Overview of the HASOC-DravidianCodeMix Shared Task on Offensive Language Detection in Tamil and Malayalam

Bharathi Raja Chakravarthi[a], Prasanna Kumar Kumaresan[b], Ratnasingam Sakuntharaj[c], Anand Kumar Madasamy[d], Sajeetha Thavareesan[c], B Premjith[e], K Sreelakshmi[e], Subalalitha Chinnaudayar Navaneethakrishnan[f], John P. McCrae[a] and Thomas Mandl[g]

[a]*Insight Centre for Data Analytics, National University of Ireland, Galway*
[b]*Indian Institute of Information Technology and Management-Kerala, India*
[c]*Eastern University, Sri Lanka*
[d]*National Institute of Technology Karnataka Surathkal, Karnataka, India*
[e]*Center for Computational Engineering and Networking (CEN), Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India*
[f]*SRM Institute of Science and Technology, Chennai, Tamil Nadu, India*
[g]*University of Hildesheim, Germany*

Abstract

We present the results of HASOC-Dravidian-CodeMix shared task[1] held at FIRE 2021, a track on offensive language identification for Dravidian languages in Code-Mixed Text in this paper. This paper will detail the task, its organisation, and the submitted systems. The identification of offensive language was viewed as a classification task. For this, 16 teams participated in identifying offensive language from Tamil-English code mixed data, 11 teams for Malayalam-English code mixed data and 14 teams for Tamil data. The teams detected offensive language using various machine learning and deep learning classification models. This paper has analysed those benchmark systems to find out how well they accommodate a code-mixed scenario in Dravidian languages, focusing on Tamil and Malayalam.

Keywords

Sentiment analysis, Dravidian languages, Tamil, Malayalam, Kannada, Code-mixing,

1. Introduction

Advancements in technology have aimed to ease peoples' lives and have attracted many users towards digitization, particularly younger generations [1, 2]. As a result, the number of people

[1]https://dravidian-codemix.github.io/HASOC-2021/index.html

FIRE 2021: Forum for Information Retrieval Evaluation, December 17-21, 2020, Hyderabad, India

bharathi.raja@insight-centre.org (B.R. Chakravarthi); prasanna.mi20@iiitmk.ac.in (P.K. Kumaresan); sakuntharaj@esn.ac.lk (R. Sakuntharaj); m_anandkumar@nitk.edu.in (A.K. Madasamy); sajeethas@esn.ac.lk (S. Thavareesan); b_premjith@cb.amrita.edu (B. Premjith); k_sreelakshmi@cb.students.amrita.edu (K. Sreelakshmi); subalalitha@gmail.com (S.C. Navaneethakrishnan); john.mccrae@insight-centre.org (J.P. McCrae); mandl@uni-hildesheim.de (T. Mandl)

© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

using social media to express their opinions and beliefs has increased dramatically [3]. However, the lack of regulation gives individuals the freedom to post offensive content. There is also no mechanism to regulate the posting of hateful content in under-resourced languages [4, 5, 6].

Tamil is a Dravidian language spoken primarily in Sri Lanka, India, Malaysia, and Singapore [7, 8, 9]. It is an agglutinative language with a rich morphological structure [10]. Tamil has 247 letters comprising of 12 vowels, 18 consonants, 216 composite letters combining each consonant with each vowel, and one special letter known as "Ayutha eluththu". Malayalam is also a Dravidian language spoken in Kerala, India [11, 12, 13]. Malayalam also has its own script for writing; however, social media users use Latin script or mix languages when commenting or posting online [14, 15].

The HASOC-DravidianCodeMix shared task 2021 aims to provide a new gold standard corpus for offensive language identification of code-mixed text in Dravidian languages (Tamil-English and Malayalam-English). Code-mixed content online results from people mixing multiple languages, especially their native language and another commonly spoken language while expressing their views [16]. Offensive language often comprises of hate speech, such as racism, ageism, homophobia, transphobia, ableism and any hate-promoting content against an individual or group [17]. It has been an active area of research in both academia and industry for the past two decades [18]. There is an increasing demand for the identification of offensive language in code-mixed social media texts [19].

There were 16 teams involved in identifying offensive language from Tamil-English code mixed data, 11 teams in identifying offensive language from Malayalam-English code mixed data, and 14 teams in identifying offensive language in Tamil data. The teams used a variety of machine learning and deep learning classification models to identify offensive language. The purpose of this study is to examine such benchmark systems in order to determine how well they fit a code-mixed scenario in Dravidian languages, with a particular emphasis on Tamil and Malayalam.

2. Task Description

The task aims to identify offensive language content of the code-mixed comments/posts in Dravidian Languages (Tamil, Tamil-English and Malayalam-English) collected from social media. The comment/post may contain more than one sentence, but the average sentence length in the corpora is one. Each comment/post is annotated at the comment/post level. This dataset also exhibits class imbalance problems that mirrors real-world scenarios.

- **Task 1**
  Task 1 focuses on offensive language identification from Tamil text. Task 1 is a coarse-grained binary classification where each participating system has to classify YouTube comments in Tamil into two classes: Offensive and Not-offensive.
  - Not-Offensive – The comments does not contain offensive language. Example:

    Text: **பேரவை சார்பாக படம் வெற்றி பெற வாழ்த்துக்கள்**

Task	Train set	Development set	Test set	Total data points
Task 1: Tamil	5,880	-	654	6,534
Task 2: Tamil	4,000	940	1,001	5,941
Task 3: Malayalam	4,000	951	1,000	5,951

Table 1
Number of comments in datasets used for Task 1 and Task 2 and their split into train, development and test set

Translation: Congratulations on the success of the film on behalf of the Assembly

- Offensive - The comments contain hate, offensive or profane content.

  Text: **போடா வெங்காயம் ஒன்னயலாம் அடுச்சு கொள்ளமும் வெண்ணை .**
  Translation: You onion we should beat you to death butter – butter and onion are offensive words in Tamil.

- **Task 2**
  Task 2 focus on offensive language identification in code-mixed Malayalam-English and Tamil-English comments. Example: Code-mixed Tamil

  - Not-Offensive – The comments does not contain offensive language.

    Text: **iantha padam rumba nalla iruku**
    Translation of codemixed Tamil: This movie is very good

  - Offensive – The comments does not contain offensive language.

    Text: **i ammaye bhegikku**
    Translation of codemixed Malayalam: f..k this mother f..kers

2.1. Dataset description

The datasets for both Task 1 and Task 2 were prepared by collecting comments from YouTube. Table 1 shows the number of comments in each dataset.

2.1.1. Task 1: Tamil Dataset

We collected data from YouTube comments for Task 1 using the YouTube comment scrapper [1] to download the comments from particular videos. The comments were collected from movie trailers. We removed all the comments which were not in Tamil. These comments were then used to create a dataset for the offensive language classification task. This dataset contains a total of 6,534 comments and is split into train and test. The training dataset consists of 5,880 comments and the test dataset consists of 654 comments.

[1] https://pypi.org/project/youtube-comment-scraper-python/

No.	TeamName	Precision	Recall	F1-Score	Rank
1	SSN_NLP	0.856	0.864	0.859	1
2	MUCIC [20]	0.850	0.861	0.852	2
3	SSN_NLP_MLRG [21]	0.841	0.847	0.844	3
4	IRLab [22]	0.839	0.835	0.837	4
5	BITS Pilani [23]	0.831	0.846	0.835	5
6	AIML [24]	0.823	0.843	0.825	6
7	Pegasus [25]	0.812	0.807	0.810	7
8	KonguCSE	0.749	0.797	0.764	8
9	Jusgowithurs	0.750	0.817	0.750	9
10	Gothainayaki.A	0.855	0.824	0.749	10
11	MUM	0.853	0.821	0.742	11
12	SSNCSE_NLP [26]	0.747	0.725	0.735	12
13	AI_ML NIT Patna	0.710	0.717	0.714	13
14	Saahil Raj	0.706	0.547	0.599	14

Table 2
Rank list based on weighted average F1-score along with other evaluation metrics (Precision and Recall) for Task 1: Tamil track

2.1.2. Tamil and Malayalam Dataset

Task 2 data was also taken from YouTube comments and posts. These comments were used to create a dataset for the offensive language classification in both languages. The dataset includes different types of code-mixing, such as mixing Tamil and Latin characters for the Tamil dataset, code mixed data for the Malayalam dataset, and mixing at the word level. The Tamil dataset contains a total of 5,941 comments from this split into training, development and test. The training dataset consists of 4,000 comments, the development dataset contains 940 comments, and the test dataset consists of 1,001 comments. The Malayalam dataset contains a total of 5,951 comments from this split into training, development and test. The training dataset consists of 4,000 comments, the dev dataset contains 951 comments, and the test dataset consists of 1,000 comments. These datasets also are published in the same competition, HASOC-Dravidian CodeMixed, which is on Codalab.

3. Methodology

We have received fourteen, sixteen and eleven submissions for Task 1: Tamil track, Task 2: Tamil track and Task 2: Malayalam track, respectively. The submissions were evaluated based on weighted average F1-score, and rank lists were prepared accordingly. Table 2 shows the rank list of teams that participated in Task 1: Tamil track. Tables 3 and 4 show the rank lists of the teams that competed in Task 2: Tamil track and Task 2: Malayalam track, respectively. Tables 2, 3 and 4 show the precision, recall and weighted average F1-score of all the participating teams on test data. In this section, we briefly describe the methodologies of teams that participated in the three tasks.

- SSN_NLP_MLRG [21]: Team SSN_NLP_MLRG participated in the Tamil-English sub-

No.	TeamName	Precision	Recall	F1-Score	Rank
1	MUCIC [20]	0.679	0.685	0.678	1
2	AIML [24]	0.670	0.670	0.670	2
3	SSN_IT_NLP [27]	0.685	0.688	0.668	3
4	ZYBank AI	0.671	0.676	0.654	4
5	IRLab [22]	0.654	0.662	0.650	5
6	HSU [28]	0.655	0.664	0.649	6
7	IIITSurat [29]	0.679	0.673	0.636	7
8	Team Pegasus [25]	0.633	0.644	0.612	8
9	PSG [30]	0.614	0.609	0.611	9
10	SSNCSE_NLP [26]	0.615	0.607	0.610	10
11	IIITD-shanker [31]	0.599	0.568	0.573	11
12	CEN_NLP	0.596	0.540	0.539	12
13	RameshKannan	0.524	0.526	0.525	13
14	MUM	0.591	0.527	0.522	14
15	AI_ML_NIT_Patna	0.539	0.509	0.515	15
16	JBTTM	0.537	0.483	0.503	16

Table 3
Rank list based on weighted average F1-score along with other evaluation metrics (Precision and Recall) for Task 2: Tamil track

No.	TeamName	Precision	Recall	F1-Score	Rank
1	AIML [24]	0.776	0.762	0.766	1
2	MUCIC [20]	0.764	0.760	0.762	2
3	HSU [28]	0.744	0.730	0.735	3
4	IIIT Surat [29]	0.752	0.727	0.734	4
5	IRLab [22]	0.754	0.705	0.714	5
6	IIITD-ShankarB [31]	0.715	0.693	0.700	6
7	SSNCSE_NLP [26]	0.692	0.678	0.683	7
8	Pegasus [25]	0.708	0.660	0.670	8
9	CEN_NLP	0.652	0.635	0.641	9
10	MUM	0.628	0.637	0.632	10
11	JBTTM	0.577	0.584	0.580	11

Table 4
Rank list based on weighted average F1-score along with other evaluation metrics (Precision and Recall) for Task 2: Malayalam track

task. The authors implemented both traditional machine learning and deep learning models for the classification. They experimented with Support Vector Machine (SVM) [32], naive bayes, random forest and extreme gradient boosting ensemble classifiers for categorizing the offensive contents with N-gram, character and word level Term Frequency-Inverse Document Frequency (TF-IDF) and Bag-of-Words (BoW) features. The deep learning models used for the classification includes a shallow Neural Network (NN), a Long Short Term Memory (LSTM) [33] and a Convolutional Neural Network

(CNN). The embeddings in the NN were initialized using the fastText [34] pre-trained word embeddings. The authors also followed a transfer learning approach by multilingual Bidirectional Encoder Representation (mBERT) [35], ALBERT [36] (A Lite BERT for self-supervised learning of language representations), DistilBERT [37] (Distilled version of BERT[35]) with the ktrain, and ULMFiT [38] with Fastai [39] to build the classification model.

- HSU_TransEmb [28]: Team HSU_TransEmb used a Transformer ensemble system to identify the offensive contents from Tamil-English and Malayalam-English code-mixed data. The ensemble system consists of mBERT, DistilBERT and MuRIL models [40]. The preprocessed data were fed to the three ensemble BERT models, and the class probabilities were computed. The class label was identified from the sum of the class probabilities obtained from the BERT models.

- MUCIC [20]: Team MUCIC took part in both Tamil-English and Malayalam-English shared tasks. They used word-level as well as character-level N-gram based TF-IDF for extracting the features from the texts. Furthermore, they identified 40,000 frequent features in each case and constructed a combined set containing 80,000 frequent features. They employed linear SVM, random forest, logistic regression and an ensemble of these three classifiers to train the model. The logistic regression model obtained the highest F1-score of 0.881 in the Tamil-English task, whereas random forest exhibited the best performance with an F1-score of 0.783.

- IIITSurat [29]: Team IIITSurat took part in both shared tasks and employed machine learning and deep learning models for classification. Machine learning classifiers such as logistic regression, random forest, naive bayes, XG boost, and SVM were trained over TF-IDF features. In addition to machine learning models, the authors executed Deep Neural Network (DNN), CNN, BiLSTM and Transformer-based models such as BERT [35], Indic BERT [41] and MuRIL [40] for classification. Among all the models, MuRIL achieved the highest F1-scores of 0.78 and 0.91 in Malayalam-English and Tamil-English tasks, respectively.

- Pegasus [25]: Team Pegasus submitted their results in Task 1 and Task 2. They utilized XLM-RoBERTa [42] and DistilBERT models for identifying offensive language social media text. As mentioned earlier, the authors deployed the embedding generated using the BERT and fed it into a BiLSTM network. In Task 1, Team Pegasus to avoid repetition of the authors concatenated the embeddings obtained from both BERT models and passed them to a BiLSTM network. This model attained an F1-score of 0.810. The authors performed transliteration and translation on Task 2 data and applied the XLM-RoBERTa model to extract the embedding, which obtained F1-scores of 0.612 and 0.670 in Tamil-English and Malayalam-English tasks, respectively.

- IRLab [22]: Team IRLab implemented a Deep Neural Network (DNN) with TF-IDF features for Tasks 1 and 2. The authors extracted unigram to six-gram TF-IDF features and identified the first 30,000 features. A DNN with four dense layers read these features and predicted the class label for each data. They also performed hyperparameter tuning for

each model to fix the best model. Their model achieved F1-scores of 0.84, 0.65 and 0.71 in Task 1, Tamil-English, and Malayalam-English shared tasks.

- AIML [24]: Team AIML proposed an ensemble model which used character N-gram-based TF-IDF features for the identification of offensive texts. The authors considered one to six character N-gram features and trained an ensemble of SVM, logistic regression and random forest. Their model attained an F1-score of 0.83 in Task 2, whereas it achieved F1-scores of 0.67 and 0.77 in Tamil-English and Malayalam-English tasks, respectively.
- SSN_IT_NLP [27]: Team SSN_IT_NLP presents an offensive language identification model for Tamil-English data. The mBERT generates embeddings from the data, which are then fed to an ensemble of SVM, XG Boost and Linear Discriminant Analysis (LDA). The label predicted by the majority of the models was selected as the final output.
- NLP_CSE: Team NLP_CSE employed machine learning and deep learning models for predicting the offensive data. A logistic regression classifier takes TF-IDF features for training the model. Furthermore, the authors used random oversampling algorithms to deal with the class imbalance problem in the data. The model obtained an F1-score of 0.5243. In addition to the logistic regression model, the authors implemented an LSTM-based encoder-decoder architecture and a transformer-based model. The encoder-decoder model was a deep multi-layer network that also incorporated an attention mechanism. This model consisted of stacks of four encoders and four decoders. The transformer model, mBERT, was used to generate the embedding for sentences and considered the cosine similarity between sentences for classification.
- BITS_Pilani [23]: Team BITS_Pilani used a DNN which contain an embedding layer, pooling layer, dropout layer, a fully connected layer and an output layer for classifying the text into Offensive and Not offensive in the Tamil-English subtask. The model achieved an F1-score of 0.835 in the competition.
- M Subramanian et al.: Team M Subramanian et al. employed the naive bayes multinomial model, KNN, logistics regression, and SVM classifier with BoW features for classifying the social media text into offensive or not offensive categories. This team participated in the shared task for only Tamil data. The Logistic regression model attained the highest performance among the classifiers.

4. Evaluation

The distribution of the offensive languages classes are imbalanced in both datasets. 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].

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

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

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

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

$$P_{\text{weighted}} = \sum_{i=1}^{L} (P \text{ of } i \times \text{ Weight of } i) \tag{4}$$

$$R_{\text{weighted}} = \sum_{i=1}^{L} (R \text{ of } i \times \text{ Weight of } i) \tag{5}$$

$$F - Score_{\text{weighted}} = \sum_{i=1}^{L} (F - Score \text{ of } i \times \text{ Weight of } i) \tag{6}$$

5. Results and Discussion

Shared tasks on offensive language detection in CodeMix Tamil and Malayalam data were organized as part of HASOC 2021. Fourteen submissions for Track 1: Tamil and sixteen submissions in Track 2. For Malayalam, eleven teams submitted their results in Track 2. Table 5 shows the number of teams participated in each shared task. Participating teams explored N-gram based TF-IDF, BoW and different variants of BERT for representing the input text. None of the teams used language specific features. They used various conventional machine learning classifiers such as SVM, naive bayes, random forest. logistic regression, XG boost, KNN and ensemble of machine learning classifier models for the identification of the offensive language text. In addition to that, DNN, LSTM and its variants and transformer-based classifiers were also studied for the classification. Team HSU_TransEmb explored an ensemble of mBERT, DistilBERT and MuRIL for detecting offensive texts from CodeMix Tamil and Malayalam data. NLP_CSE investigated the performance of oversampling algorithms to address the class imbalance problem in the data. Tables 1, 2 and 3 show the rank lists for Task 1: Tamil track, Task 2: Malayalam track and Task 2: Tamil track, respectively. Figures 1, 2 and 3 show precision, recall and F1-scores of submissions in Track 1: Tamil, Track 2: Tamil and Track 2: Malayalam. Figure 4 shows the box-plots of the performance of the teams participated in Track 1: Tamil, Track 2: Tamil and Track 2: Malayalam.

Team SSN_NLP obtained the first rank in Track 1 with an F1-score of 0.859. MUCIC and SSN_NLP_MLRG grabbed second and third positions with F1-scores of 0.852 and 0.844. Among the 14 teams, seven scored F1-scores greater than 0.8. Looking at the models used by the teams, one can see that the teams that finished top used different kinds of feature extraction models and classifiers.

Team MUCIC attained the first position in Track 2: Tamil shared task, and they achieved an F1-score of 0.678. MUCIC used word level as well as character level N-gram based TF-IDF features for classification. They performed the predictions using SVM, random forest, logistic regression, and an ensemble of these three. The second-placed team, AIML, and the

Competition	No. of teams participated
All three tasks	6
Track 1: Tamil	7
Both tasks in Track 2	5
Track 2: Tamil alone	4
Track 2: Malayalam alone	0
Track 1: Tamil and Track 2: Tamil	1

Table 5
Number of teams participated in each shared task

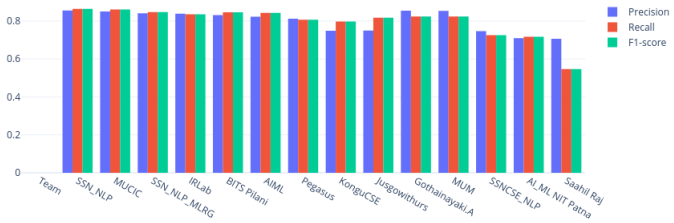

Figure 1: Bar plot describing Precision, Recall and F1-scores of the submissions for Track 1: Tamil

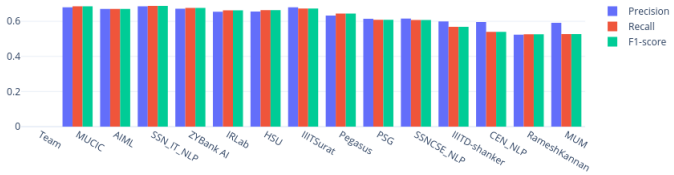

Figure 2: Bar plot describing Precision, Recall and F1-scores of the submissions for Track 2: Tamil

third-placed team, SSN_IT_NLP, scored F1-scores of 0.670 and 0.668, respectively. AIML also utilized the N-gram based TF-IDF features with SVM, logistic regression and random forest. They considered unigram to six-gram features for this analysis. SSN_IT_NLP made use of mBERT embeddings with SVM, XG boost and LDA to identify the offensive language texts among the data. Among the 16 teams that participated, ten teams recorded F1-scores greater than 0.6.

In Track 2: Malayalam, AIML reached the top position with an F1-score of 0.766. MUCIC and HSU were placed in the second and third positions with F1-scores of 0.762 and 0.735, respectively. AIML used unigram to six-gram based TF-IDF features with SVM, logistic regression and random forest classifiers for the identification of offensive language texts. MUCIC also

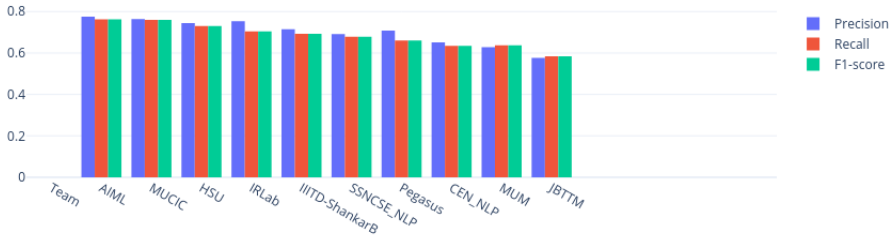

Figure 3: Bar plot describing Precision, Recall and F1-scores of the submissions for Track 2: Malayalam

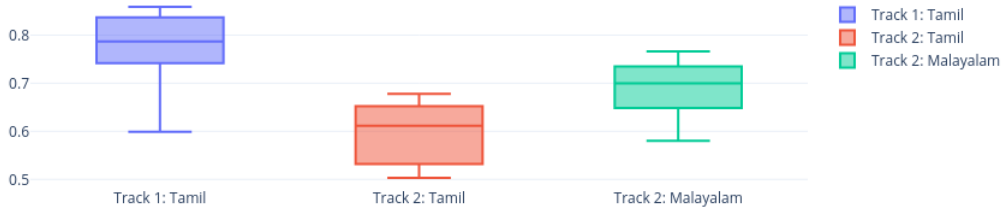

Figure 4: Box-plot for the submissions for Track 1: Tamil, Track 2: Tamil and Track 2: Malayalam

followed a similar methodology, but they used only the most frequent forty thousand n-gram based TF-IDF features from each class for classification. Team HSU utilized an ensemble of mBERT, DistilBERT and MuRIL for the detection of offensive language contents. In this task, 6 out of 11 teams obtained an F1-score greater than 0.7, and one team scored an F1-score less than 0.6.

It is interesting to note that teams that used TF-IDF features attained the top position in both tasks in Track 2. A similar trend was visible in HASOC 2020 [43]. The teams that won the HASOC 2020 shared tasks in CodeMix data used TF-IDF features with machine learning classifiers.

6. Conclusion

This paper gives an overview of the HASOC- Dravidian-CodeMix shared task at FIRE 2021. The shared task consisted of three subtasks for Tamil, CodeMix Tamil and Malayalam languages. There were 16 teams who participated in Tamil-English code mixed data, 11 teams in Malayalam-English code mixed data and 14 teams in Tamil data. Teams used methods ranging from Bag of Words, TF-IDF to BERT-based models to represent the data and applied conventional machine learning algorithms, deep neural networks and transformer networks for prediction. One team employed oversampling algorithms to deal with the imbalance in the data by synthetically generating the data points in minority classes. The analysis of the methods of the teams showed that both conventional and deep learning/transformer-based methods

exhibit similar performances in terms of the evaluation metrics used for assessing the models.

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), and Irish Research Council grant IRCLA/2017/129 (CARDAMOM-Comparative Deep Models of Language for Minority and Historical Languages). We also thank Ciara Oloughlin for her help with proof reading.

References

[1] B. R. Chakravarthi, 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, Association for Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 41–53. URL: https://aclanthology.org/2020.peoples-1.5.

[2] B. R. Chakravarthi, V. Muralidaran, 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, Kyiv, 2021, pp. 61–72. URL: https://aclanthology.org/2021.ltedi-1.8.

[3] S. Suryawanshi, B. R. Chakravarthi, M. Arcan, S. Little, P. Buitelaar, TrollsWithOpinion: A Dataset for Predicting Domain-specific Opinion Manipulation in Troll Memes, arXiv preprint arXiv:2109.03571 (2021).

[4] J. J. Andrew, JudithJeyafreedaAndrew@DravidianLangTech-EACL2021:offensive language detection for Dravidian code-mixed YouTube comments, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv, 2021, pp. 169–174. URL: https://aclanthology.org/2021.dravidianlangtech-1.22.

[5] B. Bharathi, A. S. A, SSNCSE_NLP@DravidianLangTech-EACL2021: Offensive language identification on multilingual code mixing text, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv, 2021, pp. 313–318. URL: https://aclanthology.org/2021.dravidianlangtech-1.45.

[6] B. R. Chakravarthi, R. Priyadharshini, R. Ponnusamy, P. K. Kumaresan, K. Sampath, D. Thenmozhi, S. Thangasamy, R. Nallathambi, J. P. McCrae, Dataset for Identification of Homophobia and Transophobia in Multilingual YouTube Comments, arXiv preprint arXiv:2109.00227 (2021).

[7] R. Sakuntharaj, S. Mahesan, A novel hybrid approach to detect and correct spelling in Tamil text, in: 2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS), IEEE, 2016, pp. 1–6.

[8] R. Sakuntharaj, S. Mahesan, Use of a novel hash-table for speeding-up suggestions for

misspelt Tamil words, in: 2017 IEEE International Conference on Industrial and Information Systems (ICIIS), IEEE, 2017, pp. 1–5.
[9] R. Sakuntharaj, S. Mahesan, Detecting and correcting real-word errors in Tamil sentences, Ruhuna Journal of Science 9 (2018).
[10] Nuhman, Basic Tamil Grammar, Readers Association, Kalmunai, Department of Tamil, University of Peradeniya, 2013.
[11] S. Thavareesan, S. Mahesan, Word embedding-based Part of Speech tagging in Tamil texts, in: 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS), 2020, pp. 478–482. doi:10.1109/ICIIS51140.2020.9342640.
[12] S. Thavareesan, S. Mahesan, Sentiment Lexicon Expansion using Word2vec and fastText for Sentiment Prediction in Tamil texts, in: 2020 Moratuwa Engineering Research Conference (MERCon), 2020, pp. 272–276. doi:10.1109/MERCon50084.2020.9185369.
[13] S. Thavareesan, S. Mahesan, Sentiment Analysis in Tamil Texts: A Study on Machine Learning Techniques and Feature Representation, in: 2019 14th Conference on Industrial and Information Systems (ICIIS), 2019, pp. 320–325. doi:10.1109/ICIIS47346.2019.9063341.
[14] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, N. Jose, S. Suryawanshi, E. Sherly, J. P. McCrae, DravidianCodeMix: Sentiment Analysis and Offensive Language Identification Dataset for Dravidian Languages in Code-Mixed Text, arXiv preprint arXiv:2106.09460 (2021).
[15] B. R. Chakravarthi, K. Soman, R. Ponnusamy, P. K. Kumaresan, K. P. Thamburaj, J. P. McCrae, et al., DravidianMultiModality: A Dataset for Multi-modal Sentiment Analysis in Tamil and Malayalam, arXiv preprint arXiv:2106.04853 (2021).
[16] 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), IEEE, 2020, pp. 136–141.
[17] 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://aclanthology.org/S19-2010. doi:10.18653/v1/S19-2010.
[18] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, S. Suryawanshi, N. Jose, E. Sherly, J. P. McCrae, Overview of the track on sentiment analysis for dravidian languages in code-mixed text, Forum for Information Retrieval Evaluation (2020).
[19] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, R. Kumar, Predicting the type and target of offensive posts in social media, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 1415–1420. URL: https://aclanthology.org/N19-1144. doi:10.18653/v1/N19-1144.
[20] F. Balouchzahi, S. Bashang, G. Sidorov, H. L. Shashirekha, CoMaTa OLI- Code-mixed Malayalam and Tamil Offensive Language Identification, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.
[21] A. Kalaivani, D. Thenmozhi, SSN_NLP_MLRG@Dravidian-CodeMix-FIRE2020: Senti-

ment Code-Mixed Text Classification in Tamil and Malayalam using ULMFiT, in: FIRE (Working Notes), 2020.
[22] A. Saroj, S. Pal, IRLab@IIT-BHU@Dravidian-CodeMix-FIRE2020: Sentiment Analysis on Multilingual Code Mixing Text Using BERT-BASE, in: FIRE (Working Notes), 2020.
[23] S. Tripathy, A. Pathak, Y. Sharma, Offensive Language Classification of Code-Mixed Tamil with Keras, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.
[24] J. Kumari, A. Kumar, Offensive Language Identification on Multilingual Code Mixing Text, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.
[25] P. Kalyan Jada, K. Yasaswini, K. Puranik, A. Sampath, S. Thangasamy, K. Pal Thamburaj, Analyzing Social Media Content for Detection of Offensive Text, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.
[26] N. N. Appiah Balaji, B. B, B. J, SSNCSE_NLP@Dravidian-CodeMix-FIRE2020: Sentiment Analysis for Dravidian Languages in Code-Mixed Text, in: FIRE (Working Notes), 2020.
[27] S. Divya, N. Sripriya, Offensive Content Recognition, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.
[28] S. N. V. C. Basava, A. P. Karri, Transformer Ensemble System for Detection of Offensive Content in Dravidian Languages, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.
[29] S. Bhawal, P. Roy, A. Kumar, Offensive Language Identification on Multilingual Code Mixed Text using BERT, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.
[30] S. Benhur J, K. S, Pretrained Transformers for Offensive Language Identification in Tanglish, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.
[31] S. Biradar, S. Saumya, A. Chauhan, mBERT based model for identification of offensive content in south Indian languages, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.
[32] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (1995) 273–297.
[33] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.
[34] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of tricks for efficient text classification, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Association for Computational Linguistics, 2017, pp. 427–431.
[35] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.
[36] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, Albert: A lite bert for self-supervised learning of language representations, arXiv preprint arXiv:1909.11942

(2019).
[37] 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).
[38] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 328–339. URL: https://aclanthology.org/P18-1031. doi:10.18653/v1/P18-1031.
[39] J. Howard, S. Gugger, Fastai: a layered api for deep learning, Information 11 (2020) 108.
[40] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. K. Margam, P. Aggarwal, R. T. Nagipogu, S. Dave, et al., Muril: Multilingual representations for indian languages, arXiv preprint arXiv:2103.10730 (2021).
[41] D. Kakwani, A. Kunchukuttan, S. Golla, N. Gokul, A. Bhattacharyya, M. M. Khapra, P. Kumar, inlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, 2020, pp. 4948–4961.
[42] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, arXiv preprint arXiv:1911.02116 (2019).
[43] B. R. Chakravarthi, A. K. M, J. P. McCrae, B. Premjith, K. Soman, T. Mandl, Overview of the track on HASOC-Offensive Language Identification-DravidianCodeMix., in: FIRE (Working Notes), 2020, pp. 112–120.