A SYSTEM FOR DETECTING ABUSIVE CONTENTS AGAINST LGBT COMMUNITY USING DEEP LEARNING BASED TRANSFORMER MODELS

Deepalakshmi Manikandan[1] , Malliga Subramanian[1], Kogilavani ShanmugaVadivel[1],

[1]*Department of Computer Science and Engineering, Kongu Engineering College, Erode, Tamil Nadu, India*

Abstract

The dissemination of harmful and unfriendly content has exponentially increased in the modern world as a result of social media. Many in the community of natural language processing have recently become interested in hate speech, inciting language, and abusive language. In this article, we suggest using transformer-based model methodologies like BERT and XLMROBERTa models to identify Non-Anti LGBT content (NALC), transphobic and homophobic insults directed at transgender people. In this work, English language dataset with 990 comments is tested without label. Based on the experimental results, the XLM-RoBERTa achieved superior results with respect to the precision, recall and f-measure than BERT model. Also, with respect to the accuracy the BERT gives 91% and XLM-RoBERTa gives 93%. With an accuracy of 93% in the XLMROBERTa exceeds the BERT model.

Keywords
Hate Speech, Abusive language, Transformer, BERT, XLMROBERTa

1. Introduction

With the increase in the popularity of social media sites like Twitter, Facebook, YouTube, Instagram and WhatsApp etc.,, there is a tremendous growth in exchanging messages around the world. As a result of people communicating more frequently on social media sites due to their growing popularity, this encourages derogatory remarks about them and causes people to cease using these platforms for self-expression. More than 3.6 billion people have a social identity in online platforms. The online social media platform is used as a worldwide discussion forum to share their opinions between the persons, groups or different communities. Since it is the global connection between different cultures and peoples the social media is used to spread cybercrimes and cyberhate conversations among the peoples or communities. One of cyberhate talk is hate speech where a particular language is used to express hatred content toward a minority community to insult or humiliate the group of peoples or individuals. Due to the range of linguistic patterns, the variation in language usage around the world also presents a significant difficulty. The basic goal of social media is to connect more individuals in order to facilitate the exercise of their First Amendment right to free speech. However, other groups frequently abuse these platforms to promote insulting and hateful messages that are directed at specific people or groups in general [1].

Facebook and Twitter are more widely used for messaging, both for positive and bad objectives. Nowadays, it's usual to spread misinformation, promote hate speech, and make inflammatory posts on social media sites like Twitter and YouTube. In contrast to Twitter, which is saturated with hate speech, many comments on YouTube videos fall into the offensive category. Offensive messages aren't traced

Forum for Information Retrieval Evaluation, December , 2022, India
EMAIL:latchumimani@gmail.com (Deepalakshmi Manikandan) mallinishanth72@gmail.com(Malliga Subramanian); kogilavani.sv@gmail.com(Kogilavani Shanmuga Vadivel);
ORCID:

because the corporation doesn't do much person level moderation [2]. Posting abusive words against a group or common people has a significant negative effect on those targeted by the attacks; the victims experience sadness, stress, and other mental health issues, and in some circumstances, the attacks may be so severe that the victims end up committing suicide.

With the deep learning-based transformer technique, the objectionable post was found. The English language dataset was used for the model construction in the majority of the existing research on offensive post detection. However, at the moment, people choose writing posts in a combination of languages, such as English and Hindi, English and Tamil, etc. Indians, who speak a wide variety of regional languages, have gotten used to expressing themselves on social media through a combination of their native tongue and English. As a result, the models created using the monolingual dataset are not appropriate for the current effective identification of offensive social posts.

The transformer-based models BERT and XLMROBERTa are utilized in this paper to address this issue. Finding an effective method for hate speech on homophobia and transphobia in speeches about transgender people is the objective. The researchers frequently employ the two methodologies known as ML and DL. The succeeding sections go through the work done using different algorithms.

The remainder of this paper is organized as follows: Section 2 provides the literature review with respect to deep learning models on topic of transphobic and homophobic insults. Section 3 provides the system architecture and its explanation whereas section 4 presents the results achieved and discussions on the performance. Finally, section 5 concludes the paper with future directions.

2. Literature Survey

With the aim of differentiating between the hate speech and offensive language presence in the text [3] explored the major issues of the hate speech detection on particular with Twitter. They demonstrated how the use of offensive words and language does not always indicate the hate speech content. The researchers assessed their own dataset for detection of the hate speech in the content using the logistic regression. However, the classifier encountered challenges in the hate speech detection criterion and given 0.90 F1 score and also 40% tweets were miss classified as hate speech.

Authors from [4] have designed a framework for detecting hate speech using recurrent neural networks and for distinguishing between sexism and racism. The proposed new algorithm for the hate speech detection uses the recurrent neural networks and classifies the messages as racism and sexism.

Developing the model with a code-mixed dataset is more challenging than the English dataset as the tokenization process of the dataset is different. The available resources for the Dravidian code-mixed dataset are insufficient, making this problem more challenging. The available corpus to train the model consists of limited samples, and hence there is a high chance that the words present in the test sample may fall under out of vocabulary problems. All these issues lead to lower prediction accuracies. To address this issue, transformer-based models like mBERT, distilBERT, xlm-RoBERTa, and MuRIL are used, which is pretrained on a large corpus of multiple Indian languages. The current study focuses on offensive post detection in code-mixed languages of Tanglish (Tamil–English mixed) and Manglish (Malayalam–English mixed) with the dataset provided in the HASOC-Dravidian-CodeMix-FIRE2021 challenge. An overview of the dataset can be found here. [5]

Beddiar et. al [6] mentioned the automatic identification of objectionable content using deep learning algorithms appears to yield promising results and also deep learning techniques require a substantial amount of labelled data which is of high-quality but it is generally lacking in the real time application scenarios. Further the authors mentioned that, the LSTM and CNN models performed admirably in terms of hate speech classification, with high accuracy, F1 score, recall, and precision. Recall and precision could give efficient and accurate measurements on the classifier's performance for imbalanced class datasets.

Agarwal et. al [2] mentioned, detecting the hate speech on the social media networks automatically is a critical job which has eluded the researcher’s despite of the numerous attempts in developing the training models. Most of the standard techniques are difficult to process and classify the data from the social media posts since decrease in performance during the crossdataset evaluation. Also, the impact of limiting the maximum number of tweets per user causes performance concerns, according to the study.

H. S. Alatawiet.al [7] highlights the problem in the white supremacist hate speech detection with NLP and deep learning techniques in twitter by detecting the hate speech in timely manner. The authors have presented a novel strategy for the white supremacist hate speech by using BERT and BiLSTM algorithm. Rodriguezet.al [8] presented the issues which related with the detection of hate speech in Facebook comments. The proposed new framework called FADOHS only deals with identifying the negative comments and hate speech from the Facebook. Watanabeet.al (2018) presented the hate speech detection by the unigram approach and for finding the feature pattern by the two classification methods and compared the accuracy between the models. Also, gives the study of new approaches that automatically detects the hate speech by using the unigrams and the pattern evaluation techniques and the classification by using machine learning algorithms.

Le-Hong et.al [9] highlighted the diacritics generation is a difficult problem in text processing since it requires the creation of diacritic markings for non-accented text. With an everincreasing amount of informal text without accents such as short text messages, emails or blog posts on social media, a software system which is capable of generating diacritic marks accurately is very useful and necessary in many situations. Their proposed model performs well in the good category of messages on the test set but gives poor performance in the offensive and hate categories, with the F1 scores of 39% and 54% respectively.

Karayiğit et.al [10] highlighted the issues where the abusive photos and comments can be demoralizing and hazardous for persons who share in online. It's tough and time-consuming to write a remark and filter out languages other than English. The authors mentioned with main issues as there has been no investigation of the existence of a dataset including offensive words Turkish terms. The oversampling method is used for generating the dataset and improved the outcomes in terms of performance of the feature selection, embedding, and the classification algorithms. Also, the Support Vector Machine (SVM) classifier outperforms all others models in CNN. With comparison to the CNN model the oversampling method performs good in terms of precision.

The ALBERT model's architecture was developed by Lan et.al [10] with the lighter version of the transformer-based BERT model which improves the BERT model in many ways like factorization of the embedding parameters and the cross-layer parameters across layers; both techniques aim to increase parameter efficiency and function regularization. Additionally, ALBERT models the inter-sentence coherence by substituting the sentence-order prediction loss for the next-sentence-prediction loss that was used to train BERT. Therefore, it has been demonstrated that ALBERT performs better than BERT in few datasets on a number of multisentences encoding tasks.

Francimaria R.S et.al. [12] suggested an ensemble learning approach based on the various feature spaces for reducing the unintentional gender bias in the context of detecting the hate speech which is present on the online social media. The model combines fundamental classifiers, where each classifier is trained with the various feature representation. The feature extraction technique performs an abstraction of the data in the unique way and can produce the better classification performance. As a result, even if one technique of the feature extraction fails it leads the inconsistent data samples the system which can perform well because it takes other feature characteristics. By employing this bias-sensitive terms and a replacement technique the authors reduce the gender bias in the datasets and given better results.

Pradeepkumar roy et.al [13] proposed a deep ensemble-based framework which consists of deep learning and the transformer-based models for detecting the offensive messages, posts and blogs written in the Tamil, Malayalam language on the online social platform. Also, their proposed model is initially trained with the code-mixed language on Tamil and Malayalam datasets and testing is carried out with the similar way without class. The results are experimented and tested with the traditional machine learning techniques and the advanced deep learning frameworks and the proposed deep ensemble framework outperformed among all of them.

The authors of [14][15] mentioned the work focuses on analyzing the hate speech in the Hindi-English language. This method explores the transformation-based techniques to extract the precise representation of the text. The authors developed Map only Hindhi (MoH) algorithms which means Love in Hindi which consists the language identification Roman to Hindi transliteration with help of fine-tuned Multilingual Bert and the MuriL language. The experiment is conducted with three datasets and evaluated the proposed system with Precision, Recall and F1 measure metrics. Also, this work covers detailed discussing and the analysis of the errors with respect to variety of texts which is presenting online social media.

To encourage NLP research, various shared tasks have been organized. Premjith et. al. [16] described the findings of the shared task on multimodal sentiment analysis, submitted for "The Second Workshop on Speech and Language Technologies for Dravidian Languages" and this work identifies sentiment from video.

3. Materials and Methods

3.1 Dataset Description

Code-mixed dataset of English comments gathered from YouTube media used in this work. The corpora's average sentence length is one, however the comment has multiple sentences. Each comment is annotated at the level of the comment. Problems with the class imbalance that are based on actual world events are also included in this dataset. With the class labels Non-Anti LGBT content (NALC), Homophobic, transphobic, the dataset consists of 3164 English samples for training and 991 English samples for testing. The training set contains 3001 Non-Anti LGBT content samples, 157 Homophobic samples, 6 Transphobic samples and the details are presented in Table 1. Table 2 provides examples of training samples and for testing purpose we have used test data of 990 comments without label.

Table 1. Number of comments in each label

Label	No. of Comments
Non-Anti LGBT content (NALC)	3001
Homophobic	157
Transphobic	6

Table 2. Training Samples of the Dataset

Samples	Texts	Label
Sample [1]	I support her very smart ponnu	*Non-Anti LGBT content*
Sample [2]	Please upload part 2 soon	*Non-Anti LGBT content*
Sample [92]	Hi friend call me	*Non-Anti LGBT content*
Sample [86]	Give your phone number sir	*Non-Anti LGBT content*
Sample [112]	They harass everyone in the bus and do this for living	*Homophobic*
Sample [1039]	Magalakshmi Mukunthan Ella transgalayum konnudalaam. Easiest way is to just stab them in the streets. Or We can poison their water supply…	*Transphobic*

3. 2 Proposed Transformer Models

In this section the overall architecture of the proposed system and its explanation is presented. The architecture is depicted in Figure 1. The workflow consists of preprocessing phase, development of transformer-based model which includes BERT and XLM-RoBERTA model, test prediction and results and are shown in Figure 1. The text dataset is given as input to the preprocessing stage and the initial preprocessing is carried in this stage where processed data is given as input to the transformer-based BERT and XLM-Roberta model for training. After training the model the test set is given to the system with tuned parameters. Then, the results have been obtained and compared among those two models.

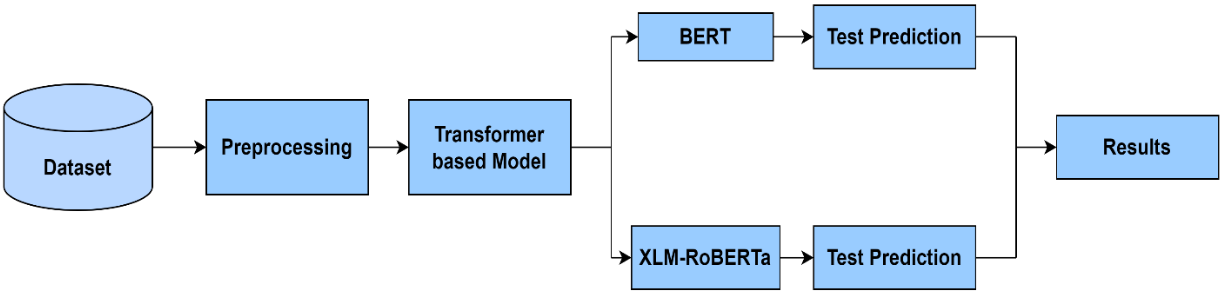

Figure.1. Proposed System Architecture

3.3 Pre-processing

The preprocessing module process the input text and removes noises in the text. Text preprocessing is a method to clean the text data and make it ready to feed data to the model. Text data contains noise in various forms like emotions, punctuation, text in a different case. In general, in Human Language, there are different ways to say the same thing, and this is only the main problem have to deal with because machines will not understand words, they need numbers so we need to convert text to numbers in an efficient manner.

Tokenization is the process of cutting a statement up into smaller terms. The Word Piece Tokenizer is used by the BERT. RoBERTa uses a byte-level BPE as a tokenizer and shares the same architecture as BERT. Tokenization is a method to segregate a particular text into small chunks or tokens. Here the tokens or chunks can be anything from words to characters, even subwords. Tokenization is divided into 3 major types namely Word Tokenization, Character Tokenization, Subword tokenization. Social media posts include a lot of distracting text that isn't thought to be a valuable attribute for classification. To prepare it for machine learning studies, we take the following actions to eliminate the noise, Stop Word Removal: Stop words in the input comments, such as formatting tags, pronouns, numbers, and prepositions, are eliminated. Also padding is carried out for processing of padding out sentences that aren't the right length. The vector representations are kept in the look-up table by embedding-it. The process of augmentation is only done in XlmRoBERTa model and not in BERT.

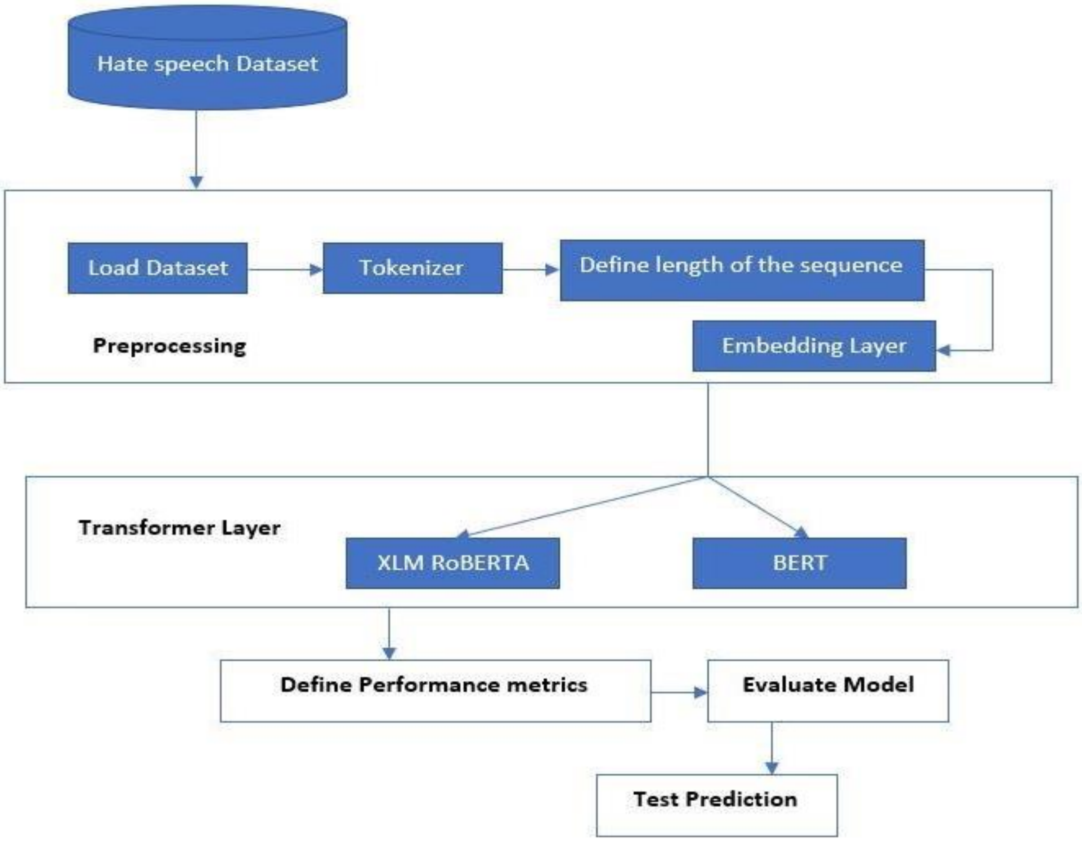

Figure. 2. Overall Workflow of the Proposed System

3.4 BERT Model

A transformer-based machine learning method called bidirectional Encoder Representations from Transformers (BERT) uses a model that has already been trained for natural language processing (NLP). At its core, BERT is a transformer language model with self-attention heads and a variable number of encoder layers. The architecture and the first transformer implementation are "nearly identical." Language modelling (15% of tokens were hidden, and BERT was trained to infer them from context) and next sentence prediction were the two tasks that BERT had been pretrained on (BERT was trained to predict if a chosen next sentence was probable or not given the first sentence). BERT gains knowledge of word contextual embeddings as a result of training. BERT can be fine-tuned with fewer resources on smaller datasets after pretraining, which requires expensive computational resources, to maximize its performance on certain tasks.

In this work, the BERT implementation comprises of 12 transformer blocks with 12 self-attention heads. The input for the BERT model is the sequence of lengths with maximum of 512. Also, the BERT embedding layer and its submodules are trained with the large corpus Wikipedia and other similar sources. The objectives of the BERT model are (a) Masked word prediction where the 15% of the text are masked and the remaining are fed into the transformer encoder. (b) Next word prediction for predicting the next word of the particular sentence. For understanding the relationship between the sentences, the BERT model will take hold of two different sentences for the input.

3.5 XLMRoBERTa Model

A scaled cross-lingual sentence encoder is called XLM-R (XLM-RoBERTa, Unsupervised Cross-lingual Representation Learning at Scale). It is trained using 2.5T of filtered Common Crawl data from 100 different languages. XLM-R performs at the cutting edge on numerous cross-lingual benchmarks. Two variations of the XLM-R transformer language model's architecture, known as XLM-RoBERTa-Base and XLM-RoBERTa-Large, were created using a distinct set of building blocks. At first, XLM-RoBERTa-Base was given a parameter count of about 270M, 12 Transformer layers, and 768 hidden units (in the context of Recurrent Neural Networks). The XLM-RoBERTa-Large version, in comparison, features an enlarged architecture tuned on 550M parameters, 24 Transformer layers, and 1024 hidden units. With 250K tokens for the Base and Large versions, the vocabulary is substantially larger than BERT's single token.

3.6 Embedding Layer

A major issue with social media texts is the use of words that are outside of their vocabulary. Social media users frequently purposefully obfuscate terms by using short words, acronyms, and misspelt words in order to avoid automatic inspection. Such words are not represented in the pretrained word embedding model, losing the morphological information. We make use of skip-gram model, which depicts each word as a collection of character n-grams. Each character has a vector value assigned to it; the total of these vector values is word embedding. Character embedding has a dimension of 300.

3.7 Transformer Layer

BERT consists of 12 transformer blocks and 12 self-attention heads. These are the processes in the Transformer layer. The BERT model takes a sequence with a maximum length of 512 as its input. It has an embedding layer where the words are stored in a lookup table and embedded into vector representations. Without the use of convolutional layers, the transformer is totally constructed via self-attention mechanism processes. The bare XLM-RoBERTa Model transformer outputs raw hidden-states without any specific head on top.

3.8 Masking and Next Word Prediction

The decoder layer is absent from BERT. The two cutting-edge techniques, "masking" and "next sentence prediction," are used while taking into account the encoder's output. When processing data, sequence-processing layers can be instructed to skip over specific timesteps in input by using masking. The method of determining the sentence's next word is known as next-word prediction. Following are the steps in the next word prediction: Start by breaking the sentence up into words. Next, choose the last word of the phrase, then determine its likelihood by consulting the vocabulary (dataset), and finally choose the following word.

4. RESULTS AND DISCUSSION

This section presents the performance metrics and comparisons of the implemented models BERT and XLM-RoBERTA.

4.1 Performance Evaluation

4.1.1 Evaluation Metrics

The effectiveness of the proposed system is validated using accuracy and other metrics namely precision, recall and f-measure are also used to demonstrate the effectiveness of the classifiers. These performance metrics are calculated based on the True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN). The formulae for computing classification accuracy, precision, recall and f-measure are given in Equations (4.1), (4.2), (4.3) and (4.4) respectively.

- **Accuracy:**

Accuracy is the proportion of true results among the total number of cases examined

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

- **Precision:**

  Precision, also called Positive predictive value. The ratio of correct positive predictions to the total predicted positives.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

- **Recall:**

  Recall, also called Sensitivity, Probability of Detection, and True Positive Rate. The ratio of correct positive predictions to the total positives examples.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.3}$$

- **F-measure:**

  The F-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall.

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision} \tag{4.4}$$

where, True Positive (TP), True Negative (TN), False Positive (FP), and False Negative indices are used to calculate these measurements (FN). TP is the total number of texts that were correctly categorized into each class. FP indicates the number of texts that were incorrectly categorized in a class other than the correct class. FN is the number of texts that were incorrectly assigned to the appropriate class. The number of texts successfully classified in a class other than the correct class is known as the TN.

4.1.2 Performance evaluation of BERT Model

The performance of the BERT model with respect to the epochs is shown in the Figure 3. The number of epoch for training is found by the trial and error strategy and we find after 7 epoch, we achieved good performance. Hence, we use 7 epochs for validation and testing.

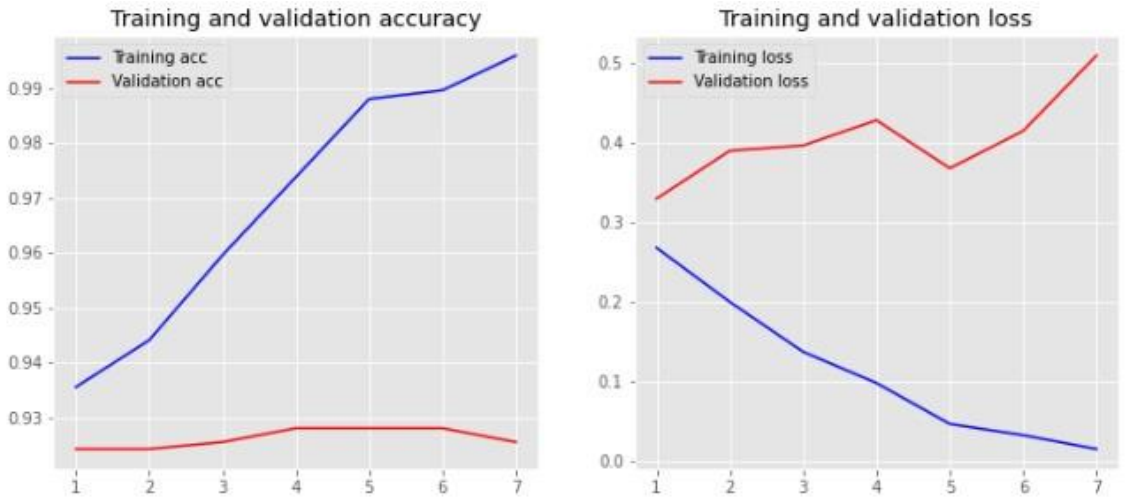

Figure 3. Accuracy and loss metrics of BERT algorithms

The left graph in the Figure 3 shows the training and validation accuracy of the BERT model for 7 epochs. It can be seen that the training accuracy gradually increases with respect to the increase of epochs whereas with respect to the validation accuracy starts decreasing at epoch number 7. Based on this observation the epoch number 7 is taken to test the model. Further the right graph shows that the training and validation loss of the BERT model. It can be understood that the training loss gradually decreases whereas the validation loss is increasing till epoch number 7 and starts to decrease in 5 and again start increasing at the epochs 6 and 7. Based on this observation, we took the epoch 7 is suitable number of epochs for the training and validation set.

Table 4. shows the loss, accuracy and validation loss metrics of the BERT model. It can be observed that the loss in epoch 1 is 0.3223 and in epoch 7 is 0.0153 which gradually decreased. Meanwhile with respect to accuracy it starts with the accuracy of 0.9058 in epoch 1 and finally it ends with 0.9959 in the epoch number 7.

Table 4. Epochs with Loss, Accuracy and Validation Loss Metrics of BERT Algorithm

	Loss	**Accuracy**	**Val_loss**
Epoch-1	0.3223	0.9058	0.314
Epoch-2	0.1856	0.9475	0.2546
Epoch-3	0.1298	0.957	0.3612
Epoch-4	0.0797	0.9728	0.3717
Epoch-5	0.0421	0.9877	0.4166
Epoch-6	0.024	0.9934	0.4295
Epoch-7	0.0153	0.9959	0.5162

Table 5. gives the performance metrics such as precision, recall, f1 score, support, macro average and weighted average values of BERT model. Also, it shows the individual class precision, recall, f1 score and support values of the models. Also, for the class Homophobic the precision, recall, f1score is 0.920, 0.980, 0.950 which is high when compared to other classes. Meanwhile for transphobic all the values are zero since this class contains the less samples.

Table 5. Performance Metrics of BERT Algorithm

	Precision	**Recall**	**F1 score**	**Support**
NALC	0.080	0.020	0.030	58
Homophobic	0.920	0.980	0.950	732
Transphobic	0.000	0.000	0.000	2
Macro Avg.	0.33	0.33	0.33	792
Weight Avg.	0.86	0.91	0.88	792

4.1.3 Performance evaluation of XLM-RoBERTa Model

Table 6. gives the performance metrics such as precision, recall, f1 score, support, macro average and weighted average values of XLM-RoBERTa model. It shows the individual class precision, recall, f1 score and support values. Also, for the class Homophobic the precision, recall, f1score is 0.95, 0.95, 0.95 which is high when compared to other classes. Meanwhile for transphobic all the values are zero since this class contains the less samples.

Table 6. Performance Metrics of XLM-RoBERTA Model

	Precision	**Recall**	**F1 score**	**Support**
NALC	0.61	0.29	0.40	58
Homophobic	0.95	0.99	0.97	732
Transphobic	0.000	0.000	0.000	2
Macro Avg	0.52	0.43	0.45	792
Weighted Avg	0.92	0.93	0.92	792

Table 7. records the performance comparisons of BERT and XLM-RoBERTa model accuracies. It is seen that the accuracy of BERT model is 91% and for XLM-RoBERTa is 93%. The XLM-RoBERTa model achieved increased performance when compared to the BERT model. Further, the performance of the BERT and XLM-RoBERTa model is presented in Figure. 4 The weighted precision, recall and f1 score of BERT Model is 0.86, 0.91, 0.88 whereas the for the XLM-RoBERTa, they are 0.92,0.93,0.92.

Table.7 Accuracy of BERT and XLM-RoBERTA Models

Model	Accuracy(%)
BERT	91
XLM-RoBERTA	93

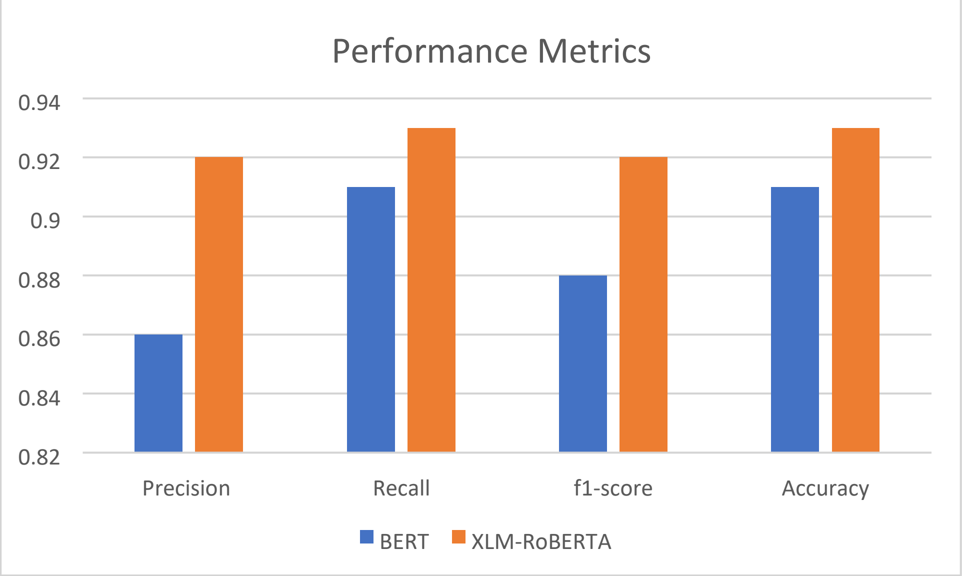

Figure 4. Accuracy of BERT and XLM-RoBERTA Models

5.Conclusion and Future Work

The task of identifying offensive content present in the code-mixed dataset of sentiment analysis with Non-Anti LGBT content (NALC), homophobic and transphobic considered and implemented. Along with this the experiment is tested with the test data with 990 comments without label. In this work. we compared the transformer-based models BERT and XLMRoBERTa using word tokenizer to extract the features from the dataset. Based on the experimental results, the XLM-RoBERTa achieved superior results with respect to the precision, recall and f-measure than BERT model. Also, with respect to the accuracy, the BERT gives 91% and XLM-RoBERTa gives 93%. In future, we plan to address the imbalanced nature of the dataset.

References

[1] Chakravarthi, B. R., Priyadharshini, R., Ponnusamy, R., Kumaresan, P. K., Sampath, K., Thenmozhi, D., McCrae, J. P. (2021). Dataset for identification of homophobia and transophobia in multilingual YouTube comments. arXiv preprint arXiv:2109. 00227.

[2] Agarwal, Shivang, and C. Ravindranath Chowdary. "Combating hate speech using an adaptive ensemble learning model with a case study on COVID-19." Expert Systems with Applications 185 (2021): 115632.

[3] Davidson, Thomas, Dana Warmsley, Michael Macy, and Ingmar Weber. "Automated hate speech detection and the problem of offensive language." In *Proceedings of the international AAAI conference on web and social media*, vol. 11, no. 1, pp. 512-515. 2017.

[4] Pitsilis, Georgios K., Heri Ramampiaro, and Helge Langseth. "Effective hate-speech detection in Twitter data using recurrent neural networks." *Applied Intelligence* 48, no. 12 (2018): 47304742.

[5] Shanmugavadivel, K., Subramanian, M., Kumaresan, P. K., Chakravarthi, B. R., B, B., Chinnaudayar Navaneethakrishnan, S., (2022). Overview of the Shared Task on Sentiment Analysis and Homophobia Detection of YouTube Comments in Code-Mixed Dravidian Languages. Working Notes of FIRE 2022 - Forum for Information Retrieval Evaluation. Hybrid: CEUR.

[6] Beddiar, Djamila Romaissa, Md Saroar Jahan, and Mourad Oussalah. "Data expansion using back translation and paraphrasing for hate speech detection." *Online Social Networks and Media* 24 (2021): 100153.

[7] Hande, Adeep, Siddhanth U. Hegde, and Bharathi Raja Chakravarthi. "Multi-task learning in under-resourced Dravidian languages." *Journal of Data, Information and Management* 4, no. 2 (2022): 137-165.

[8] Rodriguez, Axel, Yi-Ling Chen, and Carlos Argueta. "FADOHS: Framework for Detection and Integration of Unstructured Data of Hate Speech on Facebook Using Sentiment and Emotion Analysis." *IEEE Access* 10 (2022): 22400-22419.

[9] Le-Hong, Phuong. "Diacritics generation and application in hate speech detection on Vietnamese social networks." *Knowledge-Based Systems* 233 (2021): 107504.

[10] Karayiğit, Habibe, Çiğdem İnan Acı, and Ali Akdağlı. "Detecting abusive Instagram comments in Turkish using convolutional Neural network and machine learning methods." *Expert Systems with Applications* 174 (2021): 114802.

[11] Lan, Zhenzhong, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. "Albert: A lite bert for self-supervised learning of language representations." *arXiv preprint arXiv:1909.11942* (2019).

[12] Nascimento, Francimaria RS, George DC Cavalcanti, and Márjory Da Costa-Abreu. "Unintended bias evaluation: An analysis of hate speech detection and gender bias mitigation on social media using ensemble learning." *Expert Systems with Applications* 201 (2022): 117032.

[13] Roy, Pradeep Kumar, Snehaan Bhawal, and Chinnaudayar Navaneethakrishnan Subalalitha. "Hate speech and offensive language detection in Dravidian languages using deep ensemble framework." *Computer Speech & Language* 75 (2022): 101386.

[14] 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.

[15] Watanabe, Hajime, Mondher Bouazizi, and Tomoaki Ohtsuki. "Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection." *IEEE access* 6 (2018): 13825-13835.

[16] Premjith B, Bharathi Raja Chakravarthi, Malliga Subramanian, Bharathi B, Soman KP, Dhanalakshmi V, Sreelakshmi K, Arunaggiri Pandian and Prasanna Kumar Kumaresan,"Findings of the Shared Task on Multimodal Sentiment Analysis and Troll Meme Classification in Dravidian Languages", In Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages, pp. 254-260. 2022.