Sentiment analysis based offensive language identification system for code-mixed data

Kogilavani Shanmugavadivel[1], Malliga Subramanian[2], Sathishkumar V E[3,4]

[1]Department of Artificial Intelligence, Kongu Engineering College, Erode, Tamil Nadu, India
[2]Department of Computer Science and Engineering, Kongu Engineering College, Erode, Tamil Nadu, India
[3]Department of Industrial Engineering, Hanyang University, Seoul, Republic of Korea

[4]sathishkumar@hanyang.ac.kr

Abstract. Hate speech identification is the process of categorising textual information into hate and non- hate speech with the goal of identifying hate speech's targeted features. The objective of this research work is to take the Dataset from FIRE 2021 shared task code mixed data that includes YouTube comments and Twitter conversations and to detect whether the messages represents the offensive or non-offensive category. To detect the offensive language sentences, various deep learning models like Long Short-Term Memory, Bidirectional Long Short-Term Memory, Convolutional Neural Network, Gated Recurrent Unit, and hybrid model like Convolutional Neural Network with Bidirectional Long Short-Term Memory methods were utilised in this research work. The performance of all the mentioned models is evaluated using precision, recall, F1-score, and accuracy. Out of all the models, both LSTM and GRU models perform better with the accuracy of 0.81, precision of 0.85 and recall of 0.95.

Keywords: Sentiment Analysis, Offensive Language, Deep Learning, Code-Mixed Data, Regional Language.

1. Introduction

Social media web sites such as Twitter, Facebook, and YouTube have exploded in popularity. As a result, hate speech and hate-related activities are become increasingly common. The anonymity and mobility given by online media has made the cultivation and spread of offensive langauge. Hate speech is considered as any kind of communication, whether verbal, written, or physical, that is directed towards or uses disparaging or discriminatory words in reference to a person or a group based on who they are, such as their religion, ethnicity, or country. The impacts of hate crimes are already overwhelming due to the vast usage of social media and the anonymity enjoyed by internet users [1]. Although not everyone who engages in hate speech commits violent acts, many hate crime offenders attribute their motivation to hate speech on the internet. Hate speech directed towards minorities has the potential to mainstream prejudice, hate crime, and targeted violence. Text is a tremendously rich data source. Racism is the belief that various groups of people have different behavioral features that correspond to hereditary characteristics, and that these groups may be divided based on race superiority [2]. Expressions that have the potential to inspire fury or violence are known as intoxicating language [3].

The objective of this research work is to detect the problematic statements in a mix of Dravidian language comments that are stated in Tamil-English and acquired from social media. At the comment level each sentence is tagged with two different class labels. The data set was taken from a FIRE 2021 shared task that included YouTube comments and tweets. The proposed system must distinguish between offensive and non-offensive comments or messages. The major purpose is to identify blended sentences and apply a label that detects the text's sentiment, such as offensive or not offensive.

2. Literature Survey

A brief review is provided about nearly 150 deep learning-based models implemented in recent years mainly for the classification of text [4-6]. They also mentioned technical methodologies, uniqueness of each model and applications. Overview of hate speech detection techniques, challenges and future directions are discussed in [7-11]. They provided a detailed survey that specifies key areas that have been adopted to automatically recognize the content of text using natural language processing techniques. In general authors collect and label their own data. Due to the lack of unique feature identification, doing analysis in social media data is a very challenging task [12-13]. They developed Deep Neural Network models and evaluated all models using a large hate speech dataset collected from Twitter. Models like Long Short Term Memory (LSTM) and Bi- directional Long Short-Term Memory (Bi-LSTM) were implemented and found that LSTM produced better precision, F1- Score and accuracy [14-16]. For toxic comments identification, Bi-LSTM produced better recall values in [17]. Data may be represented using multiple languages can be considered as code-mixed data which come from multilingual users is more difficult to analyze.

3. Proposed System

The proposed system is used to develop an effective method for classifying the Dravidian code-mixed data set obtained from the FIRE 2021 shared work into offensive and non-offensive categories. Various pre-processing techniques are performed to clean the data set throughout the data exploration phase. LSTM, BI-LSTM, CNN, and GRU are among the deep machine learning models used to develop models. Hybrid model such as combining CNN with LSTM, are also used to increase performance even further. The proposed system workflow is represented in Figure 1.

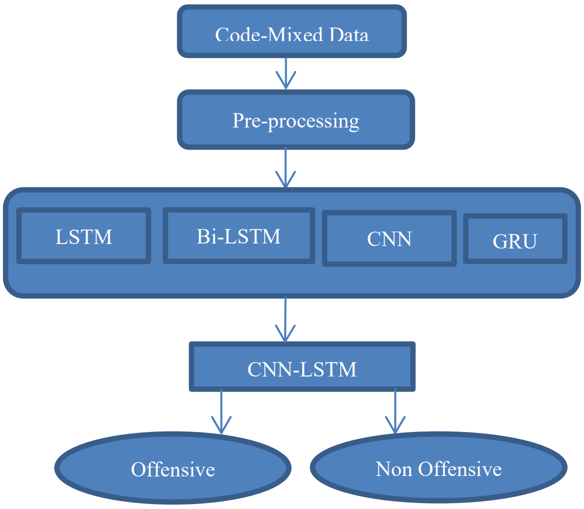


Figure 1. Proposed System Workflow.

3.1. Dataset Collection

The data is collected from a FIRE 2021 (HASOC) challenge in Dravidian Languages [20]. The data set contains code-mixed sentences in Tamil Dravidian Language. Train Dataset contains 5877 sentences, Test dataset contains 654 sentences and validation dataset contains 654 sentences. All of them are divided into two categories: offensive and non-offensive. The dataset includes the total number of comments from each of two classes and detailed description is presented in Table 1.

Table 1. Dataset Description.

Dataset	Number of Messages	Offensive	Non Offensive
Training	5876	4724	1153
Testing	654	536	118

3.2. Data Preprocessing

The data can have many irrelevant and missing parts. To handle this part, data preprocessing is done. In data preprocessing we have used data cleaning to remove the emojis and used label encoder in order to change the category from Offensive or Not offensive to 0's and 1's. Data preprocessing flow is depicted in Figure 2.

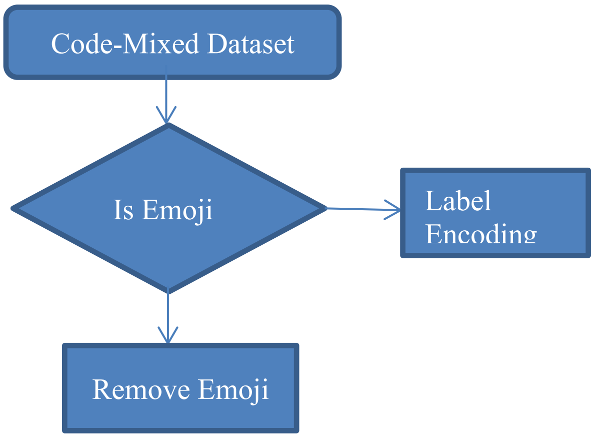


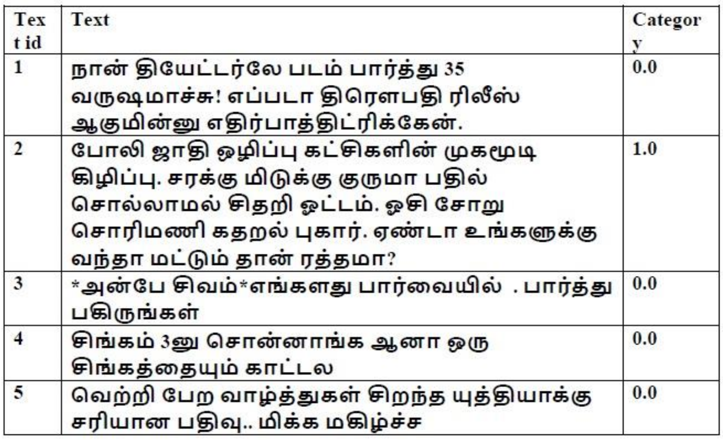
Figure 2. Data Preprocessing.

3.3. Data Cleaning

The process of finding sections of data that are incorrect, incomplete, inaccurate, irrelevant, or missing, and then altering, replacing, or deleting them as needed is known as data cleaning. Data cleaning is considered a crucial part of data science.

3.4. Label Encoding

For categorical data, label encoding is a popular encoding approach. The process of converting labels into numeric representation so that machines can read them is known as label encoding. After applying label encoding technique, the resultant text and the corresponding category is represented in Figure 3.



Text id	Text	Category
1	நான் தியேட்டர்லே படம் பார்த்து 35 வருஷமாச்சு! எப்படா திரௌபதி ரிலீஸ் ஆகுமின்னு எதிர்பாத்திட்ரிக்கேன்.	0.0
2	போலி ஜாதி ஒழிப்பு கட்சிகளின் முகமூடி கிழிப்பு. சரக்கு மிடுக்கு குருமா பதில் சொல்லாமல் சிதறி ஓட்டம். ஓசி சோறு சொரிமணி கதறல் புகார். ஏண்டா உங்களுக்கு வந்தா மட்டும் தான் ரத்தமா?	1.0
3	*அன்பே சிவம்*எங்களது பார்வையில் . பார்த்து பகிருங்கள்	0.0
4	சிங்கம் 3னு சொன்னாங்க ஆனா ஒரு சிங்கத்தையும் காட்டல	0.0
5	வெற்றி பேற வாழ்த்துகள் சிறந்த யுத்தியாக்கு சரியான பதிவு.. மிக்க மகிழ்ச்ச	0.0

Figure 3. Encoded Dataset.

3.5. Building Deep Learning Models

For training, given the collected dataset as input with two classes OFF and NOT, the learned model has to predict the category based on various parameter tuning. LSTMs are a form of RNN that uses long short-term memory networks for storing intermediate results. The default behaviour of the LSTM is to remember information for long periods. A bidirectional LSTM differs from a standard LSTM in that its input flows in two ways. With a conventional LSTM, we may make input flow in one direction, either backwards or forwards. We may, however, have flow of information in both directions with bi- directional input, maintaining both the future and the past. Many layers of artificial neurons make up convolutional neural networks constructed from different layers of artificial neurons. These neurons are represented with mathematical functions similar to real neurons which calculate the activation value from activation function. Hybrid learning methods are a way to improve the intended outcome. In order to improve the projected result, this study work employs CNN and LSTM. Convolutional Neural Network layers are used to extract information from the input data, and LSTMs are used to forecast order. Gated Recurrent Unit features a forget gate component, the GRU is quite similar to the LSTM, but it lacks an output gate and hence has fewer parameters.

4. Performance Evaluation

Evaluation metrics considered for this project work will be Precision, Recall and F1-Score. The results of various deep learning models are tabulated below. Among these LSTM, Bi-LSTM, CNN, CNN-LSTM, GRU models, LSTM model gives the highest accuracy with recall. LSTM gives the best result with 81% accuracy with 2 epochs and various performance measures are presented in Table 2. With the sequential_4 model, the embedding layer contains 901248 parameters and 66 dense layers.

Table 2. LSTM Performance Measure.

Class Labels	Precision	Recall	F1-Score	Support
NOT	0.50	0.21	0.30	118
OFF	0.85	0.95	0.90	536

Bi- LSTM gives the result with 80% accuracy with 2 epochs. With the sequential_1 model, the embedding layer contains 901248 parameters and 258 dense layers. The precision recall, F1-score values are presented in Table 3.

Table 3. Bi-LSTM Performance Measure.

Class Labels	Precision	Recall	F1-Score	Support
NOT	0.31	0.09	0.14	118
OFF	0.83	0.96	0.89	536

CNN gives the result with 75% accuracy with 2 epochs. With the sequential_1 model, the embedding layer contains 6697800 parameters, 129 dense layers and 268928 ConvID. Withsequential_2 model, the embedding layer contains 901248 params,6 dense layers and 20608 ConvID.With sequential_3 model, embedding layer contains 901248 params,6 dense layers and 8224 ConvID. The precision recall, F1-score values are presented in Table 4.

Table 4. CNN Performance Measure.

Class Labels	Precision	Recall	F1-Score	Support
NOT	0.26	0.19	0.22	118
OFF	0.83	0.88	0.85	536

CNN-LSTM gives the result with 79% accuracy with 50 epochs. With sequential_1 model, embedding layer contains 6697800 parameters,129 dense layers and 268928 ConvID.50 epochs have been run. The precision recall, F1-score values are presented in Table 5.

Table 5. CNN-LSTM Hybrid Model Performance Measure.

Class Labels	Precision	Recall	F1-Score	Support
NOT	0.36	0.23	0.28	118
OFF	0.84	0.91	0.88	536

GRU gives the result with 81% accuracy with 1 epoch. With sequential_9 model, the embedding layer contains 901248 parameters and 258 dense layers. The precision recall, F1-score values are presented in Table 6.

Table 6. GRU Model Performance Measure.

Class Labels	Precision	Recall	F1-Score	Support
NOT	0.42	0.17	0.24	118
OFF	0.84	0.95	0.89	536

5. Conclusion

The models are able to classify datasets as offensive or non-offensive for FIRE 2021 HASOC challenge in Dravidian Languages. In this research work various models like LSTM, BI-LSTM, CNN, GRU and hybrid models like CNN with LSTM were implemented. Out of all these models LSTM and GRU models achieved high accuracy of 81% to detect the offensive text with the given dataset. In future, the accuracy could beimproved by employing transfer learning methodology.

References

[1] S. Muthu Kumaran, P. Suresh and J. Amudhavel, "Sentimental analysis on online product reviews using LS-SVM method", Journal of Advanced Research in Dynamical and Control Systems, vol. 9, no. 12, pp. 1342-1352, 2017.

[2] S. A. Devi, P. Sapkota and M. Obulesh, "Sentiment analysis on products using social media”, Journal of Advanced Research in Dynamical and Control Systems, pp. 137-141, 2017.

[3] M. Bhargava and D. Rao, "Sentimental analysis on social media data using R programming”, International Journal of Engineering and Technology(UAE), vol. 7, no. 2, pp.80-84, 2018.

[4] S. Lai, L. Xu, K. Liu, and J. Zhao, “Recurrent convolutional neural networks for text classification”, in Twenty-ninth AAAI conference on artificial intelligence, 2015.

[5] C. Zhou, C. Sun, Z. Liu, and F. Lau, “A c-lstm neural network for text classification”, arXiv preprint arXiv:1511.08630, 2015.

[6] M. Tan, C. d. Santos, B. Xiang, and B. Zhou, “Lstm-based deep learning models for non-factoid answer selection”, arXiv preprint arXiv:1511.04108, 2015.

[7] X. Zhou, X. Wan, and J. Xiao, “Attention-based lstm network for cross-lingual sentiment classification”, in Proceedings of the conference on empirical methods in natural language processing, 2016, pp. 247–256.

[8] D. S. Sachan, M. Zaheer, and R. Salakhutdinov, “Revisiting lstm networks for semi-supervised text classification via mixed objective function”, in Proceedings of the AAAIConference on Artificial Intelligence, vol. 33, 2019, pp. 6940–6948.

[9] Mohtaj, V. Woloszyn, S. Möller, “TUB at HASOC 2020: Character based LSTM for hate speech detection in indo-european languages”, in FIRE (Working Notes), CEUR, 2020.

[10] R. Raj, S. Srivastava, S. Saumya, “NSIT & IIIT DWD @HASOC 2020: Deep learning model for hate-speech identification in indo-european languages”, in: FIRE (Working Notes),CEUR, 2020.

[11] H. Madhu, S. Satapara, H. Rathod, Astralis “@HASOC 2020: Analysis on identification of hate speech in indo-european languages with fine-tuned transformers”, in: FIRE (Working Notes), CEUR, 2020.

[12] M. Iyyer, W.-t. Yih, and M.-W. Chang, “Search-based neural structured learning for sequential question answering”, in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 1821–1831

[13] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, “Glue: A multi-task benchmark and analysis platform for natural language understanding”, arXiv preprintarXiv:1804.07461, 2018.

[14] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, “Fasttext. Zip: Compressing text classification models”, arXiv preprint arXiv:1612.03651, 2016.

[15] P. Liu, X. Qiu, and X. Huang, “Recurrent neural network for text classification with multi-task learning”, arXiv preprint arXiv:1605.05101, 2016.

[16] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A convolutional neural network for modelling sentences”, in 52nd Annual Meeting of the Association for Computational Linguistics ACL 2014 - Proceedings of the Conference, 2014.