Overview of the track on HASOC-Offensive Language Identification-DravidianCodeMix

Bharathi Raja Chakravarthi[a], Anand Kumar M[b], John P. McCrae[a], B Premjith[c], K.P Soman[c] and Thomas Mandl[d]

[a]Insight SFI Research Centre for Data Analytics, Data Science Institute, National University of Ireland, Galway.
[b]Department of Information Technology, National Institute of Technology Karnataka Surathkal, India
[c]Center for Computational Engineering and Networking (CEN), Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India
[d]Information Science, University of Hildesheim, Germany

Abstract

We present the results and main findings of the HASOC-Offensive Language Identification on code mixed Dravidian languages. The task featured two tasks. Task 1 is about offensive language identification in Malayalam language where the comment were written in both native script and Latin script. Task 2 is about offensive language identification in Tamil and Malayalam languages where the comments were written in Latin script (non-native script). For both the task, given a comment the participants should develop a system to classify the text into offensive or not-offensive. In total 96 participants participated and 12 participants submitted the papers. In this paper, we present the task, data, the results and discuss the system submission and methods used by participants.

Keywords

Dravidian languages, Tamil, Malayalam, Offensive language identification

1. Introduction

The impact social media has in our society is unquestionable. The freedom of expressing one's opinion in social media leads to the misuse of the platform by many. Absence of moderation rules imposed by the social media contributes to the unhealthy usage of such platforms. Eventually, the number of offensive comments on social media increased exponentially. This attracts many researchers to find an appropriate way to filter out such comments from social media. Identification of offensive contents is a challenging task, especially when the data is code-mixed.

There is an increasing demand for offensive language detection on social media texts which are largely code-mixed. Code-mixing is a prevalent phenomenon in a multilingual community and the code-mixed texts are sometimes written in non-native scripts [1]. People find it easier to communicate by mixing two or more languages together or lean toward writing their native language in Latin script [2, 3]. Systems trained on monolingual data fail on code-mixed data due to the complexity of code-switching at different linguistic levels in the text [4]. This shared task presents a new gold standard corpus for offensive language detection of code-mixed text in Dravidian languages (Malayalam-English and Tamil-English). Malayalam and Tamil belong to the South Dravidian languages and are

FIRE 2020: Forum for Information Retrieval Evaluation, December 16-20, 2020, Hyderabad, India
EMAIL: bharathi.raja@insight-centre.org (B.R. Chakravarthi); m_anandkumar@nitk.edu.in (A.K. M); john.mccrae@insight-centre.org (J.P. McCrae); b_premjith@cb.amrita.edu (B. Premjith); kp_soman@amrita.edu (K.P Soman); mandl@uni-hildesheim.de (T. Mandl)
ORCID: 0000-0002-4575-7934 (B.R. Chakravarthi)

considered as under-resourced [5, 6]. Malayalam and Tamil have their own scripts for writing such as Malayalam script and Tamil script. However, social media users uses Latin script to write it in online. Therefore, most of the social media comments/posts in Malayalam and Tamil are available in Latin script or in code-mixed form [7, 8]. The research in offensive language identification from social media comments/posts is still in its infancy, particularly in code-mixed Indian languages. The unavailability of gold standard corpus and limited scientific study in the area motivates the introduction of organizing this shared task. Shared tasks such as Task-6 in SemEval-2019 [9], Task-12 in SemEval-2020 [10], and Hate Speech and Offensive Content Identification in Indo-European Languages (HASOC)-2019 [11] inspired us to organize this event for Dravidian languages.

2. Task Description

The goal of this task is to identify offensive language from a code-mixed dataset of comments/posts in Dravidian Languages (Malayalam-English and Tamil-English) collected from social media. The comment/post may contain more than one sentence but the average sentence length of the corpora is 1. Each comment/post is annotated with offensive language label at the comment/post level. The task-1 dataset also has class imbalance problems depicting real-world scenarios. The participants were provided with development, training and test dataset.

Task1:

This is a message-level label classification task. Given a YouTube comment in code-mixed Malayalam, systems have to classify it into offensive or not-offensive.

Task2:

This is a message-level label classification task. Given a tweet or Youtube comments in Tanglish and Manglish (Tamil and Malayalam using written using Roman Characters), systems have to classify it into offensive or not-offensive.

3. Dataset Description

3.1. Task 1 data

For the Task 1, we downloaded data from YouTube comments. The comments were downloaded from movie trailers during 2019. All the comment from those movie trailers were downloaded using a YouTube comment scrapper [1]. We utilized these comments to make a dataset for offensive language identification classification dataset. The dataset contains all types of code-mixing such as mixing the scripts of Malayalam script and Latin script, mixing at the word level, mixing at inter-sentential and intra-sential [12, 13, 14].

3.2. Task 2 data

The Tamil code-mixed dataset for Task 2 was collected from the Twitter tweets and comments on the Helo App. We have considered only the comments/posts in the Latin characters. Malayalam dataset for Task 2 has collected from YouTube comments. The training dataset for the Tanglish and Manglish used for the Task 2 contained 4000 comments. In Tamil, 2997 comments were collected from Twitter and 1003 are from Helo App. Out of 4000 comments, 1980 comments are offensive and 2020 comments are not offensive. The test dataset consists of 940 comments on which 475 are offensive

[1] https://github.com/egbertbouman/youtube-comment-downloader

and 465 are not offensive. Malayalam training set contains 1953 offensive comments and 2047 not-offensive comments, whereas the test set consist of 512 offensive and 488 not-offensive comments. The comments were annotated manually and verified by experts. The comments were annotated with two tags - OFF (offensive comment) and NOT (not-offensive comment). The dataset was given to the participants in CSV format.

The baseline system for Task 2 for Manglish and Tanglish data used Support Vector Machine (SVM) classifier [15] with Term Frequency - Inverse Document Frequency (TD-IDF) features. This system didn't undergo any preprocessing and analyzed the data at the word level. The SVM classifier was trained using a linear kernel and a regularization parameter (C) value = 1. Comments are evenly distributed in both classes, and hence the baseline model didn't consider any approach to make classifier a balanced one. The baseline model for Malayalam achieved the F-score of 0.68, precision score of 0.69 and the recall score of 0.68 and Tamil achieved the F-score of 0.89, precision and recall of 0.89.

4. Evaluation

All teams were allowed a total of 3 submission per Task. Participants had to submit prediction for Tasks in language of their choosing. A submission's final score for each Task was computed as the highest score of the submission. The systems were evaluated on precision, recall and F1-score. This takes into account the varying degrees of importance of each class in the dataset. We used a classification report tool from Scikit learn[2].

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

$$\text{F-Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

5. Methodology

We received a total of 12 submissions for Task 1 for Malayalam, 13 submission for Task 2 for Malayalam, and 12 submission for Task 2 for Tamil. The systems were evaluated based on F1 scores and a rank list was prepared. Table 1, Table 2 and Table 3 show the rank lists of Malayalam and Tamil tasks. We briefly describe below the methodologies used by the participating teams who submitted the paper.

- SivaSai@BITS [16]: The authors proposed a novel and flexible approach of selective translation and transliteration to be able to reap better results out of fine-tuning and ensemble multilingual transformer networks like XLM-RoBERTa and mBERT.
- CENmates [19]: The participants used TF-IDF vectors along with character level n-grams as features to the proposed system for system development. They developed and evaluated four systems consisting of logistic regression, XGBoost, long short-term memory networks, and attention networks. They noted that simple TD-IDF approach with character level n-gram features using machine learning classifiers was producing good results almost the same as the results obtained with deep learning-based classifiers.

[2]https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html

TeamName	Precision	Recall	F-Score	Rank
SivaSai@BITS [16]	0.95	0.95	0.95	1
IIITG-ADBU [17]	0.95	0.95	0.95	1
CFILT-IITBOMBAY	0.94	0.94	0.94	2
SSNCSE-NLP [18]	0.94	0.94	0.94	2
CENMates [19]	0.93	0.93	0.93	3
NIT-AI-NLP [20]	0.93	0.93	0.93	3
YUN [21]	0.93	0.93	0.93	3
Zyy1510 [22]	0.93	0.93	0.93	3
Gauravarora [23]	0.92	0.91	0.91	4
WLV-RIT [24]	0.89	0.90	0.89	5
Kjdong(only not)	0.70	0.83	0.76	6
Ajees [25]	0.69	0.38	0.44	7

Table 1
Rank list based on F1-score along with other evaluation metrics (Precision and Recall) for Malayalam Subtask 1

TeamName	Precision	Recall	F-Score	Rank
CENmates [19]	0.78	0.78	0.78	1
SivaSai [16]	0.79	0.75	0.77	2
KBCNMUJAL [26]	0.77	0.77	0.77	2
IIITG-ABDU [17]	0.77	0.76	0.76	3
SSNCSE-NLP [18]	0.78	0.74	0.75	4
Gauravarora [23]	0.76	0.72	0.74	5
CFILT [27]	0.74	0.70	0.72	6
NITP [20]	0.71	0.68	0.69	7
Ajees [25]	0.72	0.67	0.68	8
Baseline	0.69	0.68	0.68	8
YUN [21]	0.67	0.67	0.67	9
Zyy1510 [22]	0.68	0.67	0.67	9
CUSAT [28]	0.54	0.54	0.54	10

Table 2
Rank list based on F1-score along with other evaluation metrics (Precision and Recall) for Malayalam Subtask 2

- CUSATNLP [28]: The participants did extensive preprocessing and removed the hashtags, URLs, and emojis. They used the simple LSTM layers, a recurrent dropout (0.2). The system results were very low compared to other participants.

- KBCNMUJAL [26]: The participants as well used the common preprocessing techniques to simplify the text messages in addition to that they also remove the special tags. They performed elaborate analysis with classical machine learning model with char n-gram and word n-gram as features. They also reported results for ensemble models. They ranked second for Malayalam and third for Tamil in Task 2.

- Gauravarora [23]: Participants proposed pre-training ULMFiT on synthetically generated code-mixed data, generated by modelling code-mixed data generation as a Markov process using Markov chains. Their model achieved 0.88 weighted F1-score for code-mixed Tamil-English

TeamName	Precision	Recall	F-Score	Rank
SivaSaiBITS [16]	0.90	0.90	0.90	1
SSNCSE-NLP [18]	0.88	0.88	0.88	2
Gauravarora [23]	0.88	0.88	0.88	2
KBCNMUJAL [26]	0.87	0.87	0.87	3
IIITG-ADBU [17]	0.87	0.87	0.87	3
Zyy1510 [22]	0.88	0.87	0.87	3
CENmates [19]	0.86	0.86	0.86	4
CFILT [27]	0.86	0.86	0.86	4
YUN [21]	0.85	0.85	0.85	5
NIT-AI-NLP [20]	0.84	0.84	0.84	6
Baseline	0.85	0.84	0.84	6
Ajees [25]	0.84	0.83	0.83	7

Table 3
Rank list based on F1-score along with other evaluation metrics (Precision and Recall) for Tamil Subtask 2

language in Task and got 2nd rank on the leader-board. Additionally, their model achieved 0.91 weighted F1-score (4th Rank) for mixed-script Malayalam-English in Task 1 and 0.74 weighted F1-score (5th Rank) for code-mixed Malayalam-English language in Task 2.

- SSNCSE [18]: The participants used char n-gram, TFIDF and fine-tuned BERT in combination with machine learning models such as MLP, Random Forest and Naive Bayes.

- NITP-AI-NLP [20]: The authors 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. They have hown 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.

- YUN [21]: The authors proposed an ensemble model which makes full use of the information of rich sequential patterns. More precisely, the proposed model contains a self-attention based on the BiLSTM and the sub-word representation learning. Experimental results of their model on the Malayalam-English of Task 1, Tamil-English and Malayalam-English of Task 2 have achieved the F1 values of 0.93, 0.85 and 0.67, respectively, and ranked 3rd, 5th, 9th, respectively.

- Zyy1510 [22]: ensemble model combines with different models to improve the F-1 value of the framework. The ensemble model is a combination of a BiLSTM (Bidirectional LSTM), an LSTM+Convolution, and a CNN (Convolution Neural Network) model. The proposed model have achieved an F-1 of 0.93 (ranked 3 rd) in Malayalam-English of task1, and F-1 of 0.87 (ranked 3 rd) and 0.67 (ranked 9 th) in Tamil-English and Malayalam-English of task2, respectively

- Ajees [25]: Mainly three types of machine learning models were experimented using two types of word embedding techniques. The first architecture was a simple MLP classifier using two hidden layers. The second one was a combination of CNN-BiLSTM network with four convolutional layers. And the third one was a BiLSTM stack with two hidden layers. In order to represent the individual words in the comments, a simple CountVectorizer as well as BERT was used. CountVectorizer is used to convert text data to a vector of token counts. It also provides the pre-processing of text data before generating the vector representation. This capability makes it highly flexible towards feature representation for text processing. On the other hand, BERT is a pre-trained NLP model that provides contextualized word embeddings. Static word embedding techniques provide the same vector for polysemous words without proper consideration of their context. Whereas the dynamic embedding techniques like ELMo and BERT can consider the context of words before generating their embeddings. BERT word representations take the entire input sentence into the equation for calculating the word embeddings.

- WLV-RIT [24]: applying cross-lingual contextual word embeddings and transfer learning to make predictions to Malayalam data. They further improve the results using various fine tuning strategies.
- IIITG-ADBU [17] paper presents the results obtained by our SVM and XLM-RoBERTa based classifiers in the shared task.
- CFILT[27] presents an ensemble of multilingual BERT models for this task and devise a novel training strategy involving data augmentation using random transliteration. They achieve an F-score of 0.95 for hate speech and offensive content detection on the Malayalam code-mixed YouTube comments test data in task 1. In task 2, they achieve F-scores of 0.86 and 0.72 respectively for hate speech and offensive content detection on Tamil and Malayalam code-mixed Twitter test data.

6. Results and Discussion

In Malayalam Task 1, teams SivaSai@BITS and IIITG-ADBU shared the first position with an F1-score of 0.95. These two systems achieved precision and recall score of 0.95. Teams from CFILT-IITBOMBAY and SSNCSE-NLP achieved the second position with an F-score of 0.94. The top four teams attained F-score higher than 0.90. The difference between the evaluation scores of top teams is minuscule. Table 1 presents the results of the Malayalam Task 1. Teams placed in the first and second position utilized transformer-based model for classification of YouTube comments into OFF and NOT. Transliteration of Romanized text into the native script is also found to be effective in this method. Another important fact visible from the results is the Support Vector Machine classifiers with TF-IDF features also reach top positions. Other systems submitted to the task use deep learning models using Bidirectional LSTM, LSTM, CNN and ULMFiT.

In Malayalam Task 2, CENmates reached the first position with an F-score of 0.78. Teams SivaSai@BITS and KBCNMUJAL bagged the second place, and their F-score was 0.77. The scores of the top five teams are close. Team CENmates used TF-IDF features with character n-gram as features for classification using machine learning algorithms. SivaSai@BITS used the same approach followed for Malayalam Task 1 for this task also. Team KBCNMUJAL used character and word n-grams features for with machine learning classifiers. Other teams used transformer-based models and Deep Learning-based models. Table 2 presents the result of Malayalam Task 2.

In Tamil Task 2, team SivaSai@BITS placed in the first position with an F-score of 0.90. They used the transformer-based model for this task also. Team SSNCSE-NLP grabbed the second position, and they scored an F-score of 0.88. They used TF-IDF with character n-gram features for classification. Gauravarora, who also came second position followed a pre-trained ULMFiT for the classification. Three teams reached third position with an F-score of 0.87. These three teams used entirely different features and classifiers for the prediction task. Team KBCNMUJAL uses character n-gram and word n-gram features for representing text with ensemble models for classification. Team IIITG-ADBU used

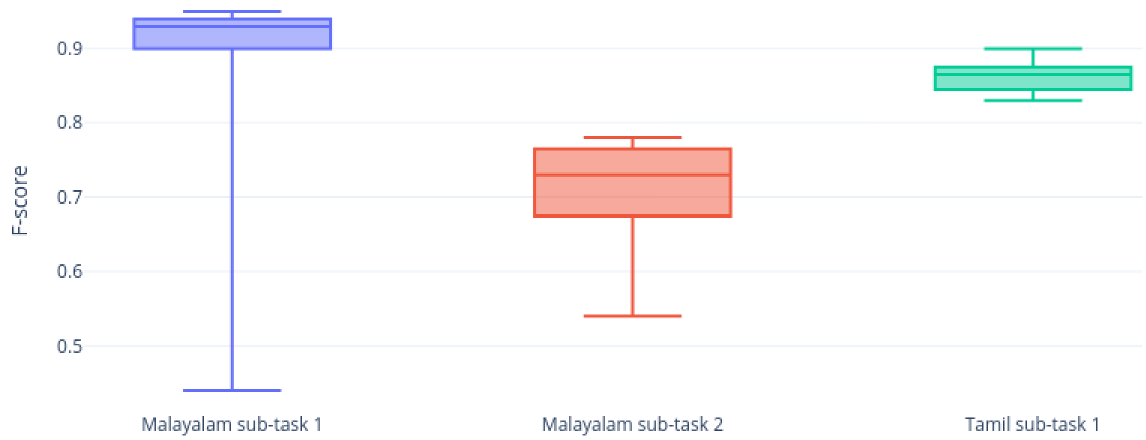

Figure 1: Box-plot for the submissions for Malayalam Task 1, Malayalam Task 2 and Tamil Task 1

the same model used for other tasks for this task also. Team Zyy1510 used an ensemble of BiLSTM, LSTM+Convolution and a Convolution for the classification of social media texts into OFF and NOT. Table 3 presents the results of Tamil Task 2.

When we analyse the models submitted to the Tasks, most of them used either transformer-based models or conventional machine learning classifier with TF-IDF features. The performance of the deep learning models such as Bidirectional LSTM, LSTM, and CNN was not up to the mark. Transformer-based models used BERT for generating the embedding.

Figure 1 shows the box-plots of the performance of the systems for Malayalam Task 1, Malayalam Task 2 and Tamil Task 2, respectively. From the Figures, it is clear that Median of the F-score of all the submissions is close to the top score in Malayalam Task 1 and Malayalam Task 2.

7. Conclusion

We presented the overview and results of the shared task on HASOC-Offensive language detection in Dravidian languages. A wide range of systems were evaluated on two task in two languages relying on a thoroughly annotated dataset. The task setup provided an opportunity to test models on code-mixed Dravidian languages with mixed scripts. NLP for code-mixed texts is very challenging and under-resourced setting also make it even more challenging. We found that many systems were based on transformers and pre-trained embedding based systems. We that this shared task makes a lasting contribution to the research in Dravidian languages.

Acknowledgments

This publication is the outcome of the research supported in part by a research grant from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289_P2 (Insight_2), co-funded by the European Regional Development Fund as well as by the EU H2020 programme under grant agreements 825182 (Prêt-à-LLOD), and Irish Research Council grant IRCLA/2017/129 (CARDAMOM-Comparative Deep Models of Language for Minority and Historical Languages).

We would like to thank Mrs.Sharmila Devi V, PhD scholar, Karpagam College of Engineering, Coimbatore for sharing the task 2 Tamil code-mixed offensive dataset.

References

[1] N. Jose, B. R. Chakravarthi, S. Suryawanshi, E. Sherly, J. P. McCrae, A survey of current datasets for code-switching research, in: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020.

[2] P. V. Veena, M. A. Kumar, K. P. Soman, An effective way of word-level language identification for code-mixed facebook comments using word-embedding via character-embedding, in: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017, pp. 1552–1556. doi:10.1109/ICACCI.2017.8126062.

[3] B. R. Chakravarthi, N. Rajasekaran, M. Arcan, K. McGuinness, N. E.O'Connor, J. P. McCrae, Bilingual lexicon induction across orthographically-distinct under-resourced Dravidian languages, in: Proceedings of the Seventh Workshop on NLP for Similar Languages, Varieties and Dialects, Barcelona, Spain, 2020.

[4] R. Priyadharshini, B. R. Chakravarthi, M. Vegupatti, J. P. McCrae, Named entity recognition for code-mixed Indian corpus using meta embedding, in: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020.

[5] B. R. Chakravarthi, M. Arcan, J. P. McCrae, WordNet gloss translation for under-resourced languages using multilingual neural machine translation, in: Proceedings of the Second Workshop on Multilingualism at the Intersection of Knowledge Bases and Machine Translation, European Association for Machine Translation, Dublin, Ireland, 2019, pp. 1–7. URL: https://www.aclweb.org/anthology/W19-7101.

[6] B. R. Chakravarthi, Leveraging orthographic information to improve machine translation of under-resourced languages, Ph.D. thesis, NUI Galway, 2020.

[7] B. R. Chakravarthi, M. Arcan, J. P. McCrae, Comparison of Different Orthographies for Machine Translation of Under-Resourced Dravidian Languages, in: 2nd Conference on Language, Data and Knowledge (LDK 2019), volume 70 of *OpenAccess Series in Informatics (OASIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2019, pp. 6:1–6:14. URL: http://drops.dagstuhl.de/opus/volltexte/2019/10370. doi:10.4230/OASIcs.LDK.2019.6.

[8] B. R. Chakravarthi, R. Priyadharshini, B. Stearns, A. Jayapal, S. S, M. Arcan, M. Zarrouk, J. P. McCrae, Multilingual multimodal machine translation for Dravidian languages utilizing phonetic transcription, in: Proceedings of the 2nd Workshop on Technologies for MT of Low Resource Languages, European Association for Machine Translation, Dublin, Ireland, 2019, pp. 56–63. URL: https://www.aclweb.org/anthology/W19-6809.

[9] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, R. Kumar, SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval), in: Proceedings of the 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019, pp. 75–86. URL: https://www.aclweb.org/anthology/S19-2010. doi:10.18653/v1/S19-2010.

[10] M. Zampieri, P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov, H. Mubarak, L. Derczynski, Z. Pitenis, Ç. Çöltekin, Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020), arXiv preprint arXiv:2006.07235 (2020).

[11] T. Mandl, S. Modha, P. Majumder, D. Patel, M. Dave, C. Mandlia, A. Patel, Overview of the HASOC track at FIRE 2019: Hate speech and offensive content identification in Indo-European languages, in: Proceedings of the 11th Forum for Information Retrieval Evaluation, 2019, pp. 14–17.

[12] 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 Spoken Lan-

guage Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL), European Language Resources association, Marseille, France, 2020, pp. 177–184. URL: https://www.aclweb.org/anthology/2020.sltu-1.25.

[13] 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 Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL), European Language Resources association, Marseille, France, 2020, pp. 202–210. URL: https://www.aclweb.org/anthology/2020.sltu-1.28.

[14] A. Hande, R. Priyadharshini, B. R. 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 Emotions in Social Media, Barcelona, Spain, 2020.

[15] K. Soman, R. Loganathan, V. Ajay, Machine learning with SVM and other kernel methods, PHI Learning Pvt. Ltd., 2009.

[16] S. Sai, Y. Sharma, Siva@HASOC-Dravidian-CodeMix-FIRE-2020: Multilingual Offensive Speech Detection in Code-mixed and Romanized Text, in: FIRE (Working Notes), 2020.

[17] A. Baruah, K. A. Das, F. A. Barbhuiya, K. Dey, IIITG-ADBU@HASOC-Dravidian-CodeMix-FIRE2020: Offensive Content Detection in Code-Mixed Dravidian Text, in: FIRE (Working Notes), 2020.

[18] N. N. Balaji, B. Bharathi, SSNCSE-NLP@HASOC-Dravidian-CodeMix- FIRE2020: Offensive Language Identification on Multilingual Code Mixing Text, in: FIRE (Working Notes), 2020.

[19] V. P V, P. Ramanan, R. Devi G, CENMates@HASOC-Dravidian-CodeMix-FIRE2020: Offensive Language Identification on Code-mixed Social Media Comments, in: FIRE (Working Notes), 2020.

[20] S. Kumar, Abhinav adn Saumya, J. P. Singh, NITP-AI-NLP@HASOC-Dravidian-CodeMix-FIRE2020: A Machine Learning Approach to Identify Offensive Languages from Dravidian Code-Mixed Text, in: FIRE (Working Notes), 2020.

[21] K. Dong, YUN@HASOC-Dravidian-CodeMix-FIRE2020: A Multi-component Sentiment Analysis Model for Offensive Language Identification, in: FIRE (Working Notes), 2020.

[22] Y. Zhu, X. Zhou, Zyy1510@HASOC-Dravidian-CodeMix-FIRE2020: An Ensemble Model for Offensive Language Identification, in: FIRE (Working Notes), 2020.

[23] G. Arora, Gauravarora@HASOC-Dravidian-CodeMix- FIRE2020: Pre-training ULMFiT on Synthetically Generated Code-Mixed Data for Hate Speech Detection, in: FIRE (Working Notes), 2020.

[24] T. Ranasinghe, M. Zampieri, WLV-RIT @ HASOC 2020: Offensive Language Identification in Code-switched Texts, in: FIRE (Working Notes), 2020.

[25] A. A P, Ajees@HASOC-Dravidian-CodeMix-FIRE2020, in: FIRE (Working Notes), 2020.

[26] V. Pathak, M. Joshi, P. Joshi, M. Mundada, T. Joshi, KBCNMUJAL@HASOC-Dravidian-CodeMix-FIRE2020: Using Machine Learning for Detection of Hate Speech and Offensive Codemix Social Media text, in: FIRE (Working Notes), 2020.

[27] P. Singh, P. Bhattacharyya, CFILT IIT Bombay@HASOC-Dravidian-CodeMix FIRE 2020: Assisting ensemble of transformers with random transliteration, in: FIRE (Working Notes), 2020.

[28] S. Renjit, CUSAT-NLP@HASOC-Dravidian-CodeMix-FIRE2020: Identifying Offensive Language from Manglish Tweets, in: FIRE (Working Notes), 2020.