Offensive language identification in Dravidian code mixed social media text

Sunil Saumya[1], **Abhinav Kumar**[2] and **Jyoti Prakash Singh**[2]
[1]Indian Institute of Information Technology Dharwad, Karnataka, India
[2]National Institute of Technology Patna, Bihar, India
sunil.saumya@iiitdwd.ac.in,
abhinavanand05@gmail.com, jps@nitp.ac.in

Abstract

Hate speech and offensive language recognition in social media platforms have been an active field of research over recent years. In non-native English spoken countries, social media texts are mostly in code mixed or script mixed/switched form. The current study presents extensive experiments using multiple machine learning, deep learning, and transfer learning models to detect offensive content on Twitter. The data set used for this study are in Tanglish (Tamil and English), Manglish (Malayalam and English) code-mixed, and Malayalam script-mixed. The experimental results showed that 1 to 6-gram character TF-IDF features are better for the said task. The best performing models were naive bayes, logistic regression, and vanilla neural network for the dataset Tamil code-mix, Malayalam code-mixed, and Malayalam script-mixed, respectively instead of more popular transfer learning models such as BERT and ULMFiT and hybrid deep models.

1 Introduction

The hate speech is generally defined as any communication which humiliates or denigrates an individual or a group based on the characteristics such as colour, ethnicity, sexual orientation, nationality, race and religion. Due to huge volume of user-generated content on the web, particularly social networks such as Twitter, Facebook, and so on, the problem of detecting and probably restricting Hate Speech on these platforms has become a very critical issue (Del Vigna12 et al., 2017). Hate speech lasts forever on these social platforms compared to physical abuse and terribly affects the individual on the mental status creating depression, sleeplessness and even suicide (Ullmann and Tomalin, 2020).

Owing to the high frequency of posts, detecting hate speech on social media manually is almost impossible. Some recent researches have indicated that the automation of hate speech detection is a more reliable solution. (Davidson et al., 2017) extracted N-gram TF-IDF features from tweets using logistical regression to classify each tweet in hate, offensive and non-offensive classes. Another model for the detection of the cyberbullying instances was presented by (Kumari and Singh, 2020) with a genetic algorithm to optimize the distinguishing features of multimodal posts. (Agarwal and Sureka, 2017) used the linguistic, semantic and sentimental feature to detect racial content. The LSTM and CNN based model for recognising hate speech in the social media posts were explored by (Kapil et al., 2020). (Badjatiya et al., 2017) exploited the semantic word embedding to classify each tweet into racist, sexist and neither class. Another deep learning model for the detection of hate speech was proposed by (Paul et al., 2020). However, most of the works for hate speech detection were validated with English datasets only.

In a country such as India, the majority of people in social media use at least two languages, primarily English and Hindi (or say Hinglish). These texts are considered bilingual. In a bilingual setting, the script of the entire post may be same with words coming from both of these languages termed as mixed-code (or code mix) text. A few popular code mixed posts in India are English and Hindi (or say Hinglish), Tanglish (Tamil and English) (Chakravarthi et al., 2020c), Manglish (Malayalam and English) (Chakravarthi et al., 2020a), Kanglish (Kannada and English) (Hande et al., 2020), and so on. The Tamil language is one of the world's longest-enduring traditional languages, with a set of experiences tracing all the way back to 600 BCE. Tamil writing is overwhelmed by verse, particularly Sangam writing, which is made out of sonnets formed between 600 BCE and 300 CE. The main Tamil creator was the writer and thinker Thiruvalluvar, who composed the Tirukkua, a gathering of

Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pages 36–45
April 20, 2021 ©2021 Association for Computational Linguistics

compositions on morals, legislative issues, love and ethical quality broadly thought to be the best work of Tamil writing. Tamil has the oldest extant literature among Dravidian languages. All Dravidian languages evolved from classical Tamil language (Thavareesan and Mahesan, 2019, 2020a,b). Even though they have their own scripts still in the Internet code-mixing comments can be found in these languages (Chakravarthi, 2020b). Identifying hate content in such bilingual or code mixed language is a very challenging task (Jose et al., 2020; Priyadharshini et al., 2020; Chakravarthi, 2020a). An automatic model which is trained in a monolingual context to detect hate posts may not yield the same result when tested bilingually or with a code-mix (Puranik et al., 2021; Hegde et al., 2021; Yasaswini et al., 2021; Ghanghor et al., 2021b,a). This is because each system learns and recognises words in the given vocabulary. When a new word is encountered, which is not in the vocabulary, it is marked as an undefined token that makes no difference in the estimation of the model. Therefore, when checked with the language in other scripts the model's performance decreases.

The current study identifies the hate content in Tanglish, Manglish and Malayalam script mixed in tweets and validated with the dataset provided in *HASOC-Dravidian-CodeMix-FIRE2020 challenge* (Chakravarthi et al., 2020b). The dataset proposed in the challenge was collected from Twitter. A variety of deep learning models have been examined in the current paper to distinguish offensive posts from script-mixed posts. Along with that we also examined a few transfer learning models like BERT (Devlin et al., 2018a) and ULMFit (Howard and Ruder, 2018) for the classification task.

The rest of the article is summarized as follows: Section 2 presents the overview of articles proposed in the domain of hate or offensive speech. The task and dataset description is explained in Section 3. This followed by the explanation of the proposed methodology in Section 4. The experimental results and discussion are explained in Section 5 and 6. The paper concludes by highlighting the main findings in Section 7.

2 Related works

The hate speech identification in social media texts suffers from many challenges; such as code-mixed social media content, script-mixed social media contents, and so on. This section sheds light on a few state-of-art techniques presented to handle such issues.

Most of the analysis proposed for the detection of hate contents were validated with monolingual datasets. It is relatively easy to build a monolingual model, since (i) it is readily accessible, (ii) it learns a single language dictionary, and (iii) unknown token frequency is lower in the test data. (Davidson et al., 2017) worked on 25000 tweets in English and reported that tweets that contained racism and homophobic contexts were hate speech and tweets that contained sexism contexts were offensive contents. Other work, on English data, was proposed by (Waseem and Hovy, 2016) where n-gram features were extracted for identifying sexiest, racism, and none class.

Apart from this, some works were reported on a multilingual dataset where scripts of two or more languages are mixed. (Kumar et al., 2018) proposed a model for multi-lingual datasets containing aggressive and non-aggressive comments in English as well as Hindi from Facebook and Twitter. (Samghabadi et al., 2018) used ensemble learning based on various machine learning classifiers such as Logistic Regression, SVM with word n-gram, character n-gram, word embedding, and sentiment as a feature set. They found that combined words and character n-gram features performed better than an individual feature. (Srivastava et al., 2018) identified online social aggression on the Facebook comment in a multilingual scenario and Wikipedia toxic comments using stacked LSTM units followed by convolution layer and fastText as word representation. They achieved 0.98 AUC for Wikipedia toxic comment classification and a weighted F1 score of 0.63 for the Facebook test set and 0.59 for the Twitter test set. (Mandl et al., 2020; Chakravarthi et al., 2020d) presented several models and their results for English, Hindi, and German datasets. They reported the best model as long short term memory-based network that could capture the multilingual context in a better way. Bohra et al. (2018) extended the earlier research of hate speech detection for code mixed tweets of Hindi and English. (Kumari et al., 2021) presented a Convolutional Neural Network (CNN) and Binary Particle Swarm Optimization (BPSO) based model to classify multimodal posts with images and text into non-aggressive, medium-aggressive and high-aggressive classes. Another multilingual context could be code-mixed where two languages are written in a single script. For example, (Chakravarthi et al., 2021b; Chakravarthi and Muralidaran, 2021; Chakravarthi et al., 2021a; Suryawanshi and Chakravarthi, 2021) proposed a code mixed Dravidian data in Tamil, Malayalam, and Kannada. (Bohra et al., 2018) developed a Hinglish dataset from Twitter. They reported preliminary experiment results of Support Vector Machine (SVM) and Random Forest (RF) classifiers with n-grams and lexicon-based features with an accuracy of 0.71.

Research in hate and offensive language, as described above, is mainly conducted in a monolingual setting. The paper aims to propose a machine learning system for the code-mixed and script-mixed dataset to identify hate contents.

3 Task and Data description

The current study performs two tasks; (i) Task 1 includes the development of an offensive and non-offensive classification system for distinguishing *script-mixed Malayalam* comments, and (ii) 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. Table 1 shows the overview of the data set used in this analysis. As can be seen in Table 1, there are three sets of data, out of which in the first sets, Malayalam code-mixed and Tamil code-mixed, the posts were written in a single script English, but in the last set, the posts were written in two different scripts (Malayalam script-mixed).

4 Methodology

Three different models were developed to identify hate or offensive contents in Dravidian posts; (i) conventional learning based models, (ii) neural network-based models, and (iii) transfer learning-based models. In this section, we explain the working of each model in detail. A detailed diagram for presented models is shown in Figure 1. The results of the models are explained in Section 5.

4.1 Conventional learning based models

In conventional machine learning-based classifications, the current study explored the use of different N-gram TF-IDF word and character features. In the case of character, 1 to 6 gram character TF-IDF features were used, whereas, in case of a word, 1 to 3 gram word TF-IDF features were used. The extracted features were fed to classifiers like Support Vector Machine (SVM), Logistic Regression (LR), Naive Bayes (NB), Random Forest (RF). The detailed performance report of word n-grams and character n-grams are shown in Section 5.

4.2 Neural learning-based models

Initially, the character n-grams TF-IDF features (1-6 grams) extracted in previous Section 4.1 were used as an input to a vanilla neural network (VNN) model. For the vanilla neural network, four fully connected layers were sequenced, having 1024, 256, 128, and 2 neurons in first, second, third and fourth layer, respectively. We kept two neurons in the final layer (or output layers) to identify each input in offensive groups. Based on the probabilities of softmax activation with output neurons, the last class was determined. In the intermediate layers, the activation function was ReLu. The proposed vanilla neural network was trained with cross-entropy loss function and Adam optimizer. The training dropout was 0.3 and the batch size was 32.

Consequently, other deep learning models for offensive groups prediction were also developed. A hybrid attention-based Bi-LSTM and CNN network was built as shown in Figure 1. The detailed working of the CNN and attention-based Bi-LSTM network for text classification can be seen in (Jang et al., 2020; Xu et al., 2020; Saumya et al., 2019). To CNN, character embedding was the input, whereas to Bi-LSTM, word embedding was the input. To prepare the character embedding, a one-hot vector representation of characters were used. Every input was padded with a maximum of 200 characters with repetition. The total unique character found in the vocabulary was 70. Therefore, a (200×70) dimensional embedding matrix was given as an input to CNN. To extract the features from the convolution layer, 128 different filters for each 1-gram, 2-gram, 3-gram, and 4-gram were used. The output of the first convolution layer was fed to the second convolution layer with similar filter dimensions. The features extracted from the CNN layers were then represented in a vector having 128 features using a dense layer.

To prepare the word embedding input for Bi-LSTM was we used FastText[1] utilizing the language-specific code-mixed Tamil and Malayalam text for Tamil and Malayalam models, respec-

[1] 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

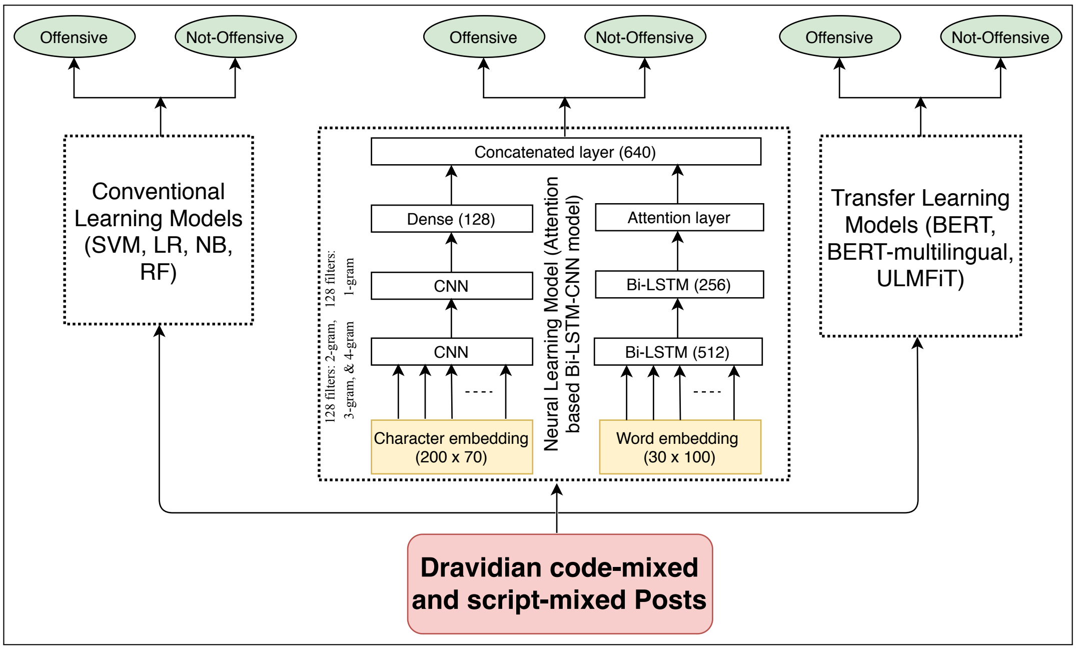
Offensive
Not-Offensive
Offensive
Not-Offensive
Offensive
Not-Offensive
Conventional Learning Models (SVM, LR, NB, RF)
Concatenated layer (640)
Dense (128)
CNN
CNN
128 filters: 1-gram
128 filters: 2-gram, 3-gram, & 4-gram
Character embedding (200 x 70)
Neural Learning Model (Attention based Bi-LSTM-CNN model)
Attention layer
Bi-LSTM (256)
Bi-LSTM (512)
Word embedding (30 x 100)
Transfer Learning Models (BERT, BERT-multilingual, ULMFiT)
Dravidian code-mixed and script-mixed Posts

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

tively. The skip-gram architecture was trained for ten epochs to extract the FastText embedding vectors. A maximum of 30 words embedding vectors was given input to the network in a time stamp manner. Every word was represented in a 100-dimensional vector which was extracted from the embedding layer. Finally, a (30×100) dimensional matrix input was given to 2-layered stacked Bi-LSTM layer, 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.

Hyperparameters tuning was done to check the performance of the proposed deep-neural model. We conducted comprehensive experiments by adjusting the learning rate, batch size, optimizer, epoch, loss function and activation function. The system performance was best with the learning rate 0.001, batch size 32, Adam optimizer, epochs 100, loss function as binary cross-entropy, and ReLU activation within the internal layers of the network. At the output layer, the activation was softmax.

4.3 Transfer models

The current study used two different transfer models, BERT (Bidirectional Encoder Representations from Transformers) and ULMFiT (Universal Language Model Fine-tuning for Text Classification) to accomplish the given objectives.

Two different variations of BERT model[2] (Devlin et al., 2018b) is used in the current study; (i) BERT base (*bert-base-uncased*), and (ii) BERT multilingual (*bert-base-multilingual-uncased*). The BERT base model is trained for English language using a masked modelling technique. Whereas, BERT multilingual is trained for 102 languages with masked language modelling. We used

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

ktrain[3] libraries to develop the BERT based models. Both BERT variations are uncased that means it does not make a difference between a word written in upper case lower case. In training BERT-models, 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 model. The detailed description of the BERT model can be seen in (Sanh et al., 2019). The other transfer model used was ULMFiT. It can be applied to any task in NLP. To train ULMFiT model, we used fastai library[4]. The input and hyper-parameters were the same as we used in BERT.

5 Result

This section presents the experimental results of all three aforementioned models explained in Section 4. The results are presented in terms of precision, recall, and F_1-score of class offensive and not-offensive. The weighted average of both classes is also presented. A particular model is identified as best if it has reported the highest weighted average of precision, recall, and F_1-score. The value in bold represents the highest value for a particular dataset.

The convectional learning experiments were performed using character N-gram (1 to 6-gram) TF-IDF features. The results are shown in the Table 2 for SVM, LR, NB, RF. In the case of Tamil code-mixed text, the NB classifier performed best and achieved a precision, recall, 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. Similarly, in the case of Malayalam script-mixed RF classifier reported better performance having precision, recall of 0.95, and F_1-score of 0.94. Similar experiments were done for word TF-IDF features for 1 to 3 N-grams. The results are shown in the Table 3.

The results of the proposed neural-based models for Tamil code-mixed, Malayalam code-mixed, and Malayalam script-mixed text are listed in Table 4. As can be seen from the Table 4, the vanilla neural network (VNN) model outperformed attention-based Bi-LSTM-CNN for all three datasets. For Tamil code-mixed, VNN reported precision, recall, and F_1-score of 0.89 and for Malayalam code-mixed it reported precision, recall, and F_1-score of 0.77. Similarly, for Malayalam script-mixed data, the proposed vanilla neural network reported a precision, recall, and F_1 score of 0.95.

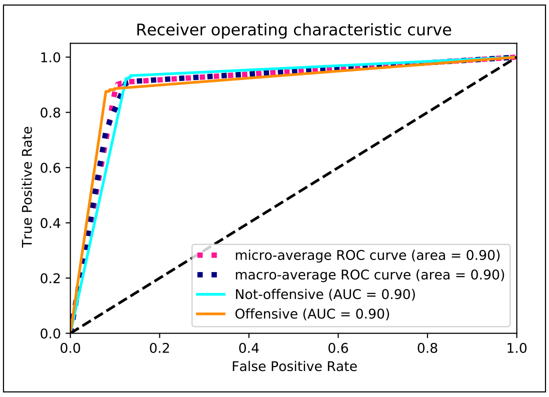

Figure 2: ROC for Naive Bayes
(Tamil code-mixed)

Finally, the experimental results of transfer models are shown in Table 5. The table shows the results of three transfer models BERT, BERT-multilingual, and ULMFiT. In Malayalam script-mixed text, the BERT-multilingual model achieved the highest precision, recall, and F_1-score of 0.93. Even, for Tamil code-mixed text, BERT-multilingual performed better than others with precision, recall, and F_1-score of 0.86. The results of the BERT model was also comparable with precision, recall, and F_1-score of 0.89, 0.84, and 0.86. But, for Malayalam code-mixed BERT performance was highest with precision, recall, and F_1-score of 0.76.

6 Result Comparison and Discussion

Of all the experimental models, conventional learning models with character 1 to 6 gram TF-IDF features showed the best output for the two datasets, Tamil code-mixed and Malayalam code-mixed. For Tamil code-mixed, the best performance was reported by NB model with precision, recall, and F_1-score of 0.90. Similarly, for Malayalam code-mixed the LR model reported best with precision, recall, and F_1-score of 0.78. However, for the Malayalam Script mixed, Vanilla Neural Network (VNN) reported best results having precision, recall, and F_1-score of 0.95, 0.95, and 0.95 respectively. The receiver operating characteristics (ROC) area under curve for all three best models are shown in Figures 2, 3, and 4.

The outcome of this comprehensive study was surprising, given that the performance of all complex models, such as the BiLSTM-CNN hybrid model and Transfer models, was relatively low, but

[3] https://github.com/amaiya/ktrain
[4] https://nlp.fast.ai/

Table 2: Results for the different classifiers with character 1 to 6-gram TF-IDF feature

Models	Class	Tamil (Code-mixed)			Malayalam (Code-mixed)			Malayalam (Script-mixed)		
		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**

Table 3: Results for the different classifiers with word 1 to 3-gram TF-IDF feature

Models	Class	Tamil (Code-mixed)			Malayalam (Code-mixed)			Malayalam (Script-mixed)		
		Precision	Recall	F_1-score	Precision	Recall	F_1-score	Precision	Recall	F_1-score
SVM	Offensive	0.64	0.94	0.76	0.75	0.55	0.63	1.00	0.53	0.69
	Not-offensive	0.88	0.46	0.60	0.64	0.81	0.72	0.92	1.00	0.96
	Weighted Avg.	0.76	0.70	0.68	0.69	0.68	0.67	0.93	0.92	0.91
LR	Offensive	0.88	0.86	0.87	0.75	0.68	0.71	0.91	0.30	0.45
	Not-offensive	0.86	0.88	0.87	0.70	0.77	0.74	0.88	0.99	0.93
	Weighted Avg.	**0.87**	**0.87**	**0.87**	**0.73**	**0.73**	**0.73**	0.88	0.88	0.85
NB	Offensive	0.67	0.82	0.74	0.68	0.62	0.65	0.47	0.83	0.60
	Not-offensive	0.76	0.59	0.67	0.65	0.71	0.68	0.96	0.81	0.88
	Weighted Avg.	0.72	0.71	0.70	0.67	0.66	0.66	0.88	0.81	0.83
RF	Offensive	0.78	0.89	0.83	0.70	0.66	0.68	0.94	0.67	0.78
	Not-offensive	0.87	0.75	0.80	0.67	0.71	0.69	0.94	0.99	0.96
	Weighted Avg.	0.83	0.82	0.82	0.69	0.68	0.68	**0.94**	**0.94**	**0.93**

Table 4: Results for the VNN and attention-based BiLSTM-CNN models

Models	Class	Tamil (code-mixed)			Malayalam (Code-mixed)			Malayalam (script-mixed)		
		Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Vanilla NN	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	0.76	0.95	0.99	0.97
	Weighted-Avg	**0.89**	**0.89**	**0.89**	**0.77**	**0.77**	**0.77**	**0.95**	**0.95**	**0.95**
Attention-based BiLSTM-CNN	Offensive	0.85	0.83	0.84	0.71	0.71	0.71	0.89	0.68	0.77
	Not-offensive	0.83	0.85	0.84	0.71	0.71	0.71	0.93	0.98	0.96
	Weighted-Avg	0.84	0.84	0.84	0.71	0.71	0.71	0.93	0.93	0.92

Table 5: Results transfer models BERT and ULMFiT

Models	Class	Tamil (code-mixed)			Malayalam (Code-mixed)			Malayalam (script-mixed)		
		Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
BERT	Offensive	0.93	0.77	0.84	0.71	0.79	0.75	0.73	0.73	0.73
	Not-offensive	0.85	0.92	0.88	0.81	0.73	0.77	0.73	0.73	0.73
	Weighted-Avg	0.89	0.84	0.86	**0.76**	**0.76**	**0.76**	0.73	0.73	0.73
BERT	Offensive	0.85	0.87	0.86	0.75	0.68	0.72	0.95	0.97	0.96
	Not-offensive	0.86	0.85	0.86	0.71	0.77	0.74	0.83	0.74	0.78
Muiltilingual	Weighted-Avg	**0.86**	**0.86**	**0.86**	0.73	0.73	0.73	**0.93**	**0.93**	**0.93**
ULMFit	Offensive	0.72	0.63	0.67	0.30	0.51	0.38	0.57	0.68	0.62
	Not-offensive	0.56	0.57	0.57	0.71	0.50	0.59	0.72	0.56	0.63
	Weighted-Avg	0.65	0.60	0.62	0.59	0.50	0.52	0.66	0.61	0.63

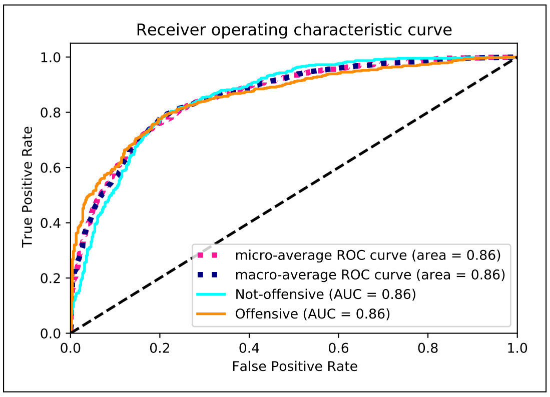

Figure 3: ROC for Logistic Regression (Malayalam code-mixed)

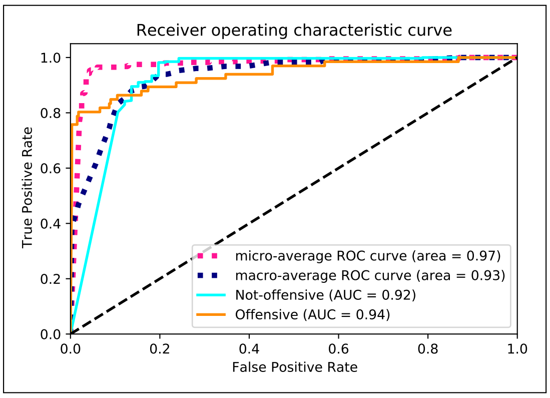

Figure 4: ROC for vanilla Neural Network (Malayalam script-mixed)

for many NLP tasks, such as text classification or language modelling, it was proven better.

The results indicate that the character n-gram TF-IDF features play a very important role for code-mixed and code-script data. Secondly, in the same sense, the performance of the transfer models is not encouraging. BERT, which is trained in the English language, treats most of the token of code-mixed and code-script data as an unknown token which could affect the model performance. The BERT-multilingual which is trained on 102 languages identifies the language of input text first and loads its vocabulary then. Even, for code-mixed and code-script data BERT-multilingual identified a single language and was subsequently processed. In effect, the overall model performance was reduced. Moreover, it was found that the language identified by BERT-multilingual for code-mixed and the code-script dataset was different for different runs. Consequently, the results fluctuated even.

7 Conclusion

Hate speech identification in code-mixed and script-mixed context is one of the most challenging tasks in NLP. The current study presented extensive experiments utilizing various conventional learning, deep learning, and transfer learning models. Three datasets used in the study were Tamil code-mixed, Malayalam code-mixed, and Malayalam script-mixed. The results reported by all models clearly show that conventional learning models along with vanilla neural model outperformed other complex deep learning, and transfer learning models. 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 Vanilla Neural Network with character N-gram TF-IDF features performed best with the weighted precision of 0.95, recall of 0.95, and F_1-score of 0.95 for Malayalam script-mixed text.

References

Swati Agarwal and Ashish Sureka. 2017. Characterizing linguistic attributes for automatic classification of intent based racist/radicalized posts on tumblr micro-blogging website. *arXiv preprint arXiv:1701.04931*.

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on WWW Companion*, pages 759–760.

Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A Dataset of Hindi-English Code-Mixed Social Media Text for Hate Speech Detection. In *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*, pages 36–41.

Bharathi Raja Chakravarthi. 2020a. HopeEDI: A multilingual hope speech detection dataset for equality, diversity, and inclusion. In *Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media*, pages 41–53, Barcelona, Spain (Online). Association for Computational Linguistics.

Bharathi Raja Chakravarthi. 2020b. *Leveraging orthographic information to improve machine translation of under-resourced languages*. Ph.D. thesis, NUI Galway.

Bharathi Raja Chakravarthi, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John Philip McCrae. 2020a. A sentiment analysis dataset for code-mixed Malayalam-English. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 177–184, Marseille, France. European Language Resources association.

Bharathi Raja Chakravarthi, M Anand Kumar, John Philip McCrae, Premjith B, Soman KP, and Thomas Mandl. 2020b. Overview of the track on HASOC-Offensive Language Identification-DravidianCodeMix. In *Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2020). CEUR Workshop Proceedings. In: CEUR-WS. org, Hyderabad, India*.

Bharathi Raja Chakravarthi and Vigneshwaran Muralidaran. 2021. Findings of the shared task on Hope Speech Detection for Equality, Diversity, and Inclusion. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*. Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John Philip McCrae. 2020c. Corpus creation for sentiment analysis in code-mixed Tamil-English text. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 202–210, Marseille, France. European Language Resources association.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Shubhanker Banerjee, Richard Saldhana, John Philip McCrae, Anand Kumar M, Parameswari Krishnamurthy, and Melvin Johnson. 2021a. Findings of the shared task on Machine Translation in Dravidian languages. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Navya Jose, Anand Kumar M, Thomas Mandl, Prasanna Kumar Kumaresan, Rahul Ponnusamy, Hariharan V, Elizabeth Sherly, and John Philip McCrae. 2021b. Findings of the shared task on Offensive Language Identification in Tamil, Malayalam, and Kannada. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Shardul Suryawanshi, Navya Jose, Elizabeth Sherly, and John P. McCrae. 2020d. Overview of the Track on Sentiment Analysis for Dravidian Languages in Code-Mixed Text. In *Forum for Information Retrieval Evaluation*, FIRE 2020, page 21–24, New York, NY, USA. Association for Computing Machinery.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *arXiv preprint arXiv:1703.04009*.

Fabio Del Vigna12, Andrea Cimino23, Felice Dell'Orletta, Marinella Petrocchi, and Maurizio Tesconi. 2017. Hate me, hate me not: Hate speech detection on facebook. In *Proceedings of the First Italian Conference on Cybersecurity (ITASEC17)*, pages 86–95.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018a. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018b. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Nikhil Kumar Ghanghor, Parameswari Krishnamurthy, Sajeetha Thavareesan, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. 2021a. 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*, Online. Association for Computational Linguistics.

Nikhil Kumar Ghanghor, Rahul Ponnusamy, Prasanna Kumar Kumaresan, Ruba Priyadharshini, Sajeetha Thavareesan, and Bharathi Raja

Chakravarthi. 2021b. IIITK@LT-EDI-EACL2021: Hope Speech Detection for Equality, Diversity, and Inclusion in Tamil, Malayalam and English. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, Online.

Adeep Hande, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. 2020. 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*, pages 54–63, Barcelona, Spain (Online). Association for Computational Linguistics.

Siddhanth U Hegde, Adeep Hande, Ruba Priyadharshini, Sajeetha Thavareesan, and Bharathi Raja Chakravarthi. 2021. UVCE-IIITT@DravidianLangTech-EACL2021: Tamil Troll Meme Classification: You need to Pay more Attention. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Beakcheol Jang, Myeonghwi Kim, Gaspard Harerimana, Sang-ug Kang, and Jong Wook Kim. 2020. Bi-lstm model to increase accuracy in text classification: combining word2vec cnn and attention mechanism. *Applied Sciences*, 10(17):5841.

Navya Jose, Bharathi Raja Chakravarthi, Shardul Suryawanshi, Elizabeth Sherly, and John P. McCrae. 2020. A Survey of Current Datasets for Code-Switching Research. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 136–141.

Prashant Kapil, Asif Ekbal, and Dipankar Das. 2020. Investigating deep learning approaches for hate speech detection in social media. *arXiv preprint arXiv:2005.14690*.

Ritesh Kumar, Guggilla Bhanodai, Rajendra Pamula, and Maheshwar Reddy Chennuru. 2018. Trac-1 shared task on aggression identification: Iit (ism) @ coling'18. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 58–65.

Kirti Kumari and Jyoti Prakash Singh. 2020. Identification of cyberbullying on multi-modal social media posts using genetic algorithm. *Transactions on Emerging Telecommunications Technologies*, page e3907.

Kirti Kumari, Jyoti Prakash Singh, Yogesh K Dwivedi, and Nripendra P Rana. 2021. Multi-modal aggression identification using convolutional neural network and binary particle swarm optimization. *Future Generation Computer Systems*.

Thomas Mandl, Sandip Modha, Anand Kumar M, and Bharathi Raja Chakravarthi. 2020. 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*, FIRE 2020, page 29–32, New York, NY, USA. Association for Computing Machinery.

Sayanta Paul, Sriparna Saha, and Mohammed Hasanuzzaman. 2020. Identification of cyberbullying: A deep learning based multimodal approach. *Multimedia Tools and Applications*, pages 1–20.

Ruba Priyadharshini, Bharathi Raja Chakravarthi, Mani Vegupatti, and John P. McCrae. 2020. Named Entity Recognition for Code-Mixed Indian Corpus using Meta Embedding. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 68–72.

Karthik Puranik, Adeep Hande, Ruba Priyadharshini, Sajeetha Thavareesan, and Bharathi Raja Chakravarthi. 2021. IIITT@LT-EDI-EACL2021-Hope Speech Detection: There is always hope in Transformers. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*. Association for Computational Linguistics.

Niloofar Safi Samghabadi, Deepthi Mave, Sudipta Kar, and Thamar Solorio. 2018. Ritual-uh at trac 2018 shared task: Aggression identification. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 12–18.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sunil Saumya, Jyoti Prakash Singh, and Yogesh K Dwivedi. 2019. Predicting the helpfulness score of online reviews using convolutional neural network. *Soft Computing*, pages 1–17.

Saurabh Srivastava, Prerna Khurana, and Vartika Tewari. 2018. Identifying aggression and toxicity in comments using capsule network. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 98–105.

Shardul Suryawanshi and Bharathi Raja Chakravarthi. 2021. Findings of the shared task on Troll Meme Classification in Tamil. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2019. Sentiment Analysis in Tamil Texts: A Study on Machine Learning Techniques and Feature Representation. In *2019 14th Conference on Industrial and Information Systems (ICIIS)*, pages 320–325.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2020a. Sentiment Lexicon Expansion using Word2vec and fastText for Sentiment Prediction in Tamil texts. In *2020 Moratuwa Engineering Research Conference (MERCon)*, pages 272–276.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2020b. Word embedding-based Part of Speech tagging in Tamil texts. In *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, pages 478–482.

Stefanie Ullmann and Marcus Tomalin. 2020. Quarantining online hate speech: technical and ethical perspectives. *Ethics and Information Technology*, 22(1):69–80.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Jingyun Xu, Yi Cai, Xin Wu, Xue Lei, Qingbao Huang, Ho-fung Leung, and Qing Li. 2020. Incorporating context-relevant concepts into convolutional neural networks for short text classification. *Neurocomputing*, 386:42–53.

Konthala Yasaswini, Karthik Puranik, Adeep Hande, Ruba Priyadharshini, Sajeetha Thavareesan, and Bharathi Raja Chakravarthi. 2021. IIITT@DravidianLangTech-EACL2021: Transfer Learning for Offensive Language Detection in Dravidian Languages. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.