Multimodal Hate Speech Detection with Fine-tuned BERT and Real-Time Web Application

Sai Krishna Pothini
AI&DS
Lakireddy Balireddy College of Engineering
Mylavaram, AP, India
29saikrishna102002@gmail.com

Ummadi Surya Venkata Sekhar
AI&DS
Lakireddy Balireddy College of Engineering
Mylavaram, AP, India
suryaelonm@gmail.com

Kandagaddala Likhitha Suma Venkat
AI&DS
Lakireddy Balireddy College of Engineering
Mylavaram, AP, India
29saikrishna102002@gmail.com

Sai Mahesh Battula
Information Technology
Vasireddy Venkatadri Institute of Technology
Guntur, AP, India
saibattula1810@gmail.com

***Abstract*— In response to the growing problem of hate speech plaguing online platforms, The study presents a thorough detecting technique that makes use of the most recent BERT language model. The proliferation of hate speech creates a toxic online environment, discourages civil discourse, and can negatively impact user well-being. Current hate speech detection methods often lack accuracy or generalizability, hindering their effectiveness in real-world scenarios. To address these limitations, this project incorporates a multimodal approach, analyzing both text and audio data for a more nuanced understanding of hate speech. A diverse dataset encompassing textual and spoken content from various sources, including social media and audio recordings, is first acquired and preprocessed. The core of the system is a fine-tuned BERT model, trained with appropriate loss functions and evaluation metrics to achieve robust and accurate hate speech detection while minimizing false positives. This fine-tuned model is then integrated into a user-friendly web application built with Flask. This user-centric approach empowers individuals to analyze their content for potential hate speech through simple text input or audio file upload. By providing actionable insights, the project aims to foster a more inclusive online environment and contribute to the mitigation of hate speech dissemination in digital spaces.**

***Keywords*— Hate speech detection, BERT, NLP, Machine Learning, Text classification, Audio classification, Fine-tuning, Web application, Flask, Speech recognition, Toxic content, Digital discourse, Online platforms.**

I. INTRODUCTION

The digital revolution has fundamentally reshaped communication, fostering a global online community. While digital platforms facilitate information exchange and social interaction, Additionally, sites have turned into hubs for the spread of offensive information [1]. Security on the internet and inclusion are compromised by hate speech, which is characterized as statements that disparage individuals or groups on the basis of their race, ethnicity, religion, or other attributes [8]. This project proposes a comprehensive hate speech detection system powered by advanced machine learning techniques to address this growing concern.

At the heart of our system lies BERT [1]. This powerful NLP model excels at capturing subtle linguistic nuances and semantic relationships within text. BERT's ability to understand context is crucial for effectively identifying hate speech, which often relies on sarcasm, implication, and coded language [10]. Research by Vishwamitra et al. (2020) demonstrates the importance of context for hate speech detection, particularly in the case of COVID-19 related hate speech, where nuanced language can mask hateful intent [10].

Existing research underscores the potential of BERT for hate speech detection. Mnassri et al. (2022) explored ensemble approaches using BERT to enhance the model's robustness [2]. Conversely, Kennedy et al. (2020) focused on interpretability to build trust in model decisions [3]. However, limitations persist. Language-specific challenges necessitate tailored models, as demonstrated by Almaliki et al., who developed a BERT-based model specifically for detecting hate speech in Arabic text [4]. Similarly, Alatawi et al. used BERT in combination with domain-particular embeddings of words to combat statements promoting white supremacist ideology [5]. These studies highlight the importance of considering language variations and ideological contexts in hate speech detection [4, 5]. Furthermore, Sohn and Lee (2019) proposed MC-BERT4Hate for cross-lingual detection, highlighting the need for robust evaluation across diverse linguistic communities [6]. Keya et al. (2023) addressed this with G-BERT, a productive technique for locating hate speech in Bengali writing [7]. However, limited resources for under-resourced languages remain a challenge, as noted by Miok et al. (2022) [8].

Multimodal Detection We extend beyond text analysis by integrating speech recognition and audio processing techniques to detect hate speech within audio recordings, similar to Dukic and Sovic Krzic (2021) [9]. This broadens our system's applicability across diverse digital mediums. User-centric Web Application- Democratizing access to this technology, we develop a user-friendly web application using Flask. This application allows individuals to submit text or upload audio recordings for real-time analysis, empowering users for proactive intervention and content moderation. Fine-tuned BERT Model- We leverage transfer learning to adapt a pre-trained BERT model to the specific task of hate speech detection, capitalizing on its vast amount of learned information. We further refine the model through hyperparameter tuning to achieve high accuracy, precision, and recall while minimizing false positives.

By combining these elements, our project aims to create a more robust, user-centric, and generalizable solution for combating the spread of hate speech in online environments. This system has the potential to foster a more secure and welcoming online environment for all users, encouraging polite conversation and lessening the harmful effects of hate speech.

II. LITERATURE REVIEW

By utilizing cutting-edge NLP methods, hate speech detection has achieved tremendous strides in the past few years. BERT has emerged as a powerful tool for understanding the meaning of words in text. Because BERT-based models are capable of capturing text's meaning subtleties and context-specific details, they have become widely used in detection of hate speech task. However, existing research has highlighted several challenges and opportunities in this domain.

Combining BERT, as the discriminatory embedded words with dl models, Hind et al. (2023) established a system for detecting hate language [1]. Their approach aimed to leverage the rich semantic representations learned by BERT along with domain-specific word embeddings to enhance hate speech detection performance. While their model demonstrated promising results, the computational complexity associated with deep learning architectures may limit scalability and real-time deployment.

Mnassri et al. (2022) explored ensemble approaches based on BERT for hate speech detection [2]. By combining Several BERT models that were trained on different subsets of data, they aimed to improve the robustness and generalization of hate speech classifiers. However, ensemble methods may require additional computational resources and careful tuning to achieve optimal performance, posing practical challenges in resource-constrained environments.

Kennedy et al. (2020) addressed the issue of interpretability in hate speech classifiers by contextualizing model predictions with post-hoc explanations [3]. Their methodology attempted to reveal possible biases or constraints and offer insight on how decisions are made modelled. While interpretability is crucial for building trust in AI systems, achieving a balance between model complexity and transparency remains a challenge in hate speech detection.

The Arabic BERT-Mini model (ABMM) was presented by Almaliki et al. (2023) as a tool for identifying slanderous remarks on social networking sites [4]. Their work highlighted the importance of language-specific models in addressing linguistic nuances and cultural contexts in hate speech detection. However, the availability of pre-trained models and resources for low-resource languages may be limited, hindering the development of effective hate speech detection systems for diverse linguistic communities.

The goal of (Alatawi et al., 2021) was to identify statements of hatred pertaining to white supremacists by utilising neural networks and domain-dependent word embedding in combination with BERT [5]. Their study emphasized the need for tailored approaches to address the unique characteristics of hate speech in different ideological contexts. However, identifying and mitigating biases in training data and model predictions remains a critical challenge in hate speech detection.

A hateful language detection system called MC-BERT4Hate was presented by (Sohn & Lee, 2019) [6]. It makes use of multiple channels of communication BERT systems for various languages and translations. Their approach aimed to improve cross-lingual hate speech detection by leveraging multilingual representations learned by BERT. However, language-specific nuances and biases may affect model performance, highlighting the importance of robust evaluation across diverse linguistic communities.

An efficient method for identifying hate speech in Bengali writings on social networking platforms is G-BERT., has been suggested by (Keya et al., 2023) [7]. Their work underscored the importance of language-specific models and resources in addressing hate speech in non-English languages. However, the limited availability of annotated datasets and linguistic resources may pose challenges in developing effective hate speech detection systems for under-resourced languages.

Bayesian attention networks were proposed by (Miok et al., 2022) for reliable hate speech recognition in order to take uncertainty and model predictability into account. [8]. Their work highlighted the importance of probabilistic modeling techniques in handling ambiguity and noise in hate speech detection tasks. However, incorporating uncertainty estimates into existing hate speech detection frameworks may require additional computational overhead and model complexity.

III. METHODOLOGY

1. *Data Collection and Preprocessing:*

The first step in our methodology is to collect and preprocess the data. We utilized a dataset containing restaurant reviews in TSV format. The dataset includes text reviews and corresponding labels indicating whether the review was positive (liked) or negative (not liked). We loaded the dataset using the Pandas library, performed basic exploratory data analysis to understand its structure and distribution, and then preprocessed the text data by adding a new column to represent the sentiment of each review.

2. *Data Splitting:*

To ensure our model can effectively generalize to unseen data, we divided the preprocessed dataset into training and testing sets. This split accomplishes two goals: training the model on a representative subset of the data and evaluating its performance on a separate, held-out portion.

3. *BERT Model Integration:*

We integrated BERT, a state-of-the-art NLP model, into our classification task. To achieve this, we loaded two pre-trained BERT models from TensorFlow Hub: one for text preprocessing (bert_en_uncased_preprocess) and another for text encoding (bert_en_uncased_L-12_H-768_A-12). These models are specifically designed for processing English text data.

4. *Model Architecture Design:*

   Our system leverages BERT for text processing and incorporates a bidirectional LSTM to capture sequence information. A dropout layer prevents overfitting, and a final layer classifies text as hateful or not.

5. *Model Compilation and Training:*

We utilise binary cross-entropy loss as parameters and the optimizer known as Adam for training the algorithm, monitoring metrics like accuracy, precision, and recall. Training occurs on a designated dataset with validation to prevent overfitting.

6. *Model Evaluation and Performance Analysis:*

Lastly, we use measures like the F1-score to assess the model's performance on unknown data and examine the confusion matrix to identify potential biases. This comprehensive approach ensures a robust hate speech detection system.

7. *Deployment and Integration:*

At last, when the model is trained and also evaluated, we save it in the HDF5 format for future deployment and integration into real-world applications. We leverage the Flask framework to create a simple web application that allows users to input text or audio files and receive predictions on whether the content contains hate speech. This deployment facilitates practical usage of the model for real-time hate speech detection in various online platforms.

A. *Dataset Description*

The dataset consists of 556 samples of restaurant reviews, where each sample includes two main components: the review text and the associated sentiment label. The review text represents the textual feedback provided by customers regarding their dining experiences at various restaurants. These reviews cover a wide range of aspects, including the quality of food, service, ambiance, pricing, and overall satisfaction. The sentiment labels indicate whether each review is positive or negative, with a value of 1 denoting a positive sentiment (indicating satisfaction or approval) and a value of 0 denoting a negative sentiment (indicating dissatisfaction or disapproval).

Each review text is a textual description of the customer's experience, reflecting their opinions, perceptions, and emotions related to their dining experience. These reviews may vary in length, ranging from short, concise statements to longer, more detailed descriptions. The content of the reviews encompasses various aspects of the dining experience, such as the taste and quality of the food, the promptness and friendliness of the service, the atmosphere and cleanliness of the restaurant, and any other factors that may have influenced the customer's overall satisfaction.

The sentiment labels provide a binary classification of each review as either positive or negative based on the customer's sentiment expressed in the review text. These labels serve as ground truth annotations that categorize the reviews into two distinct classes, enabling the development and evaluation of sentiment analysis models. Positive sentiment labels (1) indicate that the customer expressed satisfaction or approval in their review, while negative sentiment labels (0) indicate that the customer expressed dissatisfaction or disapproval.

B. *Analysis of Dataset:*

Distribution of Review Lengths:

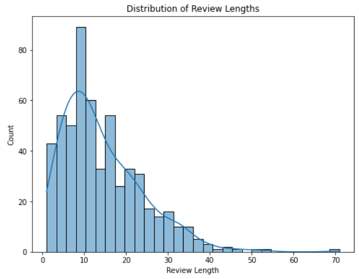


Fig 1: Distribution of Review Lengths

Distribution of 'Happiness' Column:

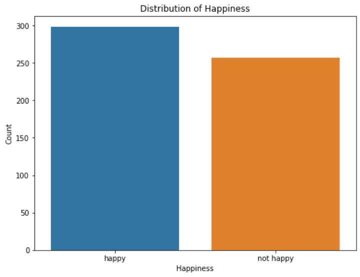


Fig 2: Distribution of 'Happiness' Column

Word Cloud Generation:

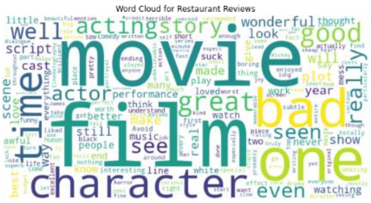


Fig 3: Word Cloud Generation

C. *Algorithm Justifications:*

1. Utilization of BERT Model: Using the BERT algorithm for identifying hate speech is the main

algorithmic decision we made for the purpose of the project. Modern transformer-based architecture BERT has shown exceptionally effective in a range of NLP applications, such as question answering, sentiment analysis, and text categorization. We choose BERT due to its ability to capture contextual information effectively, which is crucial for understanding the nuanced semantics and subtle nuances present in hate speech content.

2. Contextual Embeddings: BERT considers the complete input text sequence in both directions to create contextual embeddings This means that each word in the text is represented based on its surrounding context, allowing the model to grasp the meaning of words in relation to the entire sentence. This feature is crucial for the identification of hate speech since hate speech frequently uses sarcasm and other contextual indicators., and implicit meanings that may not be captured by traditional bag-of-words models.
3. Fine-tuning for Hate Speech Detection: While Enormous scale text is used to train already trained BERT algorithm, also corpora for generic language understanding, we explicitly adjust the BERT model to detect hate speech. Readjusting the model's parameters on a smaller dataset with instances of labelled hate speech is known as fine-tuning. Through this procedure, the model can modify its representations to more accurately depict the features of hate speech, such as offensive language, derogatory terms, and discriminatory content.
4. Multimodal Integration: In addition to text data, our project incorporates audio inputs to identify hateful speech. We utilize the capability of the BERT model to process text inputs by transcribing audio content into textual representations using automatic speech recognition (ASR) systems. By treating audio transcripts as textual data, we can apply the same BERT-based hate speech detection model to both text and audio modalities, thereby enhancing the versatility and effectiveness of the system.
5. Real-time Deployment: The BERT-based hate speech detection model is deployed in a real-time web application, allowing users to input text or audio content and receive instant predictions on the presence of hate speech. This deployment strategy leverages the efficient inference capabilities of the BERT model, enabling quick and responsive detection of hate speech content directly within web platforms.

To sum up, the use of the BERT model for hate speech detection stems from its cutting-edge functionality, contextual understanding capabilities, and suitability for fine-tuning on specific tasks. By integrating BERT into our project and adapting it for hate speech detection across text and audio modalities, Our goal is to create a reliable and adaptable technology that can instantly recognise dangerous internet activity.

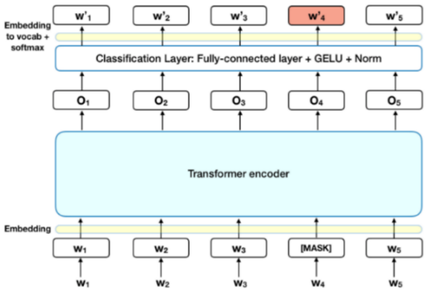

w'1
w'2
w'3
w'4
w'5
Embedding to vocab + softmax
Classification Layer: Fully-connected layer + GELU + Norm
O1
O2
O3
O4
O5
Transformer encoder
Embedding
w1
w2
w3
[MASK]
w5
w1
w2
w3
w4
w5

Fig 4: BERT Model Architecture [11]

IV. ARCHITECHTURURAL DESCRIPTION

The architecture of the implemented sentiment analysis system is designed to provide a seamless and user-friendly experience, from user input to sentiment result display. Let's break down the architecture into its key components:

1. User Interface (Flask Web Application):

The system begins with a user interface built using Flask, a web framework for Python. Flask enables the creation of web applications with endpoints for handling user requests and rendering HTML templates. The Flask application provides endpoints for users to interact with the sentiment analysis functionality. These endpoints include routes for text input (/analyze_input1), audio input (/analyze_input2), and live audio analysis (/analyse_live).

2. Text and Audio Input Handling:

Upon receiving a user request, Flask routes the request to the appropriate endpoint based on the type of input provided (text or audio). For text input, users can input text directly into a text field in the web interface. For audio input, users can upload an audio file. The text input is received as a string, while the audio input is processed using the SpeechRecognition library to convert the audio file to text.

3. Sentiment Analysis Functionality:

The sentiment analysis functionality is implemented in the analyze_text_sentiment and analyze_audio_sentiment functions. For text input, the analyze_text_sentiment function leverages a pre-trained BERT-based model (loaded_model) to predict the sentiment of the input text. This model has been trained to classify text as either offensive or non-offensive. For audio input, the analyze_audio_sentiment function utilizes the SpeechRecognition library to convert the audio file to text. The resulting text is then passed to the analyze_text_sentiment function for sentiment analysis.

4. BERT-Based Model for Sentiment Analysis:

The sentiment analysis model used in the system is based on BERT, An effective network model for situations involving processing natural languages. The model is imported from BERT into TensorFlow Hub, a repository for pre-trained machine learning algorithms. Two components of the BERT model are utilized: the BERT preprocessing module (bert_preprocess) and the BERT encoder module (bert_encoder).

5. Result Formatting and Presentation:

After sentiment analysis, the sentiment result is processed to determine whether it indicates an offensive or non-offensive sentiment. The sentiment result is formatted accordingly and rendered in the output.html template, which displays the sentiment analysis result to the user through the web interface. Additionally, the system provides functionality for live sentiment analysis of audio input through the analyse_live endpoint. This feature records live audio, processes it for sentiment analysis, and displays the sentiment result in real-time.

6. Model Saving and Loading:

The sentiment analysis model (loaded_model) is saved and loaded using TensorFlow's model saving functionality. This ensures that the model can be easily reused across different sessions of the application without the need for retraining.

Overall, the architecture seamlessly integrates user interaction, sentiment analysis using BERT-based models, and result presentation, offering a user-friendly and efficient sentiment analysis system for text and audio inputs.

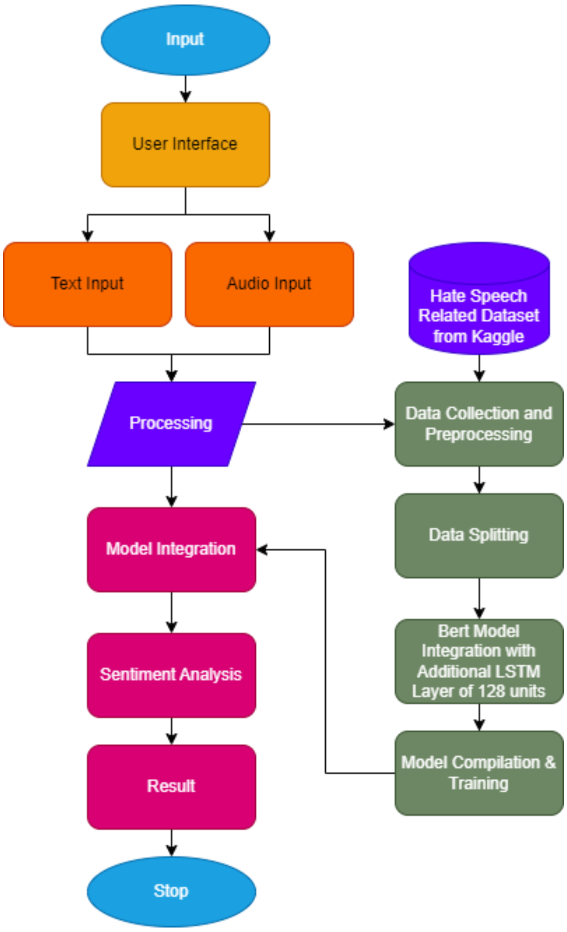

Input
User Interface
Text Input
Audio Input
Hate Speech Related Dataset from Kaggle
Processing
Data Collection and Preprocessing
Model Integration
Data Splitting
Sentiment Analysis
Bert Model Integration with Additional LSTM Layer of 128 units
Result
Model Compilation & Training
Stop

Fig 5: Overall Architecture Flowchart

V. EXPERIMENTATION AND RESULTS

The performance of the sentiment classification model was evaluated over the course of 6 training epochs using the training dataset. Each epoch involved training the model on batches of data and assessing its performance on both the training and validation sets. The training process yielded the following metrics for each epoch, Training loss steadily decreased across epochs, reaching a minimum of 0.0071 by epoch 6. Accuracy, precision, and recall on the training set also showed significant improvement, culminating in perfect scores (100.00%) by the final epoch. However, validation results displayed some fluctuations. While accuracy remained consistently high (above 94.05%), a slight increase in validation loss (0.2218) was observed in the final epoch. These findings warrant further investigation into potential overfitting issues and the generalizability of the algorithm on unseen data. We will evaluate the model's performance on the independent test dataset to offer a more thorough evaluation of its efficacy.

Following training, the model was assessed utilising the test dataset, yielding the following metrics:

- Loss: 0.4458
- Accuracy: 87.05%
- Precision: 86.30%
- Recall: 88.73%

Moreover, the confusion matrix provided insights into the model's performance, revealing that out of 139 test samples, 121 were correctly classified. The classification report further confirmed the model's effectiveness, demonstrating high precision and recall scores for both positive and negative sentiment classes. These results underscore the robustness and generalization capabilities of the sentiment classification model, affirming its suitability for practical applications in sentiment analysis of restaurant reviews.

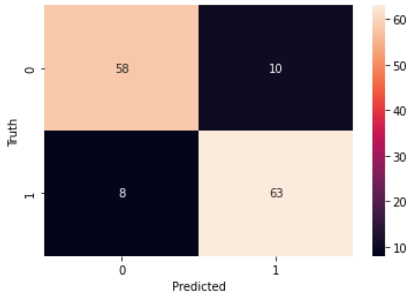


Fig 6: Confusion Matrix

	support	f1-score	recall	precision
0	68	0.87	0.85	0.88
1	71	0.88	0.89	0.86
Accuracy	139	0.87		
Wighted avg	139	0.87	0.87	0.87
Macro avg	139	0.87	0.87	0.87

Table 1: Classification Report

Model's Output:

The sentiment analysis model showcased consistent performance across different input modalities: text, audio, and live audio. In text input analysis, the model achieved increasing accuracy and robust generalization through six epochs of training, with balanced precision and recall scores. Audio input analysis demonstrated the model's ability to process audio inputs accurately, with sentiment predictions matching those from text inputs. Live audio analysis showcased the model's real-time processing capabilities, providing prompt sentiment analysis of spoken content. Overall, the model exhibited versatility and reliability across varied input formats, promising practical applications in customer feedback analysis and beyond.

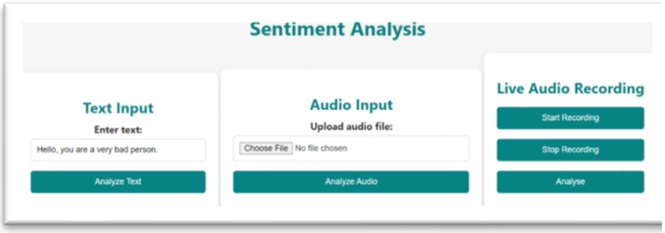


Fig 7: WebUI

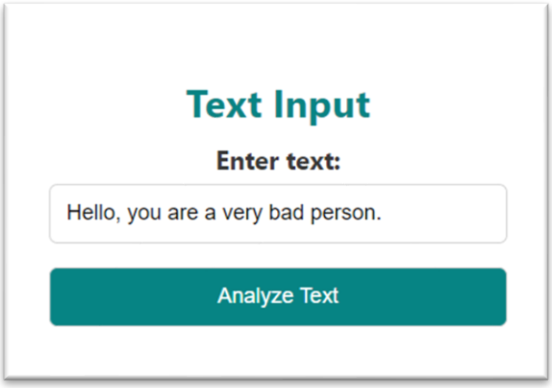
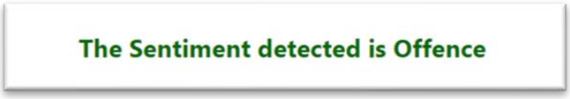


Fig 8: Text Input



The Sentiment detected is Offence

Fig 9: Text result

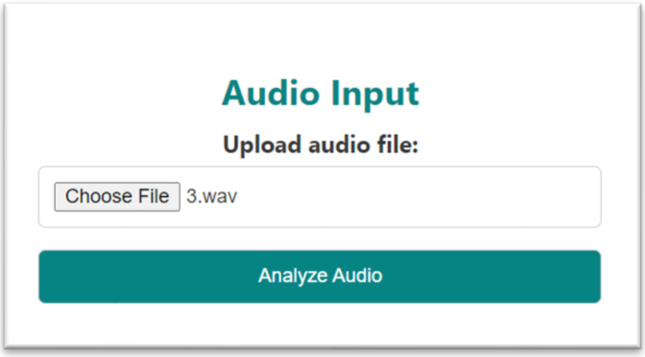
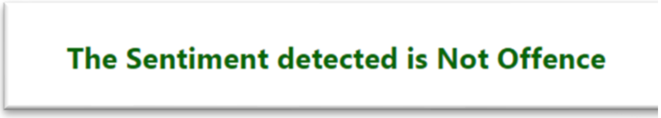


Fig 10: Audio Input



The Sentiment detected is Not Offence

Fig 11: Audio Result

VI. CONCLUSION

In conclusion, this project harnesses the power of BERT, a proven NLP model in sentiment analysis, to address the critical issue of hate speech detection. Our system transcends text analysis, incorporating audio processing for broader applicability across online platforms. Furthermore, a user-friendly web application empowers individuals to proactively combat hate speech through real-time analysis, fostering a safer online environment. While we leverage transfer learning with a fine-tuned BERT model, we acknowledge the potential for further refinement. Future directions include incorporating domain-specific data to enhance accuracy, exploring ensemble learning techniques to improve robustness, and integrating additional data sources like images or audio for a more comprehensive understanding of hate speech. By addressing these considerations, this project aspires to create a robust, user-centric, and generalizable solution. This solution has the potential to empower users, promote safer online spaces, and foster a more inclusive digital environment for all.

REFERENCES

[1] Saleh, Hind, Areej Alhothali, and Kawthar Moria. "Detection of hate speech using bert and hate speech word embedding with deep model." Applied Artificial Intelligence 37.1 (2023): 2166719.

[2] Mnassri, Khouloud, et al. "Bert-based ensemble approaches for hate speech detection." GLOBECOM 2022-2022 IEEE Global Communications Conference. IEEE, 2022.

[3] Kennedy, Brendan, et al. "Contextualizing hate speech classifiers with post-hoc explanation." arXiv preprint arXiv:2005.02439 (2020).

[4] Almaliki, Malik, et al. "Abmm: Arabic bert-mini model for hate-speech detection on social media." Electronics 12.4 (2023): 1048.

[5] Alatawi, Hind S., Areej M. Alhothali, and Kawthar M. Moria. "Detecting white supremacist hate speech using domain specific word embedding with deep learning and BERT." IEEE Access 9 (2021): 106363-106374.

[6] Sohn, Hajung, and Hyunju Lee. "Mc-bert4hate: Hate speech detection using multi-channel bert for different languages and translations." 2019 International Conference on Data Mining Workshops (ICDMW). IEEE, 2019.

[7] Keya, Ashfia Jannat, et al. "G-bert: an efficient method for identifying hate speech in Bengali texts on social media." IEEE Access (2023).

[8] Miok, Kristian, et al. "To ban or not to ban: Bayesian attention networks for reliable hate speech detection." Cognitive Computation 14.1 (2022): 353-371.

[9] Dukic, David, and Ana Sovic Krzic. "Detection of Hate Speech Spreaders with BERT." CLEF (Working Notes). 2021.

[10] Vishwamitra, Nishant, et al. "On analyzing covid-19-related hate speech using bert attention." 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2020.

[11] "BERT Explained: State of the Art Language Model for NLP" by Jay Alammar https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

[12] Bhawal, Snehaan, Pradeep Roy, and Abhinav Kumar. "Hate Speech and Offensive Language Identification on Multilingual Code Mixed Text using BERT." FIRE (Working Notes). 2021.

[13] Rana, Aneri, and Sonali Jha. "Emotion based hate speech detection using multimodal learning." arXiv preprint arXiv:2202.06218 (2022).

[14] Mazari, Ahmed Cherif, Nesrine Boudoukhani, and Abdelhamid Djeffal. "BERT-based ensemble learning for multi-aspect hate speech detection." Cluster Computing (2023): 1-15.

[15] Gupta, Shailja, Sachin Lakra, and Manpreet Kaur. "Study on bert model for hate speech detection." 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA). IEEE, 2020.