Abusive Word Detection And Removal

*

***Abstract*—The rise in social media has brought attention to cases of abusive language online and has made it necessary to address this abnormal or disruptive behaviour by using effective methods. This study presents an automated model for detecting abusive language and hate speech-related text in the social post. The training for the model is performed using a publicly available dataset. Both machine learning (ML), deep learning (DL), and transformer-based models (TL) were trained and tested, and it was found the deep learning-based Long Short-Term Memory (LSTM) performed better compared to the ML and TL models. The model performance was validated on the state-of-the-art dataset, and it found the proposed model outperformed the existing models. The proposed automated deep learning-based model can detect and replace abusive words with asterisks (*) symbol and hate speech on social media in real-time, which can help make conversation respectful.**

I. Introduction

Nowadays, a lot of hate speech is spread in the world through abusive words on the internet. Abusive words are offensive or harmful language used to insult, or hurt someone emotionally. Abusive Word Detection and Removal refers to the process of identifying the words promoting hate speech from social media platforms such as Instagram, X and removing the words to avoid negative impact on the users. One of the many social media platforms where these things occur is Twitter or X. It is a platform where people can share their feelings, thoughts in the form of messages which are posted globally on the web also known as tweets. Over 330 million active users are present and allows communication between the users with the help of hashtag and retweets on diverse topics and also allows users to keep updated with the trend and news across the world.

As we know there are two sides of any social media platform: one is positive and on the contrary the other one is negative. The negative impact is hate speech and abusive language which can target individuals on the basis of race, gender, religion and other characteristics.It can enhance the case of cyberbullying, which can lead to anxiety, depression and other mental health issues. Abusive language can have serious psychological and emotional distress on the targeted individual. By removing this, we can make conversations on social media healthy and respectful and can avoid violence and riots which can be caused due to hate speech on an individual or community. By the removal of abusive language we can create an environment where people of each and every community across the world are invited and can have respectful and safe interactions online and by this the reputation of the platforms will increase which will attract more users to the platform.

There is one solution which matches the input with the list of offensive words to detect the abusive word.This is very simple and can miss a lot of offensive and hate speech. ADD REFERENCE HERE Deep learning techniques are used to classify hate or not hate speech into two different categories.These include the application of BERT, LSTM (Long Short Term Memory) with different text embedding, and Convolutional Neural Network(CNN). Researchers have used n-grams and pre-trained embeddings features to identify hate speech in various orders using SVM, bi-LSTM, and bi-LSTM with attention, as well as CNN and RNN algorithms Overall, these studies highlight the diversity of approaches and techniques employed to detect and classify hate speech in social networks, ranging from deep learning models to conventional machine learning algorithms.

This research aim to reduce the negative impact such as hate speech, offensive sentences by using automated approach to detect the abusive or offensive word in the social post and will replace the offensive, hate or abusive word with asterisks(*) with the help of a dynamic list of abusive list dictionary.

Creation of Abusive List: One of big challenge in current time for any program to understand the context of the word. For example, a user posts ”I hhate socal ntwrk pltform but love to post my achievements there.” The post containing many word whose meanings are not available in English dictionary (hhate, socal, ntwrk, pltform) however for the end user it is a clear message and the context is also clear. The similar issues exist in case of abusive words. A user may write abusive words by including or excluding a character, or by placing an additional space between two character of the abusive word. The context of such posts are clear for the end user, but the program may fail to identify it. Therefore, a separate list of abusive words created in which all possible forms of the abusive words added. The key contribution of this research includes the followings:

1) Built a two-layered framework for identifying and replacing hate, offensive and abusive words in the social posts.
2) Achieved remarkable 93% of accuracy for identifying a social post is hate or non-hate category.
3) Developed a huge list of hate, offensive or abusive English and Hindi words, including their multiple variations.
4) The proposed framework to replace hate, offensive or abusive English and Hindi words with asterisks (*).

The rest of the article is organized as follows: Section 2 discusses the recent research on hate speech detection. Section 3 discuss the proposed framework for handling abusive words. In Section 4, the experimental outcomes discussed, Finally Section 5 conclude the work with limitations and future scope.

II. Related Works

Add 20-30 research papers on hate, offensive and abusive words prediction with their findings and limitations.

III. Proposed System

Deep learning methods refer to deep neural network-based hate speech detectors. The input data to these neural networks can be in any form of feature encoding, including traditional methods like TF-IDF and recently emerged word embedding or pre-training methods. Some popular neural network techniques are Convolutional Neural Network(CNN), LSTM (Long Short Term Memory), and bidirectional LSTM(Bi-LSTM).

1) **Word Embedding**- It is a Natural Language Processing (NLP) technique that conveys the semantic meaning of a word.It provides a useful numerical description of the term based on its context. The words are represented by dense vectors that can be used in estimating the similarities between the words.It is used in text classification, and it can be used to classify similar hate speech words after the training of the model.Word Embedding models are Word2Vec,GLOVE etc.
2) **Transformer-based Model**-It uses transformer based embedding technique small BERT, BERT etc in combination with LSTM, Bi-LSTM to enhance the performance of the model for detection of hate speech in tweets.It gives good performance on various datasets of languages such as Hindi, English, Spanish and Arabic.

The proposed system would involve the creation of the machine learning model that can recognize and categorize the tweet as abusive or non-abusive in tweets. The system will take text based tweet as input and output the abusive words marked as asterisk(*) if the sentence is abusive, which may be hate speech or offensive.

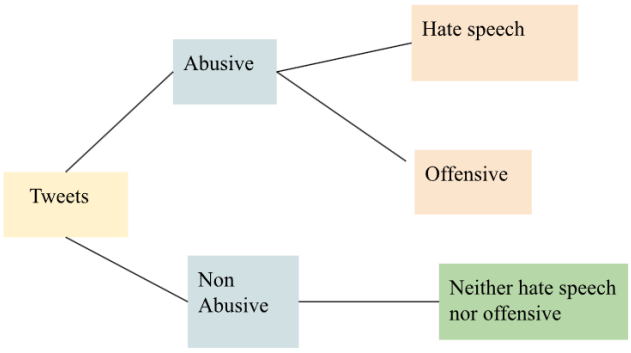

Fig. 1. An image of a dataset classes

Data Collection: This research usage a dataset available of Kaggle platform. The dataset consists of around, 25000 labelled tweets for the English language consisting of 3 classes: hate speech, offensive language, or neither hate speech nor offensive language. The description of each column of dataset is given below:

- **Count**-It defines the total number of annotations in a tweet (integer).
- **Hate speech**-Number of annotations classifying a tweet as hate speech if this is the greatest among all classes then it belongs to hate speech.
- **Offensive speech**–Number of annotations classifying a tweet as offensive speech if this is the greatest among all classes, then it belongs to offensive speech.
- **Neither offensive nor hate speech**-Number of annotations classifying a tweet as neither offensive nor hate speech if this is the greatest among all classes, then it belongs to this class.
- **Class**-It determines in which class the tweet falls, whether abusive(hate or offensive) or non-abusive.The class is a number assigned to each column 0,1,2 for hate speech, offensive speech and neither offensive nor hate speech respectively.
- **Tweet**-It consists of labelled tweets for the training and testing of the model.

Feature Engineering: It involves selecting and extracting the useful information from the text data to train the model more efficiently and accurately.The method includes counting the word frequencies and adding a new column, which is adding values to the dataset.

Model Training:It can use various machine learning models such as SVM (support vector machine), Logistic regression, deep learning techniques, etc.It depends on the topic and research on topic which model is required.Model will learn pattern and association to classify words on the basis of abusive and non-abusive.

Model Testing : It is a metric which is used to evaluate the performance of the model which includes accuracy,F1-score,recall and precision.Accuracy cannot be the only factor to check performance, we have to see all the factors.

Data Preprocessing-The quality of data and useful information that can be derived from the dataset will tell how efficiently the model will learn and produce effective output. To improve the efficiency of the model we have done text preprocessing such as emoji removal, stopword removal, change username to user, url removal and unnecessary symbols in the dataset. Regex has been used for each of the text preprocessing methods.

Feature Extraction-Feature extraction often involves reducing the dimensionality of the data while preserving its important characteristics. This helps in reducing computational complexity and removal of unnecessary features present in the dataset. We will remove columns or features from the dataset which consist of a lot of missing values and the ones who have no role in training the model.

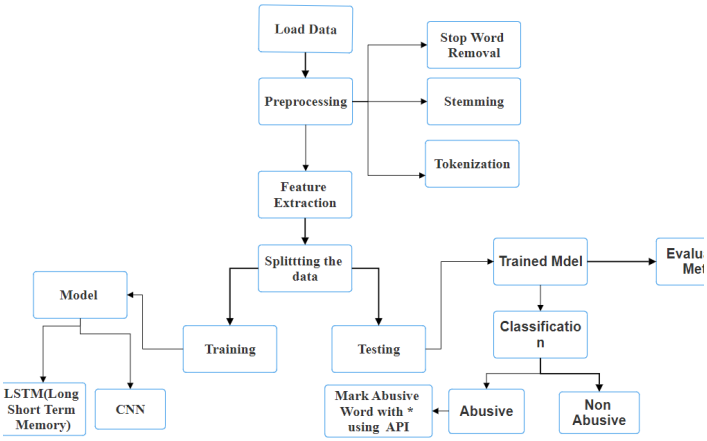

Fig. 2. Methodology

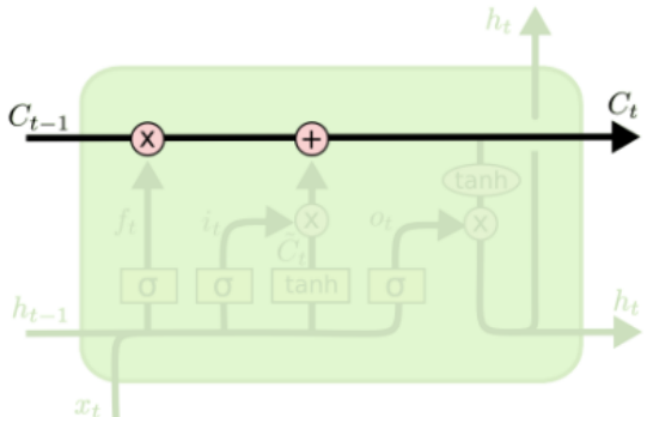

Fig. 3. CellState in LSTM

Splitting the Data-The partitioning of the dataset into training and testing sets is essential for development of the model.The splitting of data is done into training 80 percent and for testing 20 percent. 80 percent of data is used for training purposes to learn patterns, features of data and the other 20 percent is used to test the trained data. The tweets for testing purposes were 4957 whereas for training purposes it was 19826. This ratio is ideal to avoid underfitting and overfitting in the model, and detail about underfitting and overfitting is given below:-

- **Underfitting**:It happens when the model is unable to learn patterns efficiently and perform poorly on training data and testing data. This could happen if training data is smaller in size as required.
- **Overfitting**: It happens when the model learns unnecessary patterns from training data instead of learning underlying structure.

The dataset has been splitted into Training and Testing part:- The machine learning algorithm that suits the most has been selected and is used in making this model.The LSTM algorithm using deep learning is used to train the model to classify the tweets into different classes. Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies as RNN are unable to learn long term context so we use LSTM for long sentences to handle context switching.

The repeating module image consist of following parts:-

1) **Cell state (Fig.3)**:It is the horizontal line running through the top of the diagram. It runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow along it unchanged.This is responsible for the addition and removal of information to the cell state based on the context.
2) **Forget Gate Layer (Fig.4)**:It is responsible for deciding what information has to be removed from the cell state.Suppose in the present subject context has been changed the sigmoid output values will be nearly 0 which on doing point operation becomes 0 and the past information will be forgotten whereas if the subject has not changed then the sigmoid function will have the value close to 1 passing the past information from the cell state.

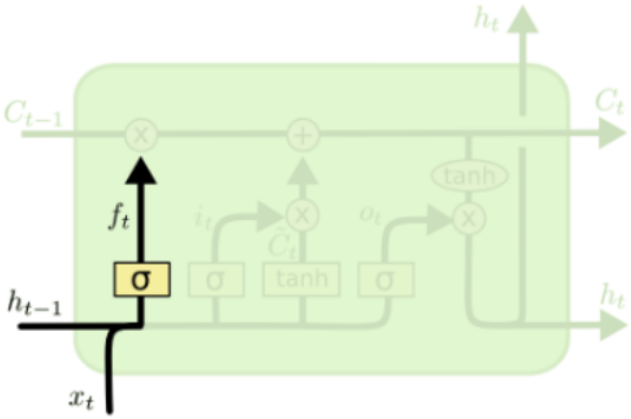

Fig. 4. Forget Layer in LSTM

3) **Input gate layer and tanh layer(Fig.5)**:The first sigmoid layer in the below image is responsible for what values have to be updated, and a tanh layer creates a vector of new candidate values that could be added to the state after combining these two to create an update to state. After passing from these layers, the add point operation will happen to add new information to the cell state.

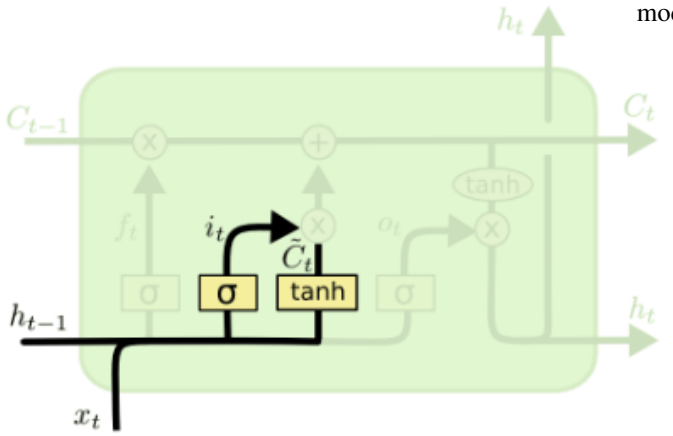

Fig. 5. Input layer in LSTM

4) **Output layer (Fig.6)**:Finally, we need to decide what we're going to output. This output will be based on our cell state, but will be a filtered version. First, we run a sigmoid layer, which decides what parts of the cell state we're going to output. Then, we put the cell state through tanh and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.

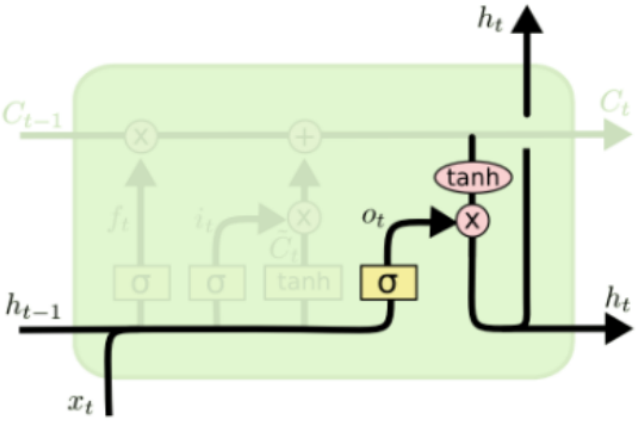

Fig. 6. Output Layer in LSTM

Pseudocode for LSTM: Define output_dim = 200 Define model:

Initialize Sequential model
Add Embedding layer:
  Input: vocab_size
  Output: output_dim
  Input length: max_length
Add LSTM layer:
  Units: 64
  Dropout: 0.3
  Recurrent dropout: 0.3
Add Dropout layer:
  Rate: 0.5
Add Dense layer:
  Units: 128
  Activation: ReLU
Add Dropout layer:
  Rate: 0.5
Add Dense layer:
  Units: 3
  Activation: Softmax

Compile model:

Optimizer: Adam
Loss: Categorical crossentropy
Metrics: Accuracy, F1 score, Precision, Recall

Testing-We have used 20 percent testing data for the testing of the model we have trained using the LSTM method.After this part we have calculated the accuracy and other parameters for the model.

Evaluation Metrics: It is a metric which is used to evaluate the performance of the model which includes accuracy,F1-score,recall and precision.Accuracy cannot be the only factor to check performance, we have to see all the factors.Classification Report is used to get all the defined parameter in the code.

A. Technologies and Tools Used

The tools and technologies are used to preprocess, analyze, model, and visualize data. They also provide an environment for programming, experimentation, and testing. The technologies that are used for preparing the model is given below:-

1) **Python**: Python is a well-liked programming language for artificial intelligence and machine learning. It provides a range of frameworks and libraries for deep learning, natural language processing, and data analysis.

2) **TensorFlow**: Google created the open-source machine learning library TensorFlow. It offers resources for configuring and training neural networks, such as the well-known Keras API
3) **Keras**: Keras is a Python-based high-level neural network API. It offers a straightforward neural network construction and training interface.
4) **Google Colab**: You can create and share documents with live code, mathematics, graphics, and narrative text with Google Colab, an open-source web tool.
5) **Pandas**: Pandas is a Python package for analysis and data
6) **NLTK**:NLTK is a well-known Python toolkit for working with human language data, covering various text processing tasks like tokenization, stemming, lemmatization, part-of-speech tagging, and more.
7) **Scikit**: Python's scikit-learn (sklearn), a popular machine learning library. Tools for data preparation, feature extraction, model selection, and evaluation are available through scikit-learn.
8) **API Ninja**-It is a profanity filter which is used to mark abusive word as asterisk(*) it has been used to have clean tweets to be posted.

IV. Result

The result has been shown with the help of evaluation metric parameters such as accuracy, precision, recall, f1-score etc.We have also created confusion matrix for 3 class for better observation.

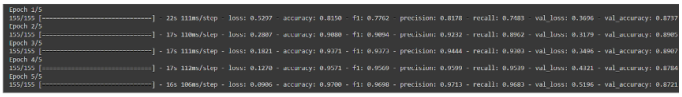

Fig. 7. The above image shows the accuracy after each epoch at the time of training of the neural network.

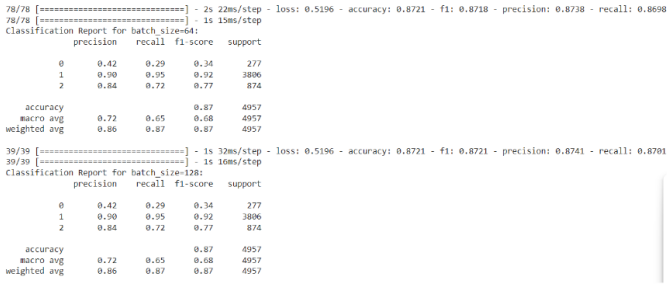

Fig. 8. The above image shows the classification report for batch-size of 64 and 128..

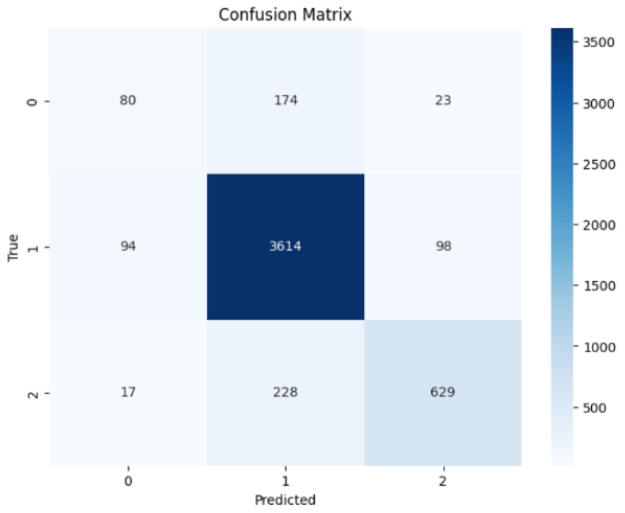

Fig. 9. This is the confusion matrix for 3 classes 0,1,2 is for hate speech, offensive speech and neither hate nor offensive speech respectively which is classified by the model..

Fig. 10. The image shows the final result of marking the abusive word with an asterisk for the entered text.

V. Conclusion

The project shows a methodology to classify tweets into classes and, based on if a tweet contains an abusive word, mark it with an asterisk(*). Classification of tweets is done using the LSTM model, and marking of abusive words is done with the help of a profanity API. The methodology initiates with the preprocessing of raw tweet dataset taken from kaggle to ensure its accuracy then feature extraction has been done to remove the unnecessary features from the dataset then we have used LSTM layer to train the data and further tested it with the testing data. When the new text is passed into the trained model, it can be classified into the class it belongs to according to the model.If the entered tweet or sentence belongs to the hate speech or offensive class, then we will mark the abusive word as asterisks(*).We are using a Ninja API which is a profanity API to identify abusive words. We can conclude that if the tweet belongs to hate speech or offensive class it will be converted into a clean tweet which will create a healthy environment for the user and may attract more users to the platform.

VI. References

- **Md Saroar Jahan and Mourad Oussalah**:A systematic review of hate speech automatic detection using natural language processing.
- **Mohmd Ahmed(2017)**:Hate Speech Detection Using a Convolution-LSTM Based Deep Neural Network.
- **Ziqi Zhang Blog** :Understanding LSTM Network.