NITP-AI-NLP@HASOC-Dravidian-CodeMix-FIRE2020: A Machine Learning Approach to Identify Offensive Languages from Dravidian Code-Mixed Text

Abhinav Kumar[a], Sunil Saumya[b] and Jyoti Prakash Singh[a]

[a]*National Institute of Technology Patna, Patna, India*
[b]*Indian Institute of Information Technology Dharwad, Karnataka, India*
[a]*National Institute of Technology Patna, Patna, India*

Abstract

Hate speech in social media has posed a threat to society. Several models for a single language, mostly English hate speech is proposed recently. However, In countries where English is not the native language, communication involves scripts and constructs of more than one language yielding code mixed text. The current work classifies offensive and non-offensive tweets or YouTube comments written in code-mix Tamil, code-mixed Malayalam, and script-mixed Malayalam languages. We explored deep learning models such as attention-based Long Short Term Memory (LSTM), Convolution Neural Network (CNN), and machine learning models such as support vector machine, Logistic regression, Random forest, and Naive Bayes to identify offensive posts from the code-mixed and script-mixed posts. From the extensive experiments, we found that the use of character N-gram Term Frequency-Inverse Document Frequency (TF-IDF) features plays a promising role in identifying offensive social media posts. The character N-gram TF-IDF based Naive Bayes classifier performed best with the weighted precision, recall, and F_1-score of 0.90 for Tamil code-mixed text. The Logistic regression classifier with character N-gram TF-IDF features performed best with the weighted precision, recall, and F_1-score of 0.78 for Malayalam code-mixed text. The Dense Neural Network with character N-gram TF-IDF features performed best with the weighted precision of 0.96, recall of 0.95, and F_1-score of 0.95 for Malayalam script-mixed text.

Keywords

Hate speech, Code-mixed, Script-mixed, Machine learning, Deep learning

1. Introduction

Social media such as facebook twitter is flooded with user generated content [1, 2, 3, 4]. In particular, hate speech on social media is on the rise at a rapid pace[1] posing a significant threat to the sustainable society. Social media YouTube defines hate speech as "any speech that involves race, age, sexual orientation, disability, religion, and racism to promote hate or violence among groups" [2]. Through these social platforms, the hate speech is reaching to concerned person even in their bedroom and last forever [5]. The hate speech has a terrible impact on users' mental status, resulting in depression, sleeplessness, and even suicide. It is challenging for the authorities to prove someone guilty due to their anonymous identity and across border laws since some countries have the freedom of expression to write. In contrast, others adopt a very stringent policy for the same message [6].

FIRE 2020: Forum for Information Retrieval Evaluation, December 16-20, 2020, Hyderabad, India

EMAIL: abhinavanand05@gmail.com (A. Kumar); sunil.saumya@iiitdwd.ac.in (S. Saumya); jps@nitp.ac.in (J.P. Singh)
ORCID:

[1] https://ucr.fbi.gov/hate-crime/

[2] https://support.google.com/youtube/answer/2801939?hl=en

The manual identification of hate speech is almost impossible and needs to be thoroughly investigated automatically. A considerable amount of research work on English hate speech has been published. Davidson et al. [7] extracted N-gram TF-IDF features from tweets and applied logistic regression to classify each tweet into three classes hate, offensive, and neither. Kumari et al. [8] presented a model to identify cyberbullying instances using features optimization with genetic algorithm. Similarly, Agarwal and Sureka [9] extracted linguistic, semantic, and sentimental features and learned an ensemble classifier to detect racist contents. Kapil et al. [6] proposed LSTM and CNN based model to identify the hate speech in social media posts whereas, Badjatiya et al. [10] learned semantic word embedding to classify each tweet as racist, sexist, or neither. Kumari and Singh [11] presented a deep learning model to detect hate speech for English text. The code-mixed and script-mixed sentences are among the major challenges for machine learning models due to the unavailability of a sufficient dataset. A code mixed dataset on Tamil with English *code-mixed Tanglish* [12] and Malayalam with English *code-mixed Manglish* [13] were recently proposed for sentiment analysis task.

The purpose of this study is to recognize the hate speech in Indian languages like *code-mixed Tamil-English*, *code-mixed Malayalam-English*, and *script-mixed Malayalam-English* languages. The dataset used in the study belongs to *HASOC-Dravidian-CodeMix-FIRE2020 challenge* [14]. The data was gathered from YouTube and Twitter with a target for two sets of tasks. Task 1 asks to develop a classification system to differentiate *script-mixed Malayalam* comments into offensive and non-offensive. Task 2 requires to build a classifier to differentiate *Tanglish* and *Manglish* (Tamil and Malayalam have written using Roman Characters) into offensive and not-offensive classes. The current paper explored several deep learning and machine learning models to identify offensive posts from the code-mixed and script-mixed posts. For deep learning, we utilized attention-based Bi-LSTM-CNN, BERT, and DNN models. In contrast, for conventional machine learning, Support vector machine, Logistic regression, Random forest, and Naive Bayes classifiers are used.

The rest of the article is organized as follows; The proposed methodology is explained in Section 2. The experiment setting and obtained results are discussed in Section 3. Finally, we conclude the paper in Section 4.

2. Methodology

The detail description of the submitted model in the FIRE-2020 workshop is listed in section 2.1, whereas the details of extensive experiments with different character N-gram TF-IDF features with the different classifiers are listed in section 2.2. The detailed statistic of the datasets used in this study is listed in Table 1. While pre-processing of the texts, we kept & and @ in our dataset by translating it into *‘and’* and *’at’* respectively and removed other special characters. We also removed single letter word, punctuation, and replaced all numeric letters into their corresponding English word (e.g., 1-one, 9-nine). Finally, all texts are converted into the lowercase.

2.1. Model-1

A hybrid attention-based Bi-LSTM and CNN network is used in the case of Tamil and Malayalam code-mixed text. The overall model diagram of the hybrid attention-based Bi-LSTM and CNN network can be seen from Figure 1. We used character embedding for the CNN network, whereas word embedding is used in the attention-based Bi-LSTM network. For character embedding, one-hot encoding vectors are used. For word embedding, we created our FastText[3] word embedding by utilizing the language-

[3] https://fasttext.cc/

Table 1
Data statistic used in this study

Language	Class	Not-offensive	Offensive	Total
Malayalam code-mixed	Training	2047	1953	4000
	Testing	473	478	951
Tamil code-mixed	Training	2020	1980	4000
	Testing	465	475	940
Malayalam script-mixed	Training	2633	567	3200
	Development	328	72	400
	Testing	334	66	400

specific code-mixed Tamil and Malayalam text for Tamil and Malayalam models, respectively. We used skip-gram techniques and trained system for 10 epochs to create the FastText word embedding vectors. We fixed 200-characters for the input sequences in the case of CNN and 30-words for the input sequences in the case of attention based Bi-LSTM network. Finally, the character embedding matrix (200×70) and word embedding matrix (30×100) passes through the CNN and attention-based Bi-LSTM network, respectively. In CNN, 128 filters of 1-gram, 2-gram, 3-gram, and 4-gram are used at different layers of CNN. The output of the CNN layer is then passed through a dense layer having 128 neurons. To process word-embedding, two Bi-LSTM layers having 512 and 256 output dimension space are used, followed by an attention layer. Finally, the output of attention-based Bi-LSTM and CNN layer is concatenated and passes through a softmax layer to predict offensive and not-offensive text. The detailed working of the CNN and attention-based Bi-LSTM network can be seen in [15, 16, 17, 18].

The performance of deep neural networks is susceptible to hyper-parameters. Therefore, we performed extensive experiments by varying the learning rate, batch size, optimizer, epochs, loss function, and activation function. The proposed system worked best with the learning rate of 0.001, batch size of 32, Adam as an optimizer, epochs = 100, binary cross-entropy as a loss function, and ReLU activation in the internal layers of the network whereas softmax activation function at the output layer.

In the case of Malayalam script-mixed text, a fine-tuned pre-trained BERT[4] model is used to classify the text into offensive and not-offensive classes. We fixed 30-words for the text to input in the model and used a batch size of 32 and a learning rate of $2e^{-5}$ to fine-tune the pre-trained *bert-base-multilingual-uncased* BERT model. The detailed description of the BERT model can be seen in [19].

2.2. Model-2

Along with the submitted model in the FIRE-2020 workshop, we also explored the uses of different word and character N-gram TF-IDF features with the conventional machine learning classifiers and deep neural network. We experimented by extracting different combinations of 1-gram, 2-gram, 3-gram, 4-gram, 5-gram, and 6-gram word features and character features from the text and applied Support Vector Machine (SVM), Logistic Regression (LR), Naive Bayes (NB), Random Forest (RF), and Dense Neural Network (DNN). For the DNN network, we used four layers of a fully connected network with 1024, 256, 128, and 2-neurons. We used a dropout rate of 0.3, ReLU, and softmax as the activation function, batch size of 32, binary cross-entropy as the loss function, and Adam as the optimizer. Various word-level N-gram TF-IDF features with the said machine learning models did not perform well compared to the submitted model. Therefore we are not reporting the results of these

[4] https://huggingface.co/transformers/pretrained_models.html

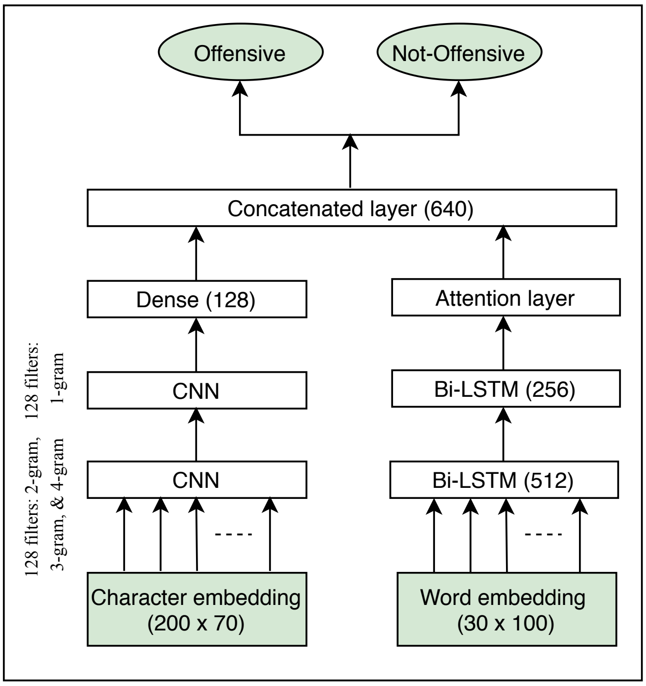

Figure 1: Proposed hybrid attention-based Bi-LSTM and CNN network

Table 2
Results for the attention-based Bi-LSTM-CNN and BERT models

Language	Model	Class	Precision	Recall	F_1-score
Tamil code-mixed	Attention-based Bi-LSTM-CNN	Offensive	0.85	0.83	0.84
		Not-offensive	0.83	0.85	0.84
		Weighted Avg.	0.84	0.84	0.84
Malayalam code-mixed	Attention-based Bi-LSTM-CNN	Offensive	0.71	0.71	0.71
		Not-offensive	0.71	0.71	0.71
		Weighted Avg.	0.71	0.71	0.71
Malayalam script-mixed	BERT	Offensive	0.95	0.97	0.96
		Not-offensive	0.83	0.74	0.78
		Weighted Avg.	0.93	0.93	0.93

models. On the other hand, when various machine learning models learned with different character N-gram TF-IDF features like 1-gram, 2-gram, 3-gram, 4-gram, 5-gram, and 6-gram, the performance was remarkable. In the case of code-mixed Tamil and Malayalam text, the top 10,000 characters N-gram (1-gram to 6-gram) TF-IDF features performed best. In the case of Malayalam script-mixed text, the top 20,000 characters N-gram (1-gram to 6-gram) TF-IDF features performed best. The detailed results of each of the classifiers are listed in section 3.

3. Results

The results of the attention-based Bi-LSTM-CNN and BERT models for Tamil code-mixed, Malayalam code-mixed, and Malayalam script-mixed text are listed in Table 2. In the case of Tamil code-mixed

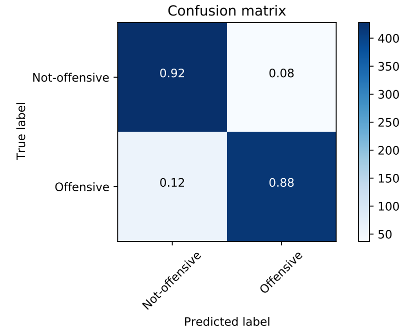

Figure 2: Confusion matrix for Naive Bayes (Tamil code-mixed)

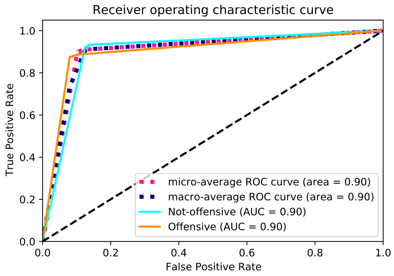

Figure 3: ROC for Naive Bayes (Tamil code-mixed)

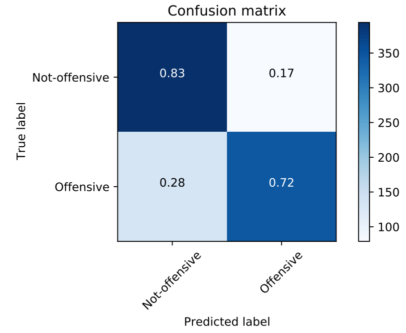

Figure 4: Confusion matrix for logistic regression classifier (Malayalam code-mixed)

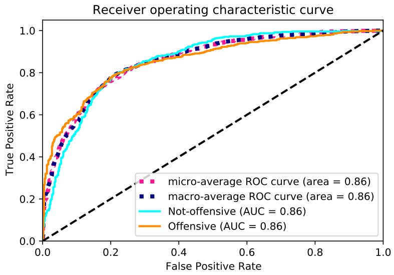

Figure 5: ROC for logistic regression classifier (Malayalam code-mixed)

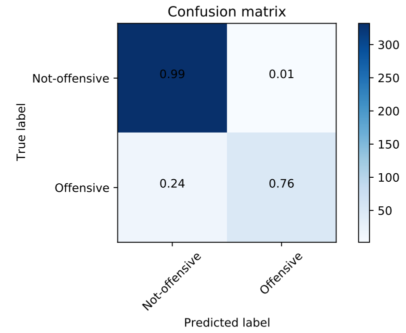

Figure 6: Confusion matrix for DNN model (Malayalam script-mixed)

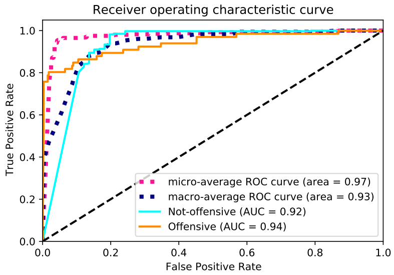

Figure 7: ROC for DNN model (Malayalam script-mixed)

text, the attention-based Bi-LSTM-CNN model achieved a weighted precision, recall, and F_1-score of 0.84. In the case of Malayalam code-mixed text, the attention-based Bi-LSTM-CNN model achieved a weighted precision, recall, and F_1-score of 0.71. In Malayalam script-mixed text, the BERT model achieved a precision, recall, and F_1-score of 0.93.

Next, experiments were performed for machine learning models using character N-gram (1 to 6-gram) TF-IDF features. The results for the Support Vector Machine (SVM), Logistic Regression (LR), Naive Bayes (NB), Random Forest (RB), and Dense Neural Network (DNN) are listed in Table 3. In the case of Tamil code-mixed text, the NB classifier performed best and achieved a precision, recall,

Table 3
Results for the different classifiers with character N-gram TF-IDF feature

		Tamil (Code-mixed)			Malayalam (Code-mixed)			Malayalam (Script-mixed)		
	Class	Precision	Recall	F_1-score	Precision	Recall	F_1-score	Precision	Recall	F_1-score
SVM	Offensive	0.87	0.90	0.88	0.83	0.69	0.75	0.97	0.56	0.71
	Not-offensive	0.89	0.86	0.88	0.73	0.86	0.79	0.92	1.00	0.96
	Weighted Avg.	0.88	0.88	0.88	0.78	0.77	0.77	0.93	0.93	0.92
LR	Offensive	0.88	0.89	0.89	0.81	0.72	0.77	0.91	0.30	0.45
	Not-offensive	0.89	0.88	0.88	0.75	0.83	0.79	0.88	0.99	0.93
	Weighted Avg.	0.89	0.89	0.89	**0.78**	**0.78**	**0.78**	0.88	0.88	0.85
NB	Offensive	0.92	0.88	0.90	0.79	0.63	0.70	0.49	0.73	0.59
	Not-offensive	0.88	0.92	0.90	0.69	0.83	0.75	0.94	0.85	0.89
	Weighted Avg.	**0.90**	**0.90**	**0.90**	0.74	0.73	0.73	0.87	0.83	0.84
RF	Offensive	0.85	0.90	0.88	0.78	0.70	0.74	0.96	0.71	0.82
	Not-offensive	0.89	0.84	0.87	0.72	0.81	0.76	0.95	0.99	0.97
	Weighted Avg.	0.87	0.87	0.87	0.75	0.75	0.75	0.95	0.95	0.94
DNN	Offensive	0.87	0.91	0.89	0.77	0.78	0.78	0.96	0.76	0.85
	Not-offensive	0.91	0.86	0.88	0.78	0.76	0.77	0.95	0.99	0.97
	Weighted Avg.	0.89	0.89	0.88	0.77	0.77	0.77	**0.96**	**0.95**	**0.95**

and F_1-score of 0.90. In the case of Malayalam code-mixed text, the LR classifier performed best with the precision, recall, and F_1-score of 0.78. In Malayalam script-mixed text, DNN performed best with a precision of 0.96, recall of 0.95, and F_1 of 0.95. The confusion matrix and ROC curve for the Tamil code-mixed, Malayalam code-mixed, and Malayalam script-mixed posts are shown in Figures 2 and 3, 4 and 5, and 6 and 7, respectively.

Among all the submitted model in the FIRE-2020 workshop for this task, the best model achieved precision, recall, and F_1-score of 0.90 for Tamil code-mixed text, precision, recall, and F_1-score of 0.78 for Malayalam code-mixed text and precision, recall, and F_1-score of 0.95 for Malayalam script-mixed text. Compared to the best-submitted model, our proposed system equally performed well in Tamil and Malayalam's code-mixed text. In the case of Malayalam script-mixed text, our proposed system performed slightly better with the precision of 0.96.

4. Conclusion

The identification of hate speech from the code-mixed and script-mixed Dravidian sentences have enormous challenges. This work explored the usability of several deep learning and machine learning-based models for classifying offensive and not-offensive sentences. The character N-gram TF-IDF based Naive Bayes classifier performed best with the weighted precision, recall, and F_1-score of 0.90 for Tamil code-mixed text. The Logistic regression classifier with character N-gram TF-IDF features performed best with the weighted precision, recall, and F_1-score of 0.78 for Malayalam code-mixed text. The Dense Neural Network with character N-gram TF-IDF features performed best with the weighted precision of 0.96, recall of 0.95, and F_1-score of 0.95 for Malayalam script-mixed text.

References

[1] S. Saumya, J. P. Singh, Detection of spam reviews: A sentiment analysis approach, Csi Transactions on ICT 6 (2018) 137–148.

[2] S. Saumya, J. P. Singh, et al., Spam review detection using lstm autoencoder: an unsupervised approach, Electronic Commerce Research (2020) 1–21.

[3] A. Kumar, J. P. Singh, S. Saumya, A comparative analysis of machine learning techniques for disaster-related tweet classification, in: 2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129), IEEE, 2019, pp. 222–227.

[4] A. Kumar, N. C. Rathore, Relationship strength based access control in online social networks, in: Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 2, Springer, 2016, pp. 197–206.

[5] K. Kumari, J. P. Singh, Y. K. Dwivedi, N. P. Rana, Towards cyberbullying-free social media in smart cities: a unified multi-modal approach, Soft Computing (2020) 11059–11070.

[6] P. Kapil, A. Ekbal, D. Das, Investigating deep learning approaches for hate speech detection in social media, arXiv preprint arXiv:2005.14690 (2020).

[7] T. Davidson, D. Warmsley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, arXiv preprint arXiv:1703.04009 (2017).

[8] K. Kumari, J. P. Singh, Identification of cyberbullying on multi-modal social media posts using genetic algorithm, Transactions on Emerging Telecommunications Technologies (2020) e3907. doi:10.1002/ett.3907.

[9] S. Agarwal, A. Sureka, Characterizing linguistic attributes for automatic classification of intent based racist/radicalized posts on tumblr micro-blogging website, arXiv preprint arXiv:1701.04931 (2017).

[10] P. Badjatiya, S. Gupta, M. Gupta, V. Varma, Deep learning for hate speech detection in tweets, in: Proceedings of the 26th International Conference on WWW Companion, 2017, pp. 759–760.

[11] K. Kumari, J. P. Singh, Ai_ml_nit patna at hasoc 2019: Deep learning approach for identification of abusive content, in: Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation (December 2019), 2019, pp. 328–335.

[12] B. R. Chakravarthi, V. Muralidaran, R. Priyadharshini, J. P. McCrae, Corpus creation for sentiment analysis in code-mixed Tamil-English text, in: Proceedings of the 1st Joint Workshop on SLTU and CCURL, ELRA, Marseille, France, 2020, pp. 202–210.

[13] B. R. Chakravarthi, N. Jose, S. Suryawanshi, E. Sherly, J. P. McCrae, A sentiment analysis dataset for code-mixed Malayalam-English, in: Proceedings of the 1st Joint Workshop on SLTU and CCURL, ELRA, Marseille, France, 2020, pp. 177–184.

[14] B. R. Chakravarthi, M. A. Kumar, J. P. McCrae, P. B, S. KP, T. Mandl, Overview of the track on "hasoc-offensive language identification- DravidianCodeMix", in: Proceedings of FIRE, 2020.

[15] A. Kumar, J. P. Singh, Location reference identification from tweets during emergencies: A deep learning approach, International journal of disaster risk reduction 33 (2019) 365–375.

[16] A. Kumar, J. P. Singh, Y. K. Dwivedi, N. P. Rana, A deep multi-modal neural network for informative twitter content classification during emergencies, Annals of Operations Research (2020) 1–32.

[17] J. P. Singh, A. Kumar, N. P. Rana, Y. K. Dwivedi, Attention-based lstm network for rumor veracity estimation of tweets, Information Systems Frontiers (2020) 1–16.

[18] S. Saumya, J. P. Singh, Y. K. Dwivedi, Predicting the helpfulness score of online reviews using convolutional neural network, Soft Computing (2019) 1–17.

[19] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, arXiv preprint arXiv:1910.01108 (2019).