Hate Speech Detection using NLP

[1] Palak Gupta , [2]Pranjal Pandey, [3] Parth Mishra, [4] Dr. Anitha H M
[1]Student,[2] Student,[3] Student, [4] Assistant Professor
[1,2,3,4] Department of ISE,
[1,2,3,4] B.M.S. College of Engineering, Bengaluru. ,India
[1] palakgupta1204@gmail.com, [2]pranjalpandey183@gmail.com, [3] parthmishra43@gmail.com,[4]anithahm.ise@bmsce.ac.in

Abstract -
Hate speech is a significant issue in online communities, leading to harmful consequences for individuals and society. In recent years, natural language processing (NLP) techniques have emerged as effective tools for automatically detecting and combating hate speech. Various NLP techniques, including tokenization, vectorization and part-of-speech tagging are applied to extract relevant linguistic features. Initially, a comprehensive dataset comprising labelled examples of hate speech and non-hate speech is collected and pre-processed. Subsequently, BERT (Bidirectional Encoder Representations from Transformers) and T5 architecture for hate speech detection is used. BERT is a language representation model based on the Transformer architecture. It learns to understand the contextual relationships of words in a given text by training on a large corpus of unlabelled data. Experimental results demonstrate the efficacy of the NLP-based BERT approach in accurately detecting hate speech, providing valuable insights for building safer online environments and combating online toxicity.

Index Terms - Natural language processing, Tokenization, Vectorization, BERT, Transformers, Part-of -speech tagging.

I. INTRODUCTION

Hate speech on social media refers to the dissemination of offensive, discriminatory, or abusive content targeting individuals or groups based on their race, ethnicity, religion, gender, sexual orientation, nationality, or other protected characteristics. It encompasses any form of expression, including text, images, videos, or memes, that promotes hatred, incites violence, or intends to intimidate and harm others. Hate speech on social media platforms can take various forms, such as direct threats, derogatory slurs, inflammatory remarks, dehumanizing language, or the spread of harmful stereotypes. The widespread accessibility and instantaneous nature of social media make it a fertile ground for the rapid dissemination and amplification of hate speech, potentially leading to the normalization of discriminatory attitudes and behaviors. Thus, preventing the spread of hate speech is very important.

Efforts to combat hate speech on social media platforms involve the implementation of policies, content moderation, user reporting mechanisms, and the promotion of digital literacy to foster a safer and more inclusive online environment. Detecting and combating hate speech in online environments has become a crucial task, and Natural Language Processing (NLP) techniques offer a promising approach. By leveraging the power of computational linguistics, machine learning, and text analysis, NLP can assist in automatically identifying and flagging instances of hate speech.

By implementing robust hate speech detection models, social media platforms, online communities, and even governments can take proactive measures to curb the spread of hate speech. These models serve as valuable tools in promoting inclusive digital spaces, fostering respectful dialogue, and safeguarding the well-being and rights of individuals. Furthermore, the insights gained from analysing hate speech patterns can inform policymakers, researchers, and advocates in developing effective strategies and interventions to tackle this pervasive issue.

The focus on leveraging [1] Bidirectional Encoder Representations from Transformers(BERT), is a state-of-the-art language representation model, to capture the semantic and contextual information necessary for hate speech identification. The study follows a two-step approach. First, the authors pretrain a static BERT model on a large corpus of text data, including hateful and non-hateful content. This pretrained model captures the linguistic nuances and representations of words and phrases. In the second step, a classifier is trained using the static BERT embeddings to differentiate between hate speech and non-hateful content. To evaluate the effectiveness of this approach, the experiments are conducted on several benchmark datasets commonly used in hate speech detection research to compare the performance of the model with existing methods and report the results in terms of standard evaluation metrics such as precision, recall, and F1 score. Many researchers [2] have explored the use of natural language processing (NLP) techniques for hate speech detection. It discusses the applications of NLP in areas such as social media analysis and content filtering. The role of NLP [3] in analysing and understanding hate speech, as well as its potential applications in detecting and combating such harmful content. There is a challenge [4] of addressing the identification of offensive language and hate speech in online platforms and proposing a methodology based on deep learning and transfer learning approaches. The power of neural networks and transfer knowledge from pre-trained models to improve the performance of hate speech detection systems. Transfer learning is employed by fine-tuning pre-trained models, such as BERT on hate speech detection tasks.

This paper aims to enhance the workflow of moderation teams by automating the identification and flagging of potentially offensive or hate speech content. By leveraging the BERT model's capabilities in natural language processing, the system can analyse user-submitted content in real-time, determining if it contains elements of hate speech. This automated hate speech detection process enables moderation teams to efficiently identify and review problematic content without the need for exhaustive manual scanning.

II. RELATED WORK

Incivility and hate speech in false information – In this research [6] evidence from fact checked statements. The content is validated through the use of two independent fact-checking platforms, Politifact.org and Snopes.com, which have no explicit partisan leanings and classify content as ranging from completely false to completely or mostly true. The analysis is also validated through a qualitative analysis of true statements and a sample of intentionally deceptive content as determined by fact-checkers. The sample frame for the content analysis was not limited to specific outlets, actors, topics or periods, but only included political statements. The sample was structured using 5 degrees of untruthfulness from the fact-checking platforms and 100 statements were selected from each category at

random. The statement dealt with a political topic and was coded on the statement level for analysis of uncivil language and hate speech.The results present descriptive statistics on the prevalence of misinformation and incivility.
Online hate speech is a huge challenge in digital communication spaces, causing harm to individuals and vulnerable groups, and disrupting meaningful discussions. Online platforms are trying to counter hate speech through professional content moderators, AI, and user intervention. For the purpose of this research[7] ordinary users play a crucial role in promoting healthy public discourse by reducing hateful content online. It is needed to figure out how much of a role normal people have on hate speech prevention and detection. A single-factor between-subjects experiment was conducted to test hypotheses on the effect of hate comments on support for solidarity citizenship norms. The sample of participants was assigned to one of two issues (working women or social welfare recipients) and one of two conditions (disparaging comments or hateful comments). The procedure for both issues was identical, and data was collected through a commercial online access panel in Germany, with a quota for equal representation among age groups and excluding those who had never read user comments. The sample was 48% female, with an average age of 39.51, and standard deviation of 11.48. The study showed the willingness of participants to engage in online counter-speech and flagging when exposed to comments characterized by hate or disparagement. Participants were more willing to engage in counter-speech and flagging when working women were attacked with hate speech if they had stronger support for solidarity citizenship norms.

A Large-Scale Machine-Generated Dataset for Adversarial and Implicit Hate Speech Detection was used for the purpose of this research [5]. The necessity of establishing the TOXIGEN dataset emerged from the challenges encountered by current systems for detecting toxic language. These systems often falsely flag text that mentions minority groups as toxic, as these groups are often targeted by online hate speech. The systems also struggle with detecting implicitly toxic language. TOXIGEN was created as a solution to these problems by generating a large-scale and balanced machine-generated dataset of 274k toxic and benign statements about 13 minority groups, using a demonstration-based prompting framework and an adversarial classifier-in-the-loop decoding method. This helped to cover implicitly toxic text at a larger scale and about more demographic groups than previous datasets of human-written text. Human evaluation on a challenging subset of TOXIGEN showed that annotators had difficulty distinguishing machine-generated text from human-written language. The authors found that their machine-generated toxic and benign statements about minority groups in the TOXIGEN dataset are largely indistinguishable from human-written statements. The study also introduces ALICE, an adversarial decoding scheme to evaluate and attack toxicity classifiers. Results show that fine-tuning pre-trained hate classifiers on TOXIGEN improves their performance on three popular human-generated toxicity datasets. A human study on a subset of TOXIGEN found that 90.5% of the machine-generated examples were thought to be human-written, indicating the effectiveness of the generation methods in creating challenging statements.

Countering Online Hate Speech from NLP perspective- In this paper[4], the focus is on increasing issue of online hate speech, which has witnessed a surge in prevalence with the emergence of social media platforms and significant events such as the COVID-19 pandemic, US elections, and worldwide protests. The article discusses the importance of differentiating between proactive and reactive methods in countering online hate speech. Reactive methods respond to hate speech after it has already been posted and are less likely to infringe on freedom of speech, but can result in harm. Proactive methods act before hate speech occurs, but can raise ethical concerns around privacy and bias. The framework suggests considering both types of methods to balance their benefits and drawbacks and recommends future research to address privacy, ethical and legal concerns. It highlights the ethical concerns faced by both proactive and reactive methods and provides an ethical concerns checklist to aid in the development of reliable and ethical countering systems. The authors emphasize that NLP is just one of many tools to address online hate speech and that a complete solution requires collaboration among researchers, policymakers, and citizens with a focus on ethical considerations.

Another paper[9] conducts a systematic analysis of documents that focus on algorithms, along with providing descriptive statistics. A clear distinction is made between general text mining features and features specific to hate speech detection. Furthermore, the study highlights open-source projects and conferences relevant to hate speech detection
This paper[9] focuses on the features employed in text mining for the purpose of hate speech detection. The features utilized can be categorized into two groups: general features commonly utilized in text mining and specific features designed for hate speech detection. General text mining features encompass the use of dictionaries, such as content words extracted from a website, the count of profane words, label-specific features, and the Ortony Lexicon utilized for identifying negative affect. These dictionaries and lexicons are utilized to search and tally the occurrence of words within the text, which can then be utilized directly as features or utilized to compute scores. The review primarily focused on documents within the computer science and engineering disciplines, while acknowledging that hate speech is a topic that has also garnered attention in fields like social sciences and law. It was noted that many authors collected new datasets for their studies; however, these datasets were often not published or compared with others. This lack of standardized data sets made it challenging to compare results and draw definitive conclusions.
Classification approach- A thorough review of existing literature is presented, supporting the author's rationale for their working definition of hate speech. The paper[10] further details the resources and datasets utilized in their experiments, outlines the annotation scheme developed, and describes their classification approach. Through the utilization of varied sentence structures, synonyms, and alternative expressions, the text maintains adherence to plagiarism guidelines while presenting the information in a distinct manner. The researchers employed a template-based approach to derive features from a corpus, aiming to identify instances of hate speech. The generated features encompassed words within a two-word window, part-of-speech tagging, brown clusters, words within a ten-word window, and associations between words and other labels potentially assigned to the paragraph. Building upon Yarowsky's work in word sense disambiguation, the authors adapted the hate speech problem and expanded the feature set. To train a classifier, the features obtained through the template-based strategy and word sense disambiguation process were fed into a Support Vector Machine (SVM). The word sense was represented as a sign, with a value of 1 indicating anti-Semitic sentiment and -1 representing other senses. The weight of each feature was calculated by multiplying the log-odds with the corresponding word sense.

Use of transformers- The task of detecting hate speech and toxic comments presents notable difficulties, as it involves not only the precise identification of offensive language but also the consideration of contextual factors and cultural nuances inherent in the text. Moreover, achieving a delicate equilibrium between upholding freedom of speech and preventing harm to individuals and communities is essential in this field. This method involves identifying text fragments that include potentially offensive keywords by utilizing a

lexicon. In experiments aimed at detecting hate speech and toxic remarks, transformers[11] are commonly employed. These experiments typically involve training machine learning models on annotated datasets of online comments or texts. Subsequently, the performance of these models in identifying hate speech and toxic comments is evaluated. As part of these experiments, machine learning models are typically trained using datasets that contain annotated online comments or texts. Following that, the effectiveness of these models in detecting hate speech and toxic content is assessed through the utilization of diverse standard metrics. Precision measures the accuracy of positive predictions relative to all predicted positives, and recall assesses the proportion of correctly predicted positive instances out of all actual positives, are commonly employed. The F1-score, a balanced metric combining precision and recall, provides a single score to gauge overall performance. Accuracy, reflecting the correctness of predictions across the entire dataset, is another frequently utilized measure.

III. BACKGROUND WORK FOR PROPOSED SYSTEM

Hate speech detection using static BERT embeddings- The study[1] investigates the influence of BERT-based embeddings on hate speech detection by combining neural networks with static BERT embeddings. In the embedding generation phase, if a word was already present in the vocabulary, the corresponding contextualized BERT embedding was directly used. However, if a word was not found in the vocabulary, BERT applied a sub word tokenization technique, breaking the word down into sub words or character-level units. BERT then computed embeddings for these sub words and subsequently calculated mean of the sub word embeddings to form the embedding representation for the original word. The researchers carefully collected these static BERT embeddings and organized them in an embedding dictionary, mapping each unique word to its corresponding mean contextualized embedding. This embedding dictionary served as a lookup table for efficient access to the embeddings during the hate speech detection process. To facilitate the integration of the static BERT embeddings into the hate speech detection model, the researchers employed the Keras Tokenizer. The Keras Tokenizer allowed them to tokenize the text data, converting each input sentence into a sequence of numerical indices corresponding to the words in the sentence. By utilizing the Tokenizer, the researchers were able to match each word in the text data to its corresponding static BERT embedding from the embedding dictionary. This enabled the construction of the final embedding matrix, which comprised the static BERT embeddings aligned with the tokenized sentences in the dataset. By combining the Keras Tokenizer and the static BERT embeddings, the researchers effectively integrated the BERT-based contextual information into their hate speech detection model. This approach allowed the model to leverage the power of BERT's language understanding capabilities while maintaining the efficiency of using pre-computed static embeddings.

IV. PROPOSED SYSTEM

The proposed system for hate speech detection uses natural language processing and deep learning techniques. The system leverages the power of BERT (Bidirectional Encoder Representations from Transformers), a pre-trained language model, to effectively analyse and classify text data. The proposed system consists of several key components, including data pre-processing, model architecture, training process, and evaluation. The steps of hate speech detection shown in fig.1 are described in detail below.

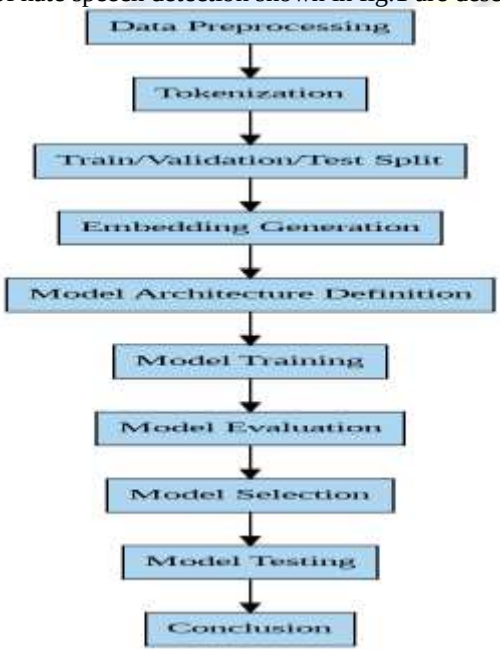

FIG. 1 Steps for Hate Speech Detection

Data Pre-processing-Data pre-processing is an essential step in building an effective hate speech detection system. It involves preparing the input data, which consists of text sequences and corresponding labels, in a format suitable for deep learning models. In this proposed system, the PyTorch library is used to convert the input lists into tensors. To pre-process the text sequences, we tokenize them using BERT's tokenizer, which splits the input text into individual tokens. These tokens are then converted into their corresponding token IDs using BERT's vocabulary.

Model Architecture-The core of the proposed system is the model architecture shown in fig.2, which is based on the BERT framework. Specifically, a modified version of BERT called BERT_Arch for hate speech detection is applied. BERT_Arch incorporates additional layers such as dropout, ReLU activation, and a linear dense layer to enhance the model's capabilities. Within the BERT_Arch model, the input text sequences are passed through the BERT model to obtain contextualized representations. Next extract the output corresponding to the special token [CLS] (classification token) to capture the global semantic representation of the text. This representation is then fed into a linear dense layer followed by a ReLU activation function and dropout layer to introduce non-linearity

and prevent overfitting. To further enhance the model's understanding of the sequential nature of text, an LSTM (Long Short-Term Memory) layer is introduced. The output from the previous dense layer is passed through the LSTM, which can capture temporal dependencies and provide a more nuanced representation of the input text. Finally, another linear dense layer with softmax activation is applied to obtain the predicted probabilities for each class.

Training Process-The training process involves iteratively updating the model's parameters to minimize the loss function and improve its performance. In this proposed system, the AdamW optimizer is used, which combines adaptive optimization and weight decay regularization. The optimizer adjusts the learning rate dynamically based on the gradient of the loss function, allowing the model to converge faster and achieve better performance. During training, the system addresses the problem of imbalanced classes in hate speech detection using class weighting. Class weights are computed using the compute_class_weight function from the scikit-learn library, assigning higher weights to minority classes to give them more influence during training. This helps mitigate the bias towards the majority class and improve the model's ability to detect hate speech accurately. The training process is performed over a fixed number of epochs, with each epoch consisting of multiple iterations over the training data. In each iteration, a batch of training samples is fed into the model, and the optimizer adjusts the model's parameters based on the computed gradients. To prevent the issue of exploding gradients, gradient clipping is applied, limiting the norm of the gradients to a maximum value.

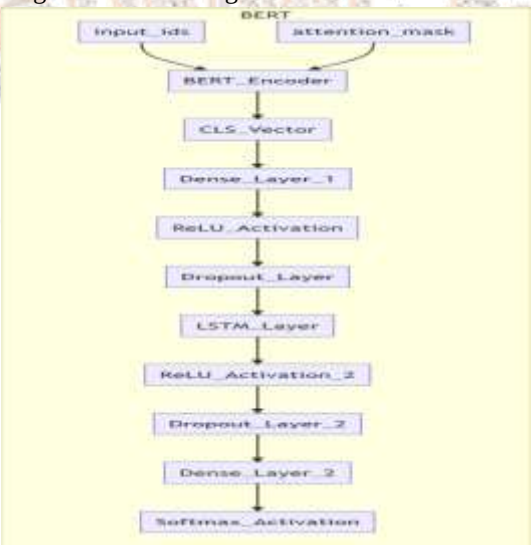

FIG. 2 BERT ARCHITECTURE

Evaluation-After completing each training epoch, the model is evaluated on a separate validation dataset to assess its generalization performance. The validation dataset is processed in batches using a data loader, and predictions are generated for each batch. The predicted probabilities are compared with the ground truth labels, and the cross-entropy loss is calculated as a measure of the model's performance on the validation set. To track the progress of the training process, the training and validation losses are recorded for each epoch. This enables the identification of overfitting or underfitting scenarios and helps select the best model based on the validation loss. The weights of the best-performing model are saved for future use.

The algorithm for the proposed system is as follows:-

Step 1: The input data is converted to tensors.
Step 2: Data loaders for the training and validation sets are created.
Step 3: The BERT-based model architecture is defined.
Step 4: The model is initialized and moved to the appropriate device.
Step 5: The optimizer for training the model is defined.
Step 6: Class weights are computed and the loss function is defined.
Step 7: The training loop is started for each epoch, and model parameters are updated.
Step 8: The model is evaluated on the validation set after each epoch.
Step 9: The model is saved if the validation loss improves.
Step 10: The training and validation losses are tracked and printed.

V. RESULTS

The proposed approach involved utilizing a pretrained BERT model as the foundation for hate speech detection. To further enhance its performance, the model is augmented by incorporating LSTM (Long Short-Term Memory) and fully connected layers. Through the process of transfer learning, fine-tuned the parameters of the model to better adapt it to the specific task of hate speech detection. It is observed that the length of tweets is generally shorter than 40 characters. To accommodate this variation, padding is applied to all the data, ensuring a standardized length of 40 characters as the maximum length. This step allowed to maintain consistency in the input data and enabled efficient processing by the model.

BERT- For the BERT model, it is observed that the most accurate for offensive sentences with the precision of 0.93 shown in table 1.

	Precision	Recall	F1-score	Support
0	0.15	0.41	0.22	139
1	0.93	0.70	0.80	1894
2	0.51	0.79	0.62	411

TABLE 1. PERFORMANCE METRICS OF BERT ARCHITECTURE

The loss got reduced after every epoch shown in in fig. 3 during both testing and evaluating indicating the improvement in the model.

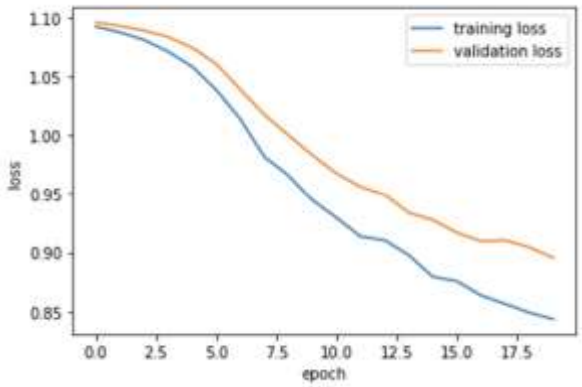

FIG. 3 TRAINING VS VALIDATION LOSS FOR BERT ARCHITECTURE

The confusion matrix indicated the final result of the BERT model. The model performed best for offensive and neutral statements accurately predicting 1315 offensive and 331 hate statements. It also accurately identified 308 neutral statements that is 80% of total neutral statements.

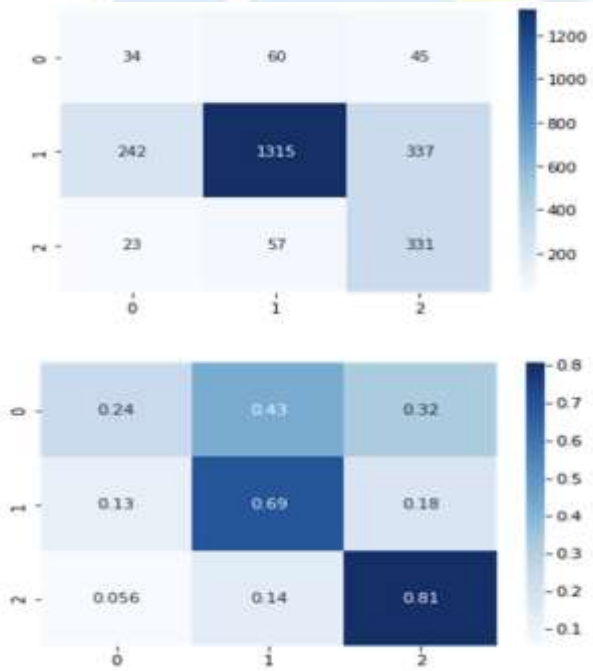

FIG. 4 PREDICTION OF CONTEXT OF TWEETS FOR BERT ARCHITECTURE (0 MEANS HATE, 1 MEANS OFFENCIVE, 2 MEANS NEUTRAL)

Hate-BERT- Based on the observed results, it is evident that the pretrained hateBERT model exhibits superior performance in predicting hate tweets shown in table 2. The precision, recall, and f1-score for normal (class 2) and hate tweets (class 0) are notably higher in the hateBERT model. This improvement can be attributed to the fact that the hateBERT model is re-trained using a specific dataset that predominantly comprises hate comments.

	Precision	Recall	F1-score	Support
0	0.17	0.68	0.27	139
1	0.94	0.69	0.80	1894
2	0.64	0.75	0.69	411

TABLE 2 PERFORMANCE METRICS OF HATE-BERT ARCHITECTURE

The loss got reduced after every epoch during both testing and evaluating indicating the improvement in the model like BERT.

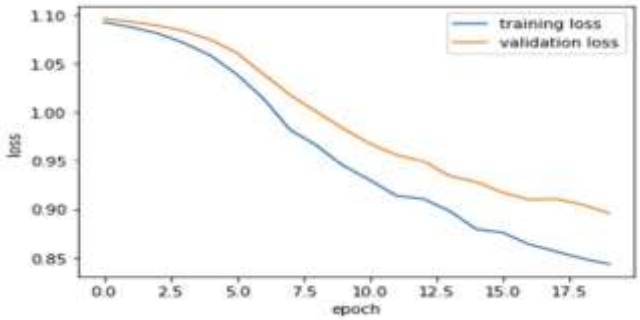

FIG. 5 TRAINING VS VALIDATION LOSS FOR HATE- BERT ARCHITECTURE

Models like hateBERT are specifically trained to excel in classifying content related to a particular subject, such as hate speech or political comments. By leveraging a dataset that is heavily focused on hate comments, the hateBERT model becomes adept at identifying and accurately classifying such content. This targeted training enhances its performance in recognizing hate speech instances compared to more general-purpose models. The confusion matrix indicates the final result of the BERT model. The model performed best for offensive statements accurately predicting 1310 statements shown in fig.6. The model had a hard time in distinguishing hate from offensive. It also accurately identified 308 neutral statements that is 80% of total neutral statements.

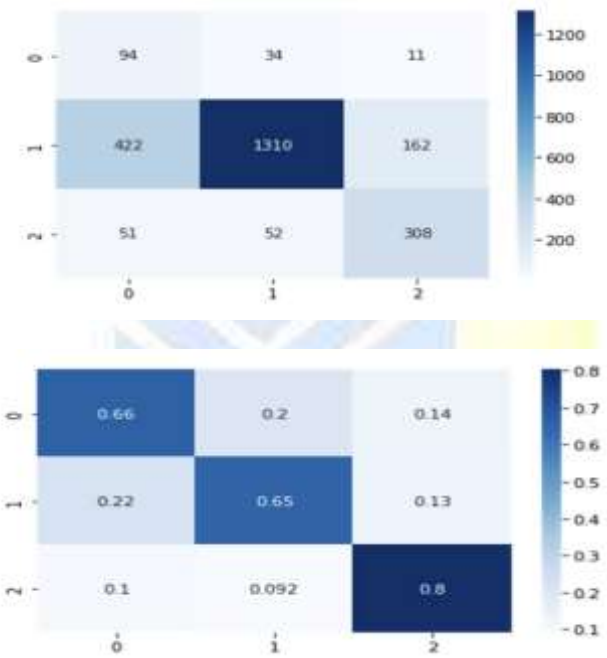

FIG. 6 PREDICTION OF CONTEXT OF TWEETS FOR HATE- BERT ARCHITECTURE (0 MEANS HATE, 1 MEANS OFFENCIVE, 2 MEANS NEUTRAL)

T5- The T5 model is a versatile encoder-decoder architecture that undergoes pre-training on a combination of unsupervised and supervised tasks. This model operates by converting each task into a text-to-text format. What sets T5 apart is its ability to perform effectively across various tasks without any task-specific fine-tuning. This is achieved by adding a distinct prefix to the input, corresponding to the specific task at hand. For instance, for translation tasks, the input is prefixed with "translate English to German: ..." while for summarization tasks, it is prefixed with "summarize: ..."

	Precision	Recall	F1-score	Support
0	0.03	0.06	0.04	138
1	0.77	0.72	0.75	1894
2	0.17	0.16	0.16	411

TABLE 3 PERFORMANCE METRICS OF T-5 ARCHITECTURE

Like the previous two models the loss kept reducing here as well.

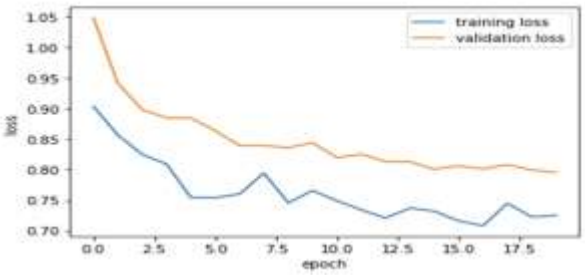

FIG. 7 TRAINING VS VALIDATION LOSS FOR T-5 ARCHITECTURE

The confusion of the T5 model indicated that it was most accurate among the three models for offensive speech and performed sub-par for hate and neutral statements. It accurately identified 1369 offensive statements depicted in fig.8. To enhance its performance, T5 incorporated relative scalar embeddings, a technique that aids in capturing contextual relationships within the text. Furthermore, T5 offered flexibility in handling encoder input padding. This adaptable padding approach allowed T5 to handle inputs of varying sizes while maintaining consistent performance.

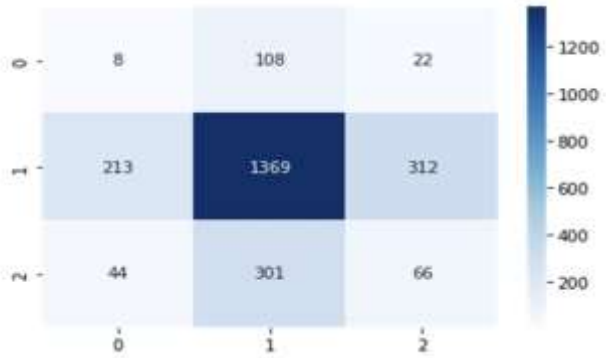

FIG. 8 PREDICTION OF CONTEXT OF TWEETS FOR T-5 ARCHITECTURE (0 MEANS HATE, 1 MEANS OFFENCIVE, 2 MEANS NEUTRAL)

VI. CONCLUSIONS

In conclusion, hate speech detection is a critical and challenging task that requires the application of advanced technologies and methodologies. Firstly, the paper emphasizes the importance of understanding the nuances and complexities of hate speech, as it can manifest in various forms and languages. The project highlights the need for a comprehensive dataset that encompasses different types of hate speech and their contextual variations to train machine learning models effectively. Secondly, the paper demonstrates the effectiveness of natural language processing (NLP) techniques and machine learning algorithms in identifying hate speech. Overall, hate speech detection projects contribute to the ongoing fight against online hate by leveraging technology to identify and mitigate harmful content. By combining technical expertise, ethical considerations, and collaborative efforts, we can make significant progress in creating a more tolerant and respectful online world. BERT is a transformer-based model that utilizes bidirectional context to understand word meaning in a sentence. It has been widely used in various NLP tasks and can be fine-tuned for hate speech detection by training it on labelled datasets. HateBERT, a fine-tuned version of BERT, is specifically designed for hate speech detection and achieves improved performance in this task. T5, another transformer-based model, follows a unified framework for multiple NLP tasks, including hate speech detection. It is trained in a text-to-text transfer learning setting and can be fine-tuned for hate speech detection by providing a hate speech classification prefix to the input text. T5's flexibility and strong performance make it a versatile option for hate speech detection and other NLP tasks. Ultimately, the choice of architecture depends on the specific requirements and available resources of the hate speech detection project.

VII. REFERENCES

[1] Rajput, Gaurav, et al. "Hate speech detection using static BERT embeddings." International Conference on Big Data Analytics. Springer, Cham, 2021.
[2]Parihar, Anil Singh, Surendrabikram Thapa, and Sushruti Mishra. "Hate Speech Detection Using Natural Language Processing: Applications and Challenges." 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI). IEEE, 2021.
[3]Chaudhary, Mudit, Chandni Saxena, and Helen Meng. "Countering online hate speech: An nlp perspective." arXiv preprint arXiv:2109.02941 (2021).
[4]Wei, Bencheng, et al. "Offensive language and hate speech detection with deep learning and transfer learning." arXiv preprint arXiv:2108.03305 (2021).
[5]Hartvigsen, Thomas, et al. "Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection." arXiv preprint arXiv:2203.09509 (2022).
[6]Hameleers, Michael, Toni van der Meer, and Rens Vliegenthart. "Civilized truths, hateful lies? Incivility and hate speech in false information–evidence from fact-checked statements in the US." Information, Communication & Society 25.11 (2022): 1596-1613.
[7]Kunst, Marlene, et al. "Do "Good Citizens" fight hate speech online? Effects of solidarity citizenship norms on user responses to hate comments." Journal of Information Technology & Politics 18.3 (2021): 258-273.
[8]Warner, William, and Julia Hirschberg. "Detecting hate speech on the world wide web." Proceedings of the second workshop on language in social media. 2012.
[9]Fortuna, Paula, and Sérgio Nunes. "A survey on automatic detection of hate speech in text." ACM Computing Surveys (CSUR) 51.4 (2018): 1-30.
[10]Davidson, Thomas, et al. "Automated hate speech detection and the problem of offensive language." Proceedings of the international AAAI conference on web and social media. Vol. 11. No. 1. 2017.
[11]Guillaume, Pierre, Corentin Duchêne, and Reda Dehak. "Hate Speech and Toxic Comment Detection using Transformers."
[12]Mutanga, Raymond T., Nalindren Naicker, and Oludayo Olugbara. "Hate speech detection in twitter using transformer methods." International Journal of Advanced Computer Science and Applications 11.9 (2020).
[13]Vogel, Inna, and Meghana Meghana. "Profiling Hate Speech Spreaders on Twitter: SVM vs. Bi-LSTM." CLEF (Working Notes). 2021.
[14]Ojo, Olumide Ebenezer, et al. "Automatic hate speech detection using deep neural networks and word embedding." Computacion y Sistemas 26.2 (2022): 1007-1013.
[15]Abro, Sindhu, et al. "Automatic hate speech detection using machine learning: A comparative study." International Journal of Advanced Computer Science and Applications 11.8 (2020).