Transformer Based Model For Offensive Content Recognition In Dravidian Languages

S Divya [1], N Sripriya [2]

[1,2] *Department of Information Technology, Sri Sivasubramaniya Nadar College of Engineering, Chennai, India*

Abstract

This paper describes a model for spotting offensive data from the comments being collected from social media. The comments posted will include expressions, emoticons and will mostly be in code mixed language and classifying these code-mixed language comments is tricky. The proposed system uses a multi-head attention model to extract features from the code-mixed Tamil input data. Various classification algorithms are applied to these extracted features to categorize offensive comments. The generated labels are optimized by performing majority voting on labels generated by different algorithms. This system is validated on the validation set and is evaluated by applying the Tamil CodeMix test data from the dataset published by the HASOC task (Task2-subtask1) at FIRE 2021. The evaluation yields an average weighted F1 score of 0.83 and is ranked 3rd position in the official ranking.

Keywords

Offensive, emoticons, Code-mix, Natural Language Processing.

1. Introduction

Universal internet access provides users with the liberty to share their thoughts and expressions in various mediums like business pages, blogs, etc. This allows interaction among people of different cultures and origins. Information sharing among various people can be done within seconds. People are open to express their positive and negative opinions about the content available on social media. The negative opinion on a given content can be expressed by using profane words, abusive language, or displeased expressions. This negative expression may be aggressive or degrade the self-esteem of the one who views the comment. The increase in the utilization of offensive words in social media accounts leads to the identification of abusive and offensive words which is an anti-harassment policy on social media [1].

This hate and offensive content posted on the web is a threat and a challenge to society. Many countries forbid offensive words in social media intending to avoid annoying viewers or trigger any misdeed. As negative opinions are mostly expressed with hate or offensive words, a variety of online forums define their policies to detect and ignore abusive words that create a negative impact on the community. Since there is no explicit definition for hate or offensive speech, and as it can vary depending on its context, detecting such abusive words is challenging. Even in the absence of ubiquitous clarity for identifying abusive words [2] stated a definition that, hate or abusive speech is defined as "any kind of communication that disparages a target group of people based on their specific characteristics such as culture, complexion, gender, nationality, religion or any other characteristics [1].

Manual identification and elimination of abusive words in online content are complex due to the availability of fast-growing online users. This task becomes still more complex due to the presence of unknown users in such forums who share their opinion in a language that is self-understandable and

FIRE 2021, Forum for Information Retrival Evaluation, December 13-17, 2021, India
EMAIL: divyas@ssn.edu.in@ssn.edu.in(S.Divya); sripriyan@ssn.edu.in (N.Sripriya)

maybe code mixed. This paved a demand for a model or technique that automates the identification of abusive and insulting remarks promptly and quickly from the online content.

This paper describes a model that could identify the hate speech and offensive content from the Indo-European language (HASOC) track at FIRE 2021. The overview of the existing systems in HASOC is presented in section 3. The transformer model for embedding is explained in section 4. Various classification algorithms applied to classify the offensive and non-offensive comments are detailed in section 5. The task description for the proposed model for HASOC and the system evaluation is explained in section 6. The concluding section concludes the work with few remarks and future scope.

2. Related work

Even when describing and realizing objectionable words is tedious, several kinds of research have been carried out in the opinion extraction domain to automate the identification of abusive words in online content. A combination of Artificial Intelligence and Natural Language Processing has paved the way for a variety of approaches in this task. In most scenarios, determining the level of intensity of mood or moods (positive/negative) can be an effective attribute in exploring the views of hate speech identification. Machine learning algorithms are used to classify the content based on their essential or relevant words and phrases [5],[6]. Subjective and non-substantive functions in the input are detected and are used for the conceptual classification of the input. These feature detections are done using methods such as parts of speech tagging, a bag of words, character n-grams, word, and character frequency, and so on. Once the contextual feature extraction is done, machine learning algorithms are used to classify text based on the component terms and expressions. Challenges in automating the offensive word identification are analyzed and a multi-view Support Vector Machine (SVM) [4] model is proposed that attains a performance closer to the state-of-the-art model. A multi-class classifier is proposed to categorize tweets into various classes such as abusive words, either abusive or offensive, neither abusive nor offensive, etc.[3]. A combination of SVM and Logistic Regression (LR) [7] is applied to the extracted features to detect abusively or hate speech.

These techniques perform better in languages that have regular grammar. Most of the comments posted by the social media users are in non-formal language code-mixing or Code Switching will not be in proper grammar format. The challenge in the processing of tasks using Code-mixed data is the lack of text data in such languages. a corpus incorporating Tamil-English Code-mixed data [17] is collected and is annotated to perform a task in such data. This dataset comprises comments from YouTube which is being annotated and benchmarked for the Sentiment Analysis task [15], [16].

Various Deep Learning models such as Recursive Neural Networks (RNN), Long Short-Term Memory (LSTM) are applied to detect hate speech by considering the dependency among the previous content in the input. Embedding models like the Fast Text, Glove, BERT [8] are used to translate the highly scattered n-dimensional vectors in a comparatively low-dimensional space. These embeddings make it easier for machine learning algorithms to be applied on vast volumes of data in which the terms are represented as vectors.

The representation of this data is generated using Term Frequency Inverse Document Frequency (TF/IDF) and this is then utilized for training various traditional ML algorithms. Corpus collection of Kannada-English Code-mixed dataset for multi-tasking such as Sentiment Analysis and offensive language identification is done [18],[19]. This dataset comprises 7,671 comments that are annotated and are benchmarked using computational models. As a basic system, various traditional ML algorithms are applied to this data and are evaluated[8], [9], [10]. A collection of 1200 Hindi and Marathi documents from comments [23] was generated on social media. This dataset is applied on a model derived with the combination of Naive Bayes (NB), Support Vector Machine (SVM) using Radial Basis Function (RBF), and Linear Kernels[11], [12]. An accuracy of 90% is obtained on the Marathi dataset and a range of 70 % to 80% is obtained on the Hindi dataset[13], [14].

A model based on Contrastive learning using twin BiLSTM networks and a clustering-based method to extract Code-mixed transliterated words. Based on the variation in configurations of language pairs, accuracy in a range of 70& 79% is obtained [20]. A sub-word LSTM architecture for learning representations in the sub-word level for sentiment analysis is proposed[21]. This supports

the ability to learn sentiment value information about important morphemes. Effective learning on the input data to generate representation[22] helps in better performance of the required task. The proposed method learns the code-mixed input using a model that can understand multiple languages and then subjects the representation to a certain model for classification.

3. Embedding using Multilingual BERT

The comments shared online will be in a code-mixed language that can facilitate easy public opinion. This becomes a challenge for automating the rating of moderate and abusive terms in the comments. For instance, the comments may be posted in English or Tanglish (Tamil + English). A system to identify the language, understand the comments, and detect the harsh or offensive terms in this input is challenging. Creating representation for each sentence is done using a language-specific element that identifies the sentence language and specific features that extract the meaning of the sentence. With an assumption that each sentence is in the same language, the information conveyed in the sentences is extracted through this representation. Sentence representation is generated using multilingual BERT in which the BERT base model is additionally pre-trained with randomly selected phrases from 104 languages to make it applicable to different language classifications.

Five exploring tasks are incorporated to estimate how Multilingual BERT handles diverse language sentences as a single language sentence.

- To identify the language, a classifier is added on top of the representation for identifying the language of the sentence.
- Since sentences with similar language tend to have similar representations, evaluation of similar language sentences being cluster together is done using V-measure over hierarchical clustering [9].
- The distance between representations is calculated for each sentence in the multi-parallel corpus and the sentences with the least distance are retrieved. In each language, linear regression is equipped to project the other language representations into English representation space with a minimum set of parallel sentences.
- Processing bilingual statements require communication on a word level, even when sentence retrieval can be done with keyword recognition. Word alignment is determined as a minimum weighted bipartite graph. The tokens in the sentences of dissimilar languages are connected and the edges are weighted using the cosine distance between the representation.
- The quality of the Machine Translation system has been determined without accessing the reference translation. The correlation with the number of edits that a human must do to attain the Human targeted summary is the evaluation metric for the Machine Translation system. The quality of translation is determined using the cosine distance between the source sentence representation and the device translation.

This embedding technique is applied to the input data to generate a representation for sentences with dissimilar languages. A classifier is placed on top of this representation to categorize the sentences based on the requirements.

4. Classification Algorithms

Three different machine learning procedures are applied to classify the data based on the offensive content available in the input data. The classifiers are given with certain labeled data from which the model extracts the characteristics that support learning the offensive and non-offensive content.

4.1. Support Vector Machine Classifier

To categorize the information can be found points being projected in the n-dimensional plane, Support Vector Classifier (SVM) [10]. This is accomplished by obtaining an optimum hyperplane among dissimilar categories of data points. A binary class classification problem can be solved by placing a hyper-plane between two categories of data points. This can be done by selecting many

possible hyper-planes between two classes. This tends to find a plane that has a maximum distance between data points of two different classes. Increased marginal distance between two classes helps in the convenient classification of additional existing data points.

These hyper-planes tend to be the decision perimeter that supports classifying the data points. Data points that fall on either side of the plane are assigned to various categories. The hyperplane size is based on the number of input features. Data points that appear closer to the hyper-planes are named as support vectors and this controls the location and angle of the hyper-planes. This helps to increase the marginal distance between the two classes.

4.2. Extreme Gradient Boosting Classifier (XGBoost)

A gradient boosting framework that works based on the decision tree ensemble learning model. XGBoost algorithm [11] is evolved from decision trees along with some additional features summed up to outperform all other frameworks.

A decision tree is utilized to produce feasible solutions as a decision on certain conditions and generates a graphical representation. A combination of multiple solutions derived from different decision trees is done through voting methods that are a meta-algorithm known as bootstrap aggregation or bagging.

Random Forest is a bagging-based technique where only a selective batch of features are chosen to produce a group of decision trees. The effectiveness of this model is enhanced by successive constructing models by diagnosing the disadvantages of the previous model. This is known as boosting. Gradient descent is incorporated with the boosting technique to reduce the errors that occur in the sequentially constructed model. As an update, optimization of gradient boosting is done using additional features like parallel processing, tree-pruning, handling missing values, standardization to avoid biasing or overfitting.

4.3. Linear Discriminant Analysis (LDA)

For each class, the statistical property of the data is estimated. For every single input data, the meaning and variance of the data in each class are determined. In the case of multiple variables, a bell-shaped curve, known as Gaussian, is used to estimate means and covariance matrix. These statistical properties are fed into the LDA equation to facilitate classification.

The input data must be prepared before that is applied to the LDA [12]. The outliers must be removed and the input data must be standardized. The LDA model uses Bayes' theorem to evaluate the probability of input data belonging to each class. The prediction is done on the basis that the class with the greatest probability is the output class.

5. Task Description and Proposed Model

Hate Speech and Offensive Content identification HASOC in Indo-European languages focus on identifying the abusive content in code-mixed languages such as English, Malayalam, Hindi. To identify the offensive content, the input data are gathered from posts and comments shared on Twitter and Facebook.

This task comprises two sub-tasks. Task 1 is a message-level classification task that develops a system to categorize the comments produced in Tamil. Task 2 is a message-level classification task that builds a system to automatically classify the code-mixed Tamil and Malayalam tweets into offensive and non-offensive classes.

In this proposed model, an automatic classification system is built to classify the offensive and non-offensive content in the code-mixed Tamil comments. The sample for the code-mixed Tamil data is shown below.

Table 1
Sample for CodeMix dataset

ID	Text	Label
tl_1	Yarayellam FDFS ppga ippove ready agitinga	Off
tl_2	Ennada viswasam mersal sarkar madhri time la likes and views create pannalayae	Not

The proposed system to automatically classify the offensive content from the given dataset is described below. The input data must be pre-processed before being converted into an n-dimensional vector and being classified into offensive and non-offensive content. The pre-processing includes the removal of duplicate words, web links, emoticons, symbols, hashtags, numbers, names, etc. The architecture diagram for the proposed model is given below.

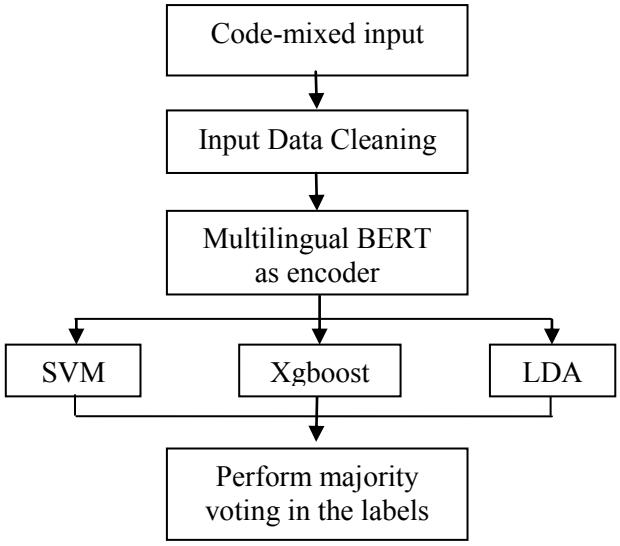

Figure 1: Architecture diagram for the proposed model

The pre-processed data is now fed as input to the embedding system. Multilingual BERT is used for generating embeddings. Since the input data may be of code-mixed language, the embedding model that processes various languages is used for generating embeddings. This projects the input sentence in an n-dimensional plane to generate an n-dimensional feature vector. These vectors are fed as input to the three different classifiers.

Support Vector Machine, Xgboost, Linear Discriminant Analysis are applied with these n-dimensional features for classifying the data into two classes. One class holds the sentences that do not have offensive words and the other holds the offensive sentences. Three various classification algorithms generate three distinct labels for each sentence. The label which is generated in the majority will be the final class label for the sentence. (Off- Offensive sentence, Not- Non-offensive sentence). The example for optimizing the class label for a sentence is given below.

Table 2
Optimizing the labels from different classification algorithms

Input	SVM	Xgboost	LDA	Final Label

Sentence 1	Off	Off	Not	Off
Sentence 2	Off	Not	Not	Not
Sentence 3	Off	Not	Off	Off

Thus, the final label to classify the offensive and non-offensive input sentences from the input is performed based on the majority voting on the labels generated by the various classification algorithms.

5.1. System Evaluation

In the proposed model, sentence embeddings are done using a system that understands and interprets multiple languages. The embedded values are classified using three distinct classification algorithms and the optimized label is considered as the final label for each input data. The dataset used for the identification of offensive language identification is Tamil code-mixed train data from the dataset published by the HASOC task (Task2-subtask1) at FIRE 2021 The dataset comprises of 4000code-mixed training data, 940 validation data, and 1001 test data. The proposed model is trained with a 4000 training set and is validated with a 940 validation set. The labels generated for the validation dataset are evaluated using weighted average classification metrics. The classification report is given below.

Table 3
Classification Report

	Precision	Recall	F1-Score	Support
Not	0.82	0.85	0.83	465
Off	0.84	0.81	0.83	475
Accuracy	-	-	0.83	940
Macro avg	0.83	0.83	0.83	940
Weighted avg	0.83	0.83	0.83	940

The above table indicated the precision, recall, and F1-score of the "NOT" and ÖFF" class labeled for each input data. Here, the metric "support" shows the count of test samples given for evaluating the system. The system is evaluated using a weighted average F1-score. This is the weighted average of precision and recall. F1 score is calculated with the relative contribution from precision and recall.

6. Conclusion and Future work

Automatic classification of offensive contents in code-mix Tamil data is done by optimizing the labels generated by three different classification algorithms. The labels generated by Machine learning approaches produce a promising output. As future work, the identification of offensive content must be done using Deep learning approaches to exactly identify the abusive words and classify the data exactly.

7. Acknowledgements

We sincerely thank the management of SSN Institutions for the infrastructure and lab facilities to carry out this research work.

8. References

[1] de Gibert, Ona, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. "Hate speech dataset from a white supremacy forum." arXiv preprint arXiv:1809.04444 (2018).

[2] Nockleyby, J. "'Hate speech in Encyclopedia of the American Constitution." Electronic Journal of Academic and Special librarianship (2000).
[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. 2017.
[4] MacAvaney, Sean, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. "Hate speech detection: Challenges and solutions." PloS one 14, no. 8 (2019): e0221152.
[5] Bruce, Rebecca F., and Janyce M. Wiebe. "Recognizing subjectivity: a case study in manual tagging." Natural Language Engineering 5, no. 2 (1999): 187-205.
[6] Wiebe, Janyce, Rebecca Bruce, and Thomas P. O'Hara. "Development and use of a gold-standard data set for subjectivity classifications." In Proceedings of the 37th annual meeting of the Association for Computational Linguistics, pp. 246-253. 1999.
[7] Waseem, Zeerak. "Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter." In Proceedings of the first workshop on NLP and computational social science, pp. 138-142. 2016.
[8] Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
[9] Rosenberg, Andrew, and Julia Hirschberg. "V-measure: A conditional entropy-based external cluster evaluation measure." In Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), pp. 410-420. 2007.
[10] Tong, Simon, and Daphne Koller. "Support vector machine active learning with applications to text classification." Journal of machine learning research 2, no. Nov (2001): 45-66.
[11] Qi, Zhang. "The text classification of theft crime based on TF-IDF and XGBoost model." In 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), pp. 1241-1246. IEEE, 2020.
[12] Sharma, Alok, and Kuldip K. Paliwal. "Linear discriminant analysis for the small sample size problem: an overview." International Journal of Machine Learning and Cybernetics 6, no. 3 (2015): 443-454.
[13] Banerjee, Shubhanker, Bharathi Raja Chakravarthi, and John P. McCrae. "Comparison of pretrained embeddings to identify hate speech in Indian code-mixed text." In 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), pp. 21-25. IEEE, 2020.
[14] Mandl, Thomas, Sandip Modha, Anand Kumar M, and Bharathi Raja Chakravarthi. "Overview of the hasoc track at fire 2020: Hate speech and offensive language identification in tamil, malayalam, hindi, english and german." In Forum for Information Retrieval Evaluation, pp. 29-32. 2020.
[15] Hande, Adeep, Siddhanth U. Hegde, Ruba Priyadharshini, Rahul Ponnusamy, Prasanna Kumar Kumaresan, Sajeetha Thavareesan, and Bharathi Raja Chakravarthi. "Benchmarking multi-task learning for sentiment analysis and offensive language identification in under-resourced dravidian languages." arXiv preprint arXiv:2108.03867 (2021).
[16] Chakravarthi, Bharathi Raja, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John P. McCrae. "A sentiment analysis dataset for code-mixed Malayalam-English." arXiv preprint arXiv:2006.00210 (2020).
[17] Chakravarthi, Bharathi Raja, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John P. McCrae. "Corpus creation for sentiment analysis in code-mixed Tamil-English text." arXiv preprint arXiv:2006.00206 (2020).
[18] Hande, Adeep, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. "KanCMD: Kannada CodeMixed dataset for sentiment analysis and offensive language detection." In Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media, pp. 54-63. 2020.
[19] Hande, Adeep, Siddhanth U. Hegde, Ruba Priyadharshini, Rahul Ponnusamy, Prasanna Kumar Kumaresan, Sajeetha Thavareesan, and Bharathi Raja Chakravarthi. "Benchmarking multi-task learning for sentiment analysis and offensive language identification in under-resourced dravidian languages." arXiv preprint arXiv:2108.03867 (2021).

[20] Mandl, Thomas, Sandip Modha, Anand Kumar M, and Bharathi Raja Chakravarthi. "Overview of the hasoc track at fire 2020: Hate speech and offensive language identification in tamil, malayalam, hindi, english and german." In Forum for Information Retrieval Evaluation, pp. 29-32. 2020.
[21] Ghanghor, Nikhil, Parameswari Krishnamurthy, Sajeetha Thavareesan, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. "IIITK@ DravidianLangTech-EACL2021: Offensive Language Identification and Meme Classification in Tamil, Malayalam and Kannada." In Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pp. 222-229. 2021.
[22] Banerjee, Shubhanker, Arun Jayapal, and Sajeetha Thavareesan. "NUIG-Shubhanker@ Dravidian-CodeMix-FIRE2020: Sentiment Analysis of Code-Mixed Dravidian text using XLNet." arXiv preprint arXiv:2010.07773 (2020).