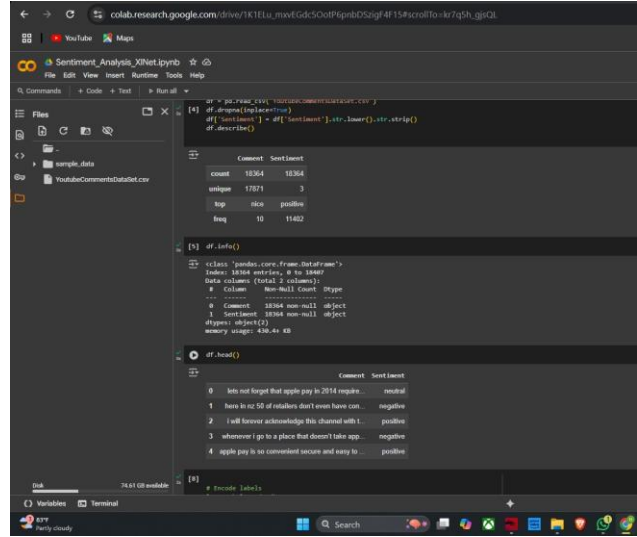


Fig. 8. System Architecture

B. Integration Testing

End-to-end functionality was validated using a small batch of diverse inputs. The complete pipeline—from input acquisition to visualization—functioned seamlessly.

Test Summary:

- Positive Sentiments: 2 inputs
- Neutral Sentiments: 3 inputs

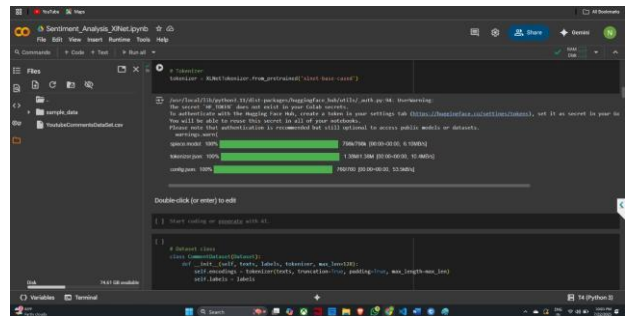


Fig. 9. Grouped Bar Chart – Sentiment Predictions

VII. OUTPUT SCREENS

To validate the effectiveness of the sentiment analysis system, multiple testing approaches were conducted—spanning functional, integration, and visual validations.

A. Unit Testing

Each module—preprocessing, model loading, and classification—was tested independently. For instance:

Input: “apple pay is convenient and fast”
Prediction: Positive (Confidence: 99.93%)

All individual tests produced expected results without errors.

C. Visual Evaluation and Metrics

Performance evaluation was supported with standard training metrics:

- **Loss Curve:** Demonstrated a smooth and stable decline, confirming convergence.
- **Accuracy Curve:** Achieved a peak validation accuracy of approximately 86.5%.
- **Confusion Matrix:** Indicated effective classification, with minor misclassifications between neutral and positive categories.

D. Web-Based Testing

The Flask interface supported real-time user interaction and prediction display.

E. Google Colab Validation

The model development and evaluation were performed in Google Colab, ensuring reproducibility and transparency in workflow.

- Notebook initialized and configured.
- Dataset loaded, explored, and tokenized.
- Training and evaluation executed seamlessly.

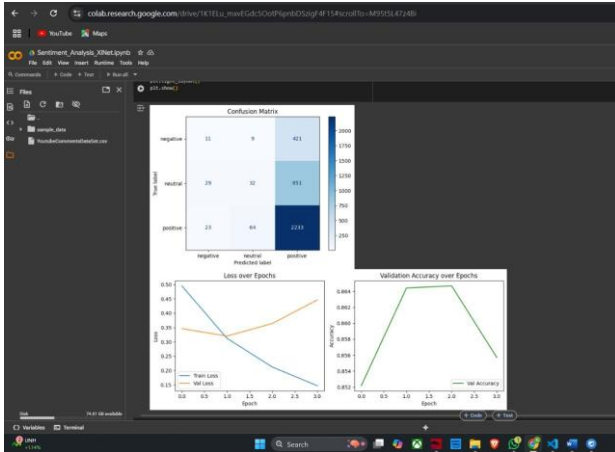


Fig. 10. Evaluation Metrics Composite Plot

Example Output:

- Total Comments Analyzed: 78
- Positive: 37
- Neutral: 2
- Negative: 39

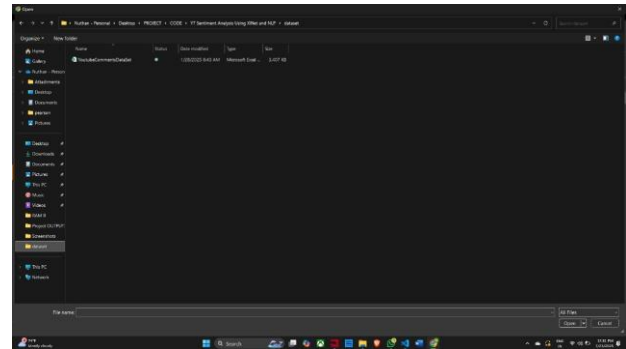


Fig. 13. Tokenizer Initialization

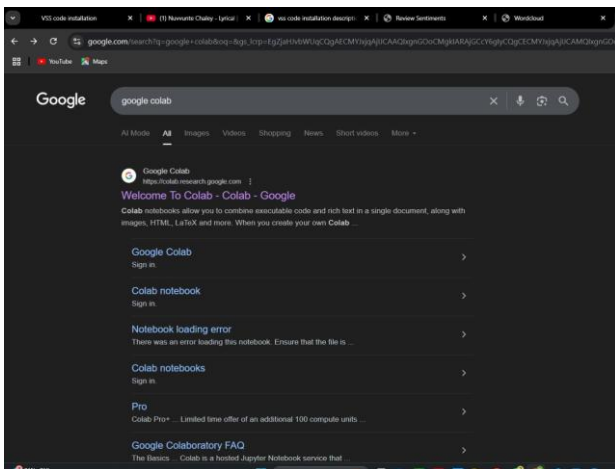


Fig. 11. Sentiment Summary of 78 Comments

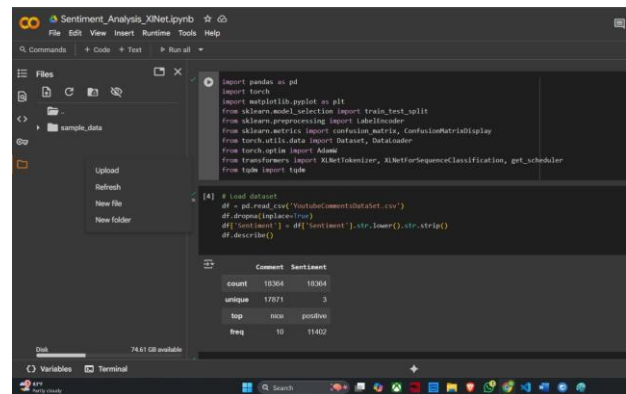


Fig. 14. Training Progress

VIII. CONCLUSION

This project presents a robust sentiment analysis system utilizing the XLNet model. By capitalizing on XLNet’s advanced contextual learning capabilities, the system effectively navigates complex language structures, including sarcasm and domain-specific terms. A modular design approach was adopted to streamline the end-to-end flow—from data collection and cleaning to sentiment prediction and result visualization.

Achieving a validation accuracy of around 86.5%, the model demonstrated strong generalization. Both functional and visual evaluations confirmed the reliability of predictions. Deployment via a Flask interface and development within Google Colab ensured accessibility and reproducibility. The system stands as a practical tool for sentiment interpretation, laying the groundwork for enhancements in multilingual support, emotion recognition, and social media analytics.

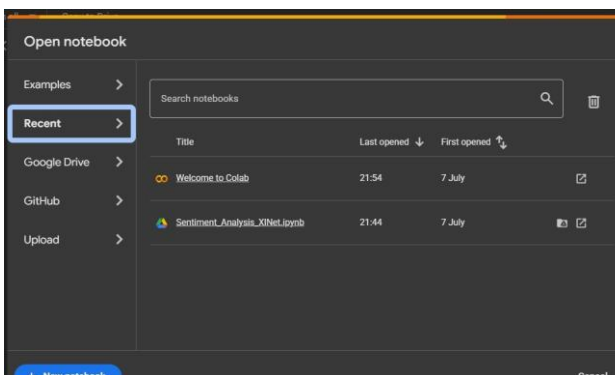


Fig. 12. Word Cloud from Processed Comments

sible, fostering inclusivity in regions with underrepresented linguistic datasets.

IX. FUTURE WORK AND ENHANCEMENTS

Looking ahead, there are several promising directions to improve and extend the capabilities of the current sentiment analysis framework. One key area is the incorporation of **emotion recognition**, which would enable the system to identify not only sentiments like positive, negative, or neutral, but also more nuanced emotional states such as joy, anger, fear, or surprise. This enhancement would significantly improve the interpretability of user feedback, particularly in sectors like mental health monitoring or customer relationship management.

Another crucial upgrade involves integrating **Explainable Artificial Intelligence (XAI)** techniques to provide transparency in the model's decision-making process. This will be especially vital for high-stakes domains such as finance, law, or healthcare, where understanding the reasoning behind model predictions is essential.

To maintain the system's relevance over time, implementing a **continuous learning** pipeline will allow the model to adapt to evolving language use, slang, and cultural expressions. This approach ensures sustained accuracy in dynamic linguistic environments.

Expanding the system's capabilities into the field of **multimodal sentiment analysis**—by combining textual data with audio, facial expression, or gesture inputs—can significantly enhance its applicability in real-time applications such as video conferencing tools, live streaming feedback monitoring, and virtual assistants.

Moreover, migrating the architecture to a **cloud-based environment** would ensure scalability, real-time performance optimization, and deployment flexibility. Support for **low-resource languages** would make the system globally acces-

REFERENCES

- [1] 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*, 2019.
- [2] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," *arXiv preprint arXiv:1906.08237*, 2019.
- [3] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.
- [4] C. N. dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *COLING 2014*, pp. 69–78.
- [5] X. Zhao and S. Li, "A review of sentiment analysis research in social media," *Int. J. Comput. Sci. Issues*, vol. 17, no. 3, pp. 1–8, 2020.
- [6] J. Pustejovsky and A. Stubbs, *Natural Language Annotation for Machine Learning*. O'Reilly Media, 2013.
- [7] L. Lee and J. Kim, "Sentiment analysis and opinion mining: A survey," *Int. J. Comput. Appl.*, vol. 975, pp. 7–16, 2019.
- [8] S. Ghosh and T. Chakraborty, "Sentiment analysis of social media data: A review," *J. King Saud Univ.—Comput. Inf. Sci.*, 2020.
- [9] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, no. 1–2, pp. 1–135, 2008.
- [10] S. Kiritchenko and S. M. Mohammad, "Best ways to read and interpret sentiment lexicons," in *Proc. Int. Conf. Comput. Intell., Model. Simulation*, 2017.
- [11] S. Xu and L. Chen, "A novel deep learning model for sentiment analysis based on XLNet and CNN," *J. Electr. Eng. Technol.*, vol. 16, no. 4, pp. 1783–1792, 2021.
- [12] S. Bansal and S. Soni, "Sentiment analysis using deep learning techniques: A survey," *Procedia Comput. Sci.*, vol. 132, pp. 1052–1058, 2018.
- [13] C. J. Hutto and E. E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis," in *ICWSM*, 2014, pp. 216–225.
- [14] J. P. McCrae and M. de Leon, "The importance of sarcasm in sentiment analysis," *J. Artif. Intell. Res.*, vol. 55, pp. 1–20, 2021.
- [15] R. Raj and P. Singh, "Application of deep learning in sentiment analysis," *J. Comput. Sci.*, vol. 44, 101095, 2020.

- [16] E. Cambria *et al.*, “New avenues in opinion mining and sentiment analysis,” *IEEE Intell. Syst.*, vol. 28, no. 2, pp. 15–21, 2013.
- [17] B. Liu, *Sentiment Analysis and Opinion Mining*. Morgan & Claypool, 2012.
- [18] S. Ruder, M. E. Peters, S. Swayamdipta, and T. Wolf, “Transfer learning in NLP,” in *NAACL Tutorials*, 2019, pp. 15–18.
- [19] T. Wolf *et al.*, “Transformers: State-of-the-art NLP,” in *EMNLP: System Demonstrations*, 2020, pp. 38–45.
- [20] C. Sun, L. Huang, and X. Qiu, “Utilizing BERT for aspect-based sentiment analysis,” *arXiv preprint arXiv:1903.09588*, 2019.
- [21] W. Medhat, A. Hassan, and H. Korashy, “Sentiment analysis algorithms and applications,” *Ain Shams Eng. J.*, vol. 5, no. 4, pp. 1093–1113, 2014.
- [22] J. Howard and S. Ruder, “Universal language model fine-tuning for text classification,” in *ACL*, 2018, pp. 328–339.
- [23] L. Zhang, S. Wang, and B. Liu, “Deep learning for sentiment analysis,” *WIREs Data Min. Knowl. Discov.*, vol. 8, no. 4, e1253, 2018.
- [24] M. S. Akhtar *et al.*, “A deep ensemble framework for sarcasm detection,” in *COLING*, 2016, pp. 377–387.
- [25] A. Rani and A. Kumar, “Comparative study of transformer-based models for multilingual sentiment analysis,” *Int. J. Inf. Technol.*, vol. 14, no. 2, pp. 423–429, 2022.
- [26] Y. Zhang and B. C. Wallace, “A sensitivity analysis of CNNs for sentence classification,” in *IJCNLP*, 2017, pp. 253–263.
- [27] M. Mozafari, R. Farahbakhsh, and N. Crespi, “A BERT-based approach for hate speech detection,” *Complexity*, vol. 2020, Article ID 8885860.
- [28] D. Patel and M. Parmar, “Enhanced sarcasm detection using hybrid deep learning,” *J. King Saud Univ.–Comput. Inf. Sci.*, 2021.
- [29] J. Jiang and E. Cambria, “EmotionFlow: Emotion recognition from textual conversations,” in *EMNLP*, 2016, pp. 1639–1649.
- [30] J. Barnes, R. Klinger, and S. Schulte im Walde, “Sentiment analysis is not solved!,” in *LREC*, 2020, pp. 628–638.
- [31] W. Gao, W. Li, and Y. Liu, “Improving aspect-level sentiment classification with gated convolution,” in *EMNLP*, 2019, pp. 4239–4248.
- [32] A. Rietzler, C. Stab, and I. Gurevych, “Domain adaptation through BERT for sentiment classification,” in *LREC*, 2020, pp. 4933–4941.
- [33] T. Chakrabarty *et al.*, “AMPERSAND: Argument mining for persuasive discussions,” in *EMNLP*, 2020, pp. 5399–5411.
- [34] J. Zhou and R. Zafarani, “Fake news detection: A survey,” *ACM Comput. Surv.*, vol. 53, no. 5, pp. 1–40, 2018.
- [35] S. Liu, H. Zhan, and Z. Zhao, “A survey of sentiment analysis based on pre-trained models,” *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 20, no. 5, pp. 1–34, 2021.