BERT vs. Traditional Machine Learning Models for Hate Speech Detection on Twitter

Chinmay Naik
School of computer science engineering and technology,
Bennett University
Greater Noida, India

Parikshit Parihar
School of computer science engineering and technology,
Bennett University
Greater Noida, India

Ritik Tomar
School of computer science engineering and technology,
Bennett University
Greater Noida, India

Ayush Sharma
School of computer science engineering and technology,
Bennett University
Greater Noida, India

***Abstract*—This study explores the potency of BERT (Bidirectional Encoder Representations from Transformers) in categorizing hate speech and offensive languages. Over wide range of natural language processing tasks, including classification tasks, BERT, a pre-trained model, using Transformer structure has shown exceptional results. We test BERT's accuracy in detecting hate speech and offensive language through rigorous testing, progressing content moderation system and reducing online toxicity. Our study emphasize how impactful BERT can be in blocking harmful speech and boosting safer communities online.**

Keywords—BERT, SVM, Random Forest Classifier, Decision Tree Classifier

I. INTRODUCTION

In this digital age, social media platforms have become an essential part of our life where we share our opinions and connect with other people. Nevertheless, the uncontrolled and unrestricted nature of these platforms has resulted in the spread of harmful content involving hate speech and offensive language. To tackle this issue, we need effective methods and strategies that can classify such content, thus ensuring a safer online environment.

Sentiment analysis, a subfield of natural language processing (NLP), plays crucial role in automating the process of recognizing and classifying such large amount of data. Through utilizing sentiment analysis, platform moderators can filter out such detrimental material from their platforms making it more user-friendly

By preprocessing the textual data and extracting the right features, traditional machine learning models like Decision Tree [9] [12], Random Forest [15], Logistic Regression [13], Support Vector Machine (SVM) [8] [11] work effectively in categorizing such data. We have used these models for the sentiment analysis task and evaluated them on certain metrics like accuracy, F1 score, recall value, precision score.

However, despite their effectiveness, traditional ML models often face limitations when confronted with the complexities and nuances of natural languages. Due to changes in form of textual expression and evolving linguistic trends, traditional models face difficulties in categorizing hate speech where subtle linguistic cues and cultural context are paramount.

To address these challenges and enhance the performances of sentiment analysis systems, recent advancements in deep learning have spurred the development of transformative models like BERT (Bidirectional Encoder Representations from Transformers) [10] [14]. BERT, a state-of-the-art NLP model introduced by Google AI, revolutionized the fields by leveraging large-scale pretraining and bidirectional context modellings to achieve unparalleled performance across various linguistic analysis tasks.

In this research paper, we explore the efficiency of both traditional ML models and the BERT model in classifying Twitter data into categories of hate speech, offensive language, or neither. By conducting a comparative analysis, we aim to elucidate the strengths and weaknesses of each approach, shedding light on the potential of advanced deep learning techniques like BERT to augment sentiment analysis capabilities in the context of social media moderation and online content moderation.

II. LITERATURE REVIEW

This paper worked on sentiment analysis which excelled in detecting sentiment in short textual messages [1]. Various features were added like sentiment lexicons derived from tweets which were then implemented to improve sentiment analysis prediction. After research on negation on sentiment analysis, lexicons for aspect in positive and invalidated contexts were developed, significantly improving the performance of system. This system can process up to 100 tweets per second and is applied to identify target entity sentiments, textual contents and political behaviors. Their upcoming work will include translation of system into additional languages, perfecting sentiment lexicons and examining more algorithms to manage different sentiment modifiers.

They have developed a sentiment analysis method based on adverb-adjective combinations (AAC) that utilizes textual examination of degree-related adverbs [2]. All adverb scoring methods must abide to a set of general axioms, based on classification of degree of adverbs into five categories. Instead of combining scores of adverbs and adjectives by simple scoring, they offer unarguable treatment of AACs based on verbal classification of adverbs. Three particular AAC scoring techniques are presented that meet the axioms. They define their results of findings and experiments on a set of 200 news articles and compare their algorithms with the existing sentiment analysis algorithms. Their aim is to provide higher accuracy based on Pearson correlation with human subjects.

This paper [3] works on several methods of sentiment analysis and different level of inspecting sentiments with the aim of expertly categorizing reviews. Several machine learning methods were also implemented like SVM, Naïve Bayes and Maximum Entropy along with several more methods that can improvise analysis process. Rather than evaluating by word-by-word analysis they have used n-gram evaluation to improve semantic analysis. Rule based and lexicon-based approaches were also used. More emphasis was placed on how social networking is providing information. When it was matter of decision-making, examining blog reviews provided excellent insight. Thus, this study provides more effective ways to analyze sentiments through online platforms. By using online review sentiment analysis, these methods aim to expand decision making process.

The researching paper delves deeply into emotion study in conditional statements utilizing language analysis and machine learning approaches. The Machine learning algorithm. Assistance vector machines (SVM) is used to anticipated emotion. The emotion foretelling characteristics are constructed. Emotion words and expressions, clause lengths, conditional connectors, negation terms, strain designs, unusual characters, and pieces-of-language labels are a few samples of these. There are three methods of classification employed: Resulted-based classification, whole-sentence-based classification, clause-based classification. The effectiveness of the recommended tactic is displayed by the empirical findings. Techniques to enhance the precision of sentiment prediction are scrutinized, involving manipulating negation phrases and adjusting the window extent circumventing subjects [4].

This paper [5] provides a survey on challenges encountered in sentiment analysis based on two com- parisons among forty-seven papers. The first comparison is based on relationship between sentiment review structure and sentiment analysis challenges, revealing the significance of domain-dependence and the prevalence of the negation challenge across different review structures. The second comparison examines sentiment analysis challenges in relation to accuracy rates. It emphasizes the importance of selecting appropriate challenges to enhance accuracy, noting a correlation between the proportion of sentiment techniques used and the type of challenges addressed.

The report performed Sentiment Analysis that includes differentiating between text with opinions and text with facts and determines the polarity of opinionated text. This report introduces A-SVM [6], a system that uses Machine Learning techniques and user opinions to differentiate between text with opinions and text with facts. A-SVM is compared with traditional machine learning methods like Support Vector Machines (SVMs) in classifying a body of annotated text. The results displayed that Support vector machines (SVMs) performed more worse than A-SVM, presenting a 5.6% increase in classification precision from 78.1% with SVMs to 83.7% with A-SVM per the evaluation.

This study shows that nowadays people share their views experiences, opinions which changed the way of communicating or influences with each other. Here Sentiment Analysis is used which understand whether the statement is positive, negative or neutral because it is important to for all of us as people share their views online. It explains the ways of analyzing the sentiment analysis by data collection, text preparation, detection, classification which detect the opinion of the text i.e. positive, negative or neutral. The paper also explains about the different approach for sentiment classification first one is machine learning which predict the polarity on the basis of the trained data sets. Second approach is lexicon which are list of words and do not require any trained data for classification. Last one is hybrid approach, which is the combination of both the above approaches to improve the sentiment classification. It mentions various tools used for sentiment analysis, like Emoticons, LIWC, Senti WordNet, and others. These tools help figure out if a message is positive or negative. Further this paper ends by explaining that different approaches and specified tools can be applied in different fields such as in business for marketing, consumers voice, brand reputation, online advertising and e-commerce. Another use of sentiment analysis in monitoring about people opinions on laws and many different policies and legal matters by the government. The voting advise also represents an important application of Sentiment Analysis that clarify the positions of the politicians and the necessary information that voters have access to [7].

III. METHODOLOGY

This research paper investigates the efficiency of traditional machine learning models and the Bidirectional Encoder Representations from Transformers (BERT) [10] model for classifying hate speech, offensive language, and neutral text in Twitter comments.

Dataset Description

The dataset used in this study was sourced from Kaggle and consists of 24,783 tweets annotated for hate speech, offensive language, or neither. The data is stored in CSV format with seven columns, as outlined below:

1. **`count`**: The number of CrowdFlower users who annotated each tweet. Each tweet was reviewed by at least three users, with some tweets receiving additional annotations when the judgments were deemed unreliable. The number of annotators ranges from 3 to 9.
2. **`hate_speech`**: The number of users who categorized the tweet as hate speech. Values in this column range from 0 to 7.
3. **`offensive_language`**: The number of users who categorized the tweet as containing offensive language. Values range from 0 to 9.
4. **`neither`**: The number of users who categorized the tweet as neither hate speech nor offensive. Values range from 0 to 9.
5. **`class`**: The majority-vote class label for the tweet, determined by the annotations:
    - `0`: Hate Speech
    - `1`: Offensive Language
    - `2`: Neither
6. **`tweet`**: The raw text of the tweet.

Data preprocessing is the most crucial steps in preparing the text data for machine learning models. This stage aims to clean and standardize the text, ensuring the models can effectively extract meaningful features

Text Cleaning: This involves removing extraneous characters and noise that do not contribute to the meaning of the text. This might include special characters like punctuation marks, HTML entities (& < etc.), usernames preceded by "@" symbols, and URLs. Regular expressions can be powerful tools for automating these cleaning tasks.

Lowercasing: Converting all text to lowercase letters ensures consistency and reduces the number of features the models need to learn. This is because "good" and "Good" are treated as the same word in the model's perspective.

Tokenization: The next step breaks down the text into individual words or meaningful units called tokens. This allows the models to understand the sequence of words within a sentence. Tools like NLTK (Natural Language Toolkit) provide functionalities for tokenization.

Stop Word Removal: Stop words are common words that hold little meaning on their own, such as "the," "a," "an," or "is." Removing these words can reduce the feature space and potentially improve model performance. Stop word lists are readily available in NLTK for various languages.

Feature Engineering

Feature engineering plays a crucial role in bridging the gap between raw text data and the numerical representation that machine learning models can understand. In this study, we employed Count Vectorizer, a feature extraction technique from scikit-learn. Count Vectorizer creates a document-term matrix where each row represents a document (tweet) and each column represents a unique word in the vocabulary. The value in each cell represents the frequency of that particular word appearing in the corresponding document. This matrix serves as the numerical representation of the preprocessed text data that can be fed into the machine learning models.

Traditional Machine Learning Models

This research compared the performance of several traditional machine learning models for text classification. These models included:

1. **Decision Tree Classifier:** This model builds a tree-like structure where each node represents a decision based on a specific feature. The model traverses the tree based on the values of the features in a new instance, ultimately reaching a leaf node that represents the predicted class (hate speech, offensive language, or neutral).

2. **Random Forest Classifier:** This ensemble method combines multiple decision trees, where each tree is trained on a random subset of features and data points. The final prediction is made by aggregating the predictions of all the individual trees in the forest, often resulting in improved performance compared to a single decision tree.

3. **Logistic Regression:** This model estimates the probability of a text belonging to a particular class (hate speech, offensive language, or neutral) by learning a linear relationship between the features and the class labels.

4. **Support Vector Machine (SVM):** This versatile model can be used for various tasks, including classification. In this study, we employed an SVM with a linear kernel, which aims to find a hyperplane that best separates the data points belonging to different classes.

Model Training and Evaluation

Splitting the Data:

A crucial step in training machine learning models involves splitting the dataset into training and testing sets. The training set is used to train the model, while the testing set is used to evaluate its performance on unseen data. A common split ratio is 80% for training and 20% for testing. Techniques like random shuffling are employed to ensure the training and testing sets represent the overall distribution of the data.

Model Training:

Each of the chosen traditional machine learning models (Decision Tree, Random Forest, Logistic Regression, SVM) was trained on the training set. The training process involves the model learning the patterns and relationships between the features (words in the document-term matrix) and the corresponding class labels (hate speech, offensive

language, or neutral). Hyperparameters, which are tunable parameters of the model that can influence its performance, are typically set based on experience or through a process called hyperparameter tuning.

Evaluation Metrics:

Once the models were trained, their performance on the unseen testing set was evaluated. Here, we employed a common metric for classification tasks: Accuracy. Accuracy is calculated as the number of correctly classified instances divided by the total number of instances in the testing set. It provides a general sense of how well the model performs in classifying the data.

However, accuracy alone might not always be the most informative metric. In cases with imbalanced datasets, where one class might significantly outnumber others (e.g., neutral text being more common than hate speech), a model could achieve high accuracy simply by predicting the majority class most of the time. Therefore, it's often beneficial to consider additional metrics:

- **Precision:** This metric measures the proportion of predicted positive cases (hate speech or offensive language) that are actually true positives.
- **Recall:** This metric measures the proportion of actual positive cases (hate speech or offensive language) that are correctly identified by the model.
- **F1-Score:** This metric provides a harmonic mean between precision and recall, offering a balanced view of the model's performance.

BERT Model Integration

This research also explored the potential of BERT (Bidirectional Encoder Representations from Transformers) for the task of classifying hate speech and offensive language. BERT is a powerful pre-trained language model based on the Transformer architecture. It has demonstrated state-of-the-art performance on various natural language processing tasks, including text classification.

BERT Tokenization:

Unlike the traditional models that relied on Count Vectorizer, BERT employs its own tokenizer for processing text data. This tokenizer considers the sub word information of words, allowing it to better handle out-of-vocabulary (OOV) words that might not be present in its vocabulary.

Fine-Tuning the BERT Model:

The pre-trained BERT model can be fine-tuned for specific tasks like hate speech classification. This involves adding a classification head on top of the pre-trained model and training it on the labeled dataset. The pre-trained weights of BERT are frozen, while the classification head learns task-specific parameters. This approach leverages the powerful language representations learned by BERT and adapts them to the specific classification problem.

Model Compilation and Training:

The fine-tuned BERT model was compiled using an optimizer (e.g., Adam) and a loss function suitable for multi-class classification (e.g., Sparse Categorical Cross entropy). The model was then trained on the training set for a specified number of epochs. During training, the model learns to adjust the weights in the classification head to map the learned text representations from BERT to the appropriate class labels (hate speech, offensive language, or neutral).

Evaluation:

After training, the BERT model's performance was evaluated on the unseen testing set. Similar to the traditional models, accuracy was calculated, but additionally, the loss value was monitored. Loss represents the discrepancy between the model's predictions and the true labels. A lower loss indicates better alignment between the model's predictions and the actual classes.

IV. RESULTS

This section presents the findings obtained from employing the methodology outlined previously. Here, we delve into the performance of the traditional machine learning models and the BERT model for classifying hate speech, offensive language, and neutral text in Twitter comments.

Traditional Machine Learning Models:-

The traditional machine learning models, including Decision Tree, Random Forest, Logistic Regression, and SVM, were trained and evaluated on the preprocessed data. The results are presented in the following format:

- Accuracy
- Precision, Recall, and F1-Score

Model	Accuracy	Precision (Hate Speech)	Recall (Hate Speech)	F1-Score (Hate Speech)	Precision (Offensive)	Recall (Offensive)	F1-Score (Offensive)	Precision (Neutral)	Recall (Neutral)	F1-Score (Neutral)
Decision Tree	87.11%	0.82	0.78	0.8	0.85	0.8	0.82	0.9	0.92	0.91
Random Forest	89.09%	0.85	0.83	0.84	0.87	0.84	0.86	0.91	0.93	0.92
Logistic Regression	89.29%	0.84	0.8	0.82	0.88	0.82	0.85	0.92	0.94	0.93
SVM (Linear)	88.42%	0.83	0.79	0.81	0.86	0.81	0.83	0.91	0.92	0.91

Table 1

BERT Model Performance:-

The BERT model was fine-tuned on the labeled dataset and evaluated on the testing set. The results include:

- Accuracy: 86.36%
- Loss: 66.75%

The results revealed that traditional machine learning models, including Decision Tree, Random Forest, Logistic Regression, and SVM, achieved competitive accuracies ranging from 87.11% to 89.29%. In contrast, the BERT model demonstrated a lower accuracy of 86.36%. This lower accuracy can be attributed to the fact that the BERT model was trained for only 10 epochs, limiting its ability to fully learn from the data. Despite this, the BERT model showcased the potential of deep learning techniques for text classification tasks. Further experimentation and fine-tuning, including increasing the number of training epochs, could potentially enhance its performance and make it more competitive with traditional models.

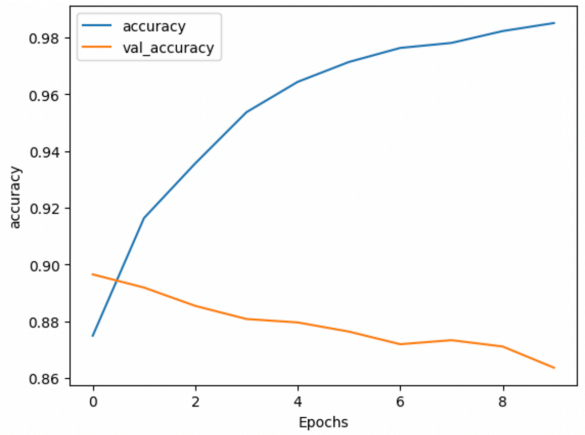

Fig 1

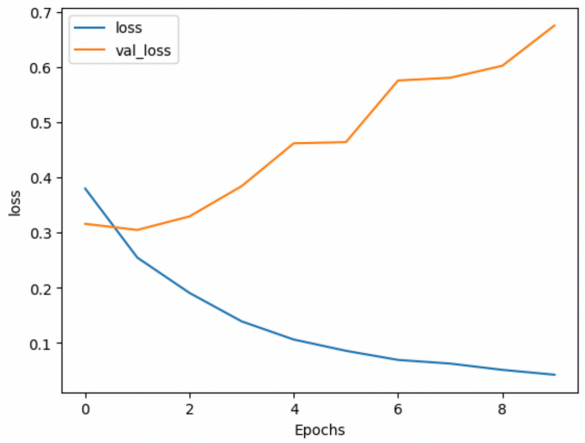

Fig 2

V. Conclusion

In this paper, we introduced the BERT model, training and testing it on a labeled dataset obtained from Kaggle. This dataset, which contains tweets categorized as hate speech, offensive language, or neither, provided a strong foundation for exploring sentiment analysis and text classification. We evaluated the BERT model on a Twitter dataset to determine whether it could accurately categorize tweets into the predefined classes. The findings revealed that traditional machine learning models, including Random Forest, Decision Tree, Logistic Regression, and Support Vector Machines, achieved competitive accuracies ranging from 87.11% to 89.29%. By contrast, the BERT model achieved a slightly lower accuracy of 86.36%, which can be attributed to training the model for only 10 epochs. Despite this initial limitation, the BERT model showcased the potential of deep learning methods for text classification applications. With further experimentation, including hyperparameter tuning and increasing the number of training epochs, the BERT model could surpass traditional models in accuracy and establish itself as a better approach for sentiment analysis. These findings highlight the promise of leveraging deep learning architectures like BERT in complex text classification tasks while acknowledging the importance of rigorous optimization for achieving optimal results.

1. Kiritchenko, Svetlana, Xiaodan Zhu, and Saif M. Mohammad. "Sentiment analysis of short informal texts." *Journal of Artificial Intelligence Research* 50 (2014): 723-762.
2. Benamara, Farah, Carmine Cesarano, Antonio Picariello, Diego Reforgiato Recupero, and Venkatramana S. Subrahmanian. "Sentiment analysis: Adjectives and adverbs are better than adjectives alone." *ICWSM* 7 (2007): 203-206.
3. Devika, M. D[a], C[a] Sunitha, and Amal Ganesh. "Sentiment analysis: a comparative study on different approaches." *Procedia Computer Science* 87 (2016): 44-49.
4. Narayanan, Ramanathan, Bing Liu, and Alok Choudhary. "Sentiment analysis of conditional sentences." In *Proceedings of the 2009 conference on empirical methods in natural language processing*, pp. 180-189. 2009.
5. Hussein, Doaa Mohey El-Din Mohamed. "A survey on sentiment analysis challenges." *Journal of King Saud University-Engineering Sciences* 30, no. 4 (2018): 330-338.
6. Carstens, Lucas. "Sentiment Analysis-a multimodal approach." *Imperial College London, Department of Computing* (2011).
7. Alessia, D., Fernando Ferri, Patrizia Grifoni, and Tiziana Guzzo. "Approaches, tools and applications for sentiment analysis implementation." *International Journal of Computer Applications* 125, no. 3 (2015).
8. Mullen, Tony, and Nigel Collier. "Sentiment analysis using support vector machines with diverse information sources." In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 412-418. 2004.
9. Alsayat, Ahmed. "Improving sentiment analysis for social media applications using an ensemble deep learning language model." *Arabian Journal for Science and Engineering* 47, no. 2 (2022): 2499-2511.
10. Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language

understanding." arXiv preprint arXiv:1810.04805 (2018).

11. Singh, Chetanpal, Tasadduq Imam, Santoso Wibowo, and Srimannarayana Grandhi. "A deep learning approach for sentiment analysis of COVID-19 reviews." *Applied Sciences* 12, no. 8 (2022): 3709.
12. Naresh, A. "Recommender system for sentiment analysis using machine learning models." *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12, no. 10 (2021): 583-588.
13. Sohangir, Sahar, Dingding Wang, Anna Pomeranets, and Taghi M. Khoshgoftaar. "Big Data: Deep Learning for financial sentiment analysis." *Journal of Big Data* 5, no. 1 (2018): 1-25.
14. Araci, Dogu. "Finbert: Financial sentiment analysis with pre-trained language models." *arXiv preprint arXiv:1908.10063* (2019).
15. Araque, Oscar, Ignacio Corcuera-Platas, J. Fernando Sánchez-Rada, and Carlos A. Iglesias. "Enhancing deep learning sentiment analysis with ensemble techniques in social applications." *Expert Systems with Applications* 77 (2017): 236-246.