TOLD: Tamil Offensive Language Detection in Code-Mixed Social Media Comments using MBERT with Features based Selection

Adaikkan Kalaivani[1,2], Durairaj Thenmozhi[3] and Chandrabose Aravindan[4]

[1]*Department of Information and Communication Engineering, Anna University, Chennai*
[2]*Research Centre, Department of CSE, Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam, TamilNadu*
[3]*Department of CSE, Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam, TamilNadu*
[4]*Department of Information Technology, Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam, TamilNadu*

Abstract

The immense growth in social media forums does increase the spread of offensive language. We detect and examine the challenges faced by automatic approaches for offensive language detection in the Tamil-English language. Among these difficulties are subtleties in code-mixed Tamil language, identifying what constitutes offensive, and handling the imbalanced data under the low resource language. This paper presents our work in the shared task of HASOC-Dravidian-CodeMix-FIRE 2021, where we explore different machine learning algorithms, deep learning techniques, and transfer learning models. We also explore various feature extraction techniques and utilize offensive features to perform this task. Our team SSN_NLP_MLRG has participated in task1 and classifies the code-mixed Tamil textual content into offensive or not-offensive. Our team best model is Multilingual BERT, and submission had a macro F1-score 0.84 of task1 of Tamil code-mixed language. Our team achieved the 3^{rd} rank on the final test results in task1 for the Tamil code-mixed language.

Keywords

Transfer learning, Code-Mixed language, Dravidian language, Transformers, Language modeling

1. Introduction

Social media is one of the platforms for the public to communicate with each other, share ideas, express thought, and their emotions freely without considering others [1]. These user-driven forums have a challenge when it comes to regulating the content fed into them. People have a different intent, some might use these forums for their intended purposes, and others might be publicly sharing inappropriate content such as offensive language, racist speech, hate speech towards others. Therefore propagation of offensive language is increased, which is widely in code-mixed languages.

Code-mixing plays a critical role in a multilingual community. The code-mixed texts are the mixture of native scripts and non-native scripts like text are in Tamil language but written

FIRE 2021: Forum for Information Retrieval Evaluation, December 13-17,2021, India.
kalaiwind@gmail.com, kalaivania@ssn.edu.in, (A. Kalaivani); theni_d@ssn.edu.in (D. Thenmozhi); aravindanc@ssn.edu.in (C. Aravindan)
0000-0002-1497-5605 (A. Kalaivani); 0000-0003-0681-6628 (D. Thenmozhi); 0000-0002-9025-4009 (C. Aravindan)

in roman script, and mixing of both the Tamil and English languages. The challenging part is to train a system on monolingual data at different linguistic levels in the code-mixed text because of the complexity of code-switching [2].The Tamil language[1] is the official language of India, Srilanka, and Singapore and is natively spoken by the Tamil people and by the Tamil diaspora around the world includes South Africa, Malaysia, United States, United Kingdom, Mauritius, Canada, and Australia. In the social media forum, the native speakers have used the Roman script to input. So, the majority of the texts for these under-resourced languages are code-mixed.

Offensive language[2] such as threatening, harassment, violence, defrauding, sexual comments, gender-specific comments, racial slurs, any content that could seriously offend someone or group. Based on their age, religion, political beliefs, marital, parental status, physical features, national origin, and disability. Offensive language[3] is the offense of using curse language in a way that could offend a reasonable person in, near, or within hearing or view of public forums or schools. More users have been experiencing online harassment. Depending on the circumstances, this offense is a punishable offense by the Court. So the direct and indirect offensive content like sarcasm [3], metaphors in code-mixed text in Dravidian languages is challenging to annotate by humans. Therefore, Automatic detection and identification of such offensive content in the code-mixed languages [4] are very challenging for these social media public forums.

This paper presents a description of our Team SSN_NLP_MLRG submission runs to the shared task of offensive language identification of code-mixed text in Dravidian languages (Tamil-English and Malayalam-English). This task is a part of the Forum for Information Retrieval Evaluation (FIRE 2021) workshop [5]. Our team participated in the shared task1 of offensive language identification of code-mixed text in the Tamil language. The challenges in the shared task of task1 are listed below:

- The problem of highly imbalanced data
- Difficult to transfer the code-mixed language
- Code-mixed data has ungrammatical sentences.
- Native languages have written in English in roman script format.
- Data have Misspelling words, Repeated Letters, Prolonged words, * words, Continuous words.

The goal of task1 is to identify and classify the social media comments are offensive or not offensive language in the Tamil-English code-mixed Language. We explore various approaches like machine learning techniques, neural networks, and pre-trained models to detect the offensive language in Tamil code-mixed language[4]. This paper contains the following sections. Section 2 has the related works. Section 3 has the experimented data and task description. Section 4 has the technique of our models. Section 5 has conducted research and presented its findings. Finally, Section 6 summarises our findings and suggests ways to improve our work.

[1]https://en.wikipedia.org/wiki/Tamil_language
[2]https://www.lawinsider.com/dictionary/offensive-content
[3]https://www.primelawyers.com.au/criminal-law/public-order-offences/offensive-language/
[4]https://github.com/kalaiwind/Dravidian-2021

2. Related work

The novel selective translation and transliteration approach for pre-processing and trained the system by using ensemble XLM-RoBERTa to detect offensive language identification in Dravidian languages [6]. The researchers used the transfer learning-based models to classify the social media comments into six categories in the Dravidian languages [7]. The pseudo-label approach for generating the Dravidian dataset in Tamil, Malayalam, and Kannada languages to classify the offensive content by using the ULMFiT model [8]. In the FIRE 2020: Forum for Information Retrieval Evaluation, an overview of the shared task of HASOC-Offensive Language Identification on code mixed Dravidian languages. They organized two tasks. Task 1 is to identify the offensive language comments in the Malayalam language. Task 2 is to identify the offensive language content in Tamil and Malayalam languages. Most of the participants were used the transformer-based model, machine learning classifier with TF-IDF character n-gram features, and deep learning models.

The researchers used the ALBERT model with the cross-lingual translation to detect hate speech and offensive language in English, Tamil, and German languages [9]. The models were naive Bayes, logistic regression, and vanilla neural network to detect offensive in code mixed Dravidian language for the dataset Tamil code-mix, and Malayalam script-mixed text [10]. The researchers used an ensemble of an AWD-LSTM based model, BERT, RoBERTa for offensive language identification, and the best results were achieved in the Malayalam-English, Tamil-English, and Kannada-English languages [11].

The overview shared task of HASOC: Hate Speech and Offensive Content Identification in Indo-European Languages has two sub-tasks for the three languages are English, German, and Hindi (code-mixed) [12, 13]. Task A is to identify the content is Hate speech and offensive or not offensive and task B is to category the comments into three classes that is hate speech, offensive, and profanity. In SemEval: offensive language detection in English, Danish, Greek, Turkish and Arabic languages [14, 15]. The majority of teams were used con-textualized Transformers, deep learning approaches [16], ELMo embeddings, BERT, RoBERTa, and the multilingual mBERT [17]. To predict the offensive language by the cross-language contextual word embedding with transfer learning methods in less-resourced languages.

Most of the work in offensive language identification from social media comments was done in high-resource languages like English. We still face the problem of handling the dataset in low-resource languages like the Dravidian language [18, 19]. The most important challenging task is to detect offensive comments in social media forums for different code-mixed low resource languages other than English. So this problem has been an active area for both the researchers in academic and industry. HASOC 2021 shared task provides the resource for the Tamil and Malayalam code-mixed languages.

3. Experimental data

This section presents the description of Tamil code-mixed data, task, and data preparation and pre-processing techniques.

Table 1
Train dataset of Tamil code-mixed langauge

Task	Category	No of Comments
Task1	NOT	4664
Task1	OFF	1145
Task1	Not-Tamil	03
Total		**5812**

3.1. Data and Task description

The organizers provided the HASOC 2021 Dravidian dataset for Tamil and Malayalam code-mixed languages, which offer comments from the social media forums. Our team SSN_NLP_MLRG participated in the task1 Tamil code-mixed dataset. The Tamil code-mixed HASOC 2021 dataset consists of 5880 posts for the train system and 654 posts for testing the model system. The task1 of Tamil shared task is a multi-class classification task and aims to classify the posts into three classes, namely Offensive (OFF): the posts contain the curse, profane, offense, threatening words. Not offensive (NOT): The comments do not have offense words. Not-Tamil: The comments do not intend in the Tamil language.

3.2. Data preparation

The Tamil train dataset contains 6534 comments. We have removed the duplication of posts from the training dataset. After removal, task1 of the Tamil language contains 4664 not offensive posts, 1145 offensive posts, and three not-Tamil posts. We have used 5812 posts to build the system. Table 1 shows the description for the Tamil code-mixed dataset. We separately collected the vocabulary of offense content from various sources. First, we have to identify the languages that contain the maximum number of comments from the training dataset. In our case, the Tamil language has the maximum number in the shared task1 of the HASOC-Dravidian-CodeMix-FIRE 2021. We detected the other language and performed two actions. If the text is in the roman script of the Tamil language then, transliterates it into the Tamil language, and the text is in other languages, translate them into the Tamil language by using google API. Figure 1 shows the statistics of the training Tamil code-mixed dataset.

3.3. Data pre-processing

The data pre-processing is important in order to clean the comments from the unnecessary noisy content and transform it into a coherent form, which can be portable for Tamil code-mixed language. We used the NLTK libraries for data cleaning. We remove @ symbol with string and the hashtag symbol with string denoted as the user's name because it does not have any expressions and affects the performance of the model. We removed the punctuation, numerals, symbols, and emojis. After that, we converted the upper case text into small case text. We replaced the misspelling offense words by using the collected offense data. We corrected the repeated letters and then translate the words by using google API. Finally, we replaced the * words into appropriate matched words presents in the collected vocabulary words.

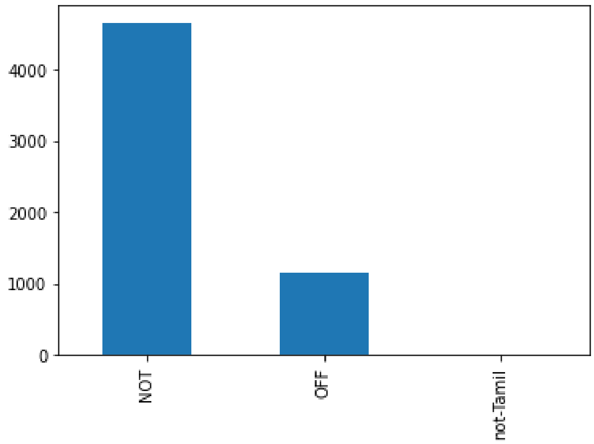

Figure 1: Statistics of Tamil data

4. Methodology

This section presents the different approaches and models experimented with for the Tamil code-mixed data.

4.1. Machine learning Techniques

We experimented with traditional machine learning algorithms namely support vector machine classifier (SVM), Naive Bayes classifier (NB), random forest classifier (RF), and Extreme gradient boosting ensemble classifier (XGB), and used to predict the offensive content in the given code-mixed posts. We used the scikit-learn library [5] implementation of these above-mentioned traditional classifiers. First, the data was pre-processed and extracted the Ngram, character level, word-level features by using Term frequency-inverse document frequency (TF-IDF) vectorization. We used sklearn CountVectorizer which helps to build vocabulary for known words and also tokenize the collected text documents. For the Naive Bayes classifier, we created a count vectorizer object to transform the training and validation data and we extracted the Ngram TF-IDF, character level TF-IDF, word-level TF-IDF features.

In FastText, the pre-trained vectors for 157 languages were trained on common crawl and Wikipedia. We used the FastText pre-trained word embedding vectors for the Tamil language namely Wikipedia Tamil vectors (wiki.ta.vec) and common crawl Tamil vectors (cc.ta.300.vec.gz). For the SVM Classifier, we used TF-IDF Vectorization to extract the features of Ngram TF-IDF vectors. For the Random Forest classifier, we used the count vectorizer and TF-IDF vectorizer for extracting the count vectors and word-level vectors respectively. For the XGBoost classifier, we extracted the features of count vectors, word level, and character level vectors.

[5] https://scikit-learn.org/stable/modules/classes.html

4.2. Deep learning Techniques

The offensive language identification of code-mixed language by the following models, namely neural network (NN), Convolutional neural network (CNN), and recurrent neural network (RNN) with LSTM (Long short term memory) layer. The architecture of a neural network consists of 1 input layer, 1 hidden layer, and 1 output layer. The input is word-level embedding vectors which were extracted by using FastText pre-trained word embedding vectors. We have set the dense is 100 with the activation as relu in the input layer. The output layer has a dense of 1 with the activation as sigmoid and used the Adam optimizer and binary loss cross-entropy.

The architecture of a convolutional neural network consists of a 1D convolutional layer followed by a 1D max-pooling layer, then followed by the 3 output layers. The word-level representation is generated through a 1D convolutional layer with the activation as relu, dense as 50, and drop out of 0.25 in the output layer 1, sigmoid activation in the output layer 2, and used Adam optimizer and binary loss cross-entropy. We obtained the most prominent features by a 1-D maximum pooling layer.

The architecture of a recurrent neural network consists of a 1D convolutional layer followed by a 1D max-pooling layer, then an LSTM layer, and followed the 3 output layers. LSTM has the ability to process its sequences and retain all the information. LSTM has a dropout of 0.25 with activation as relu and sigmoid and optimizers as Adam and set the loss as binary cross-entropy.

4.3. Transfer Learning

Transfer learning plays a turning point in the computer science field and it's led to major improvements and breakthroughs. For the past two decades, the introduction of pre-trained language models namely Universal language model fine-tuning for text classification (ULMFit) and Bidirectional Encoder Representation from Transformers (BERT) led to a revolution in the Natural Language Processing world. Most of the researchers were used BERT-based models and they also achieved state-of-the-art results in many tasks in NLP.

We used the MBERT (Multilingual BERT), ALBERT (A Lite BERT for self-supervised learning of language representations), DistilBERT (Distilled version of BERT) with the ktrain, and ULMFiT [20] with the Fastai to build the system to identify the offensive content in the Tamil code-mixed language. We used the Average-SGD Weight-Dropped LSTM (AWD-LSTM) architecture model for the binary classification task to predict the offensive content or Not-offensive.

Fastai has functions for creating language and classification model data bunches, as well as setting the batch size to 32, the learning rate to 3e-02, 3e-03, 1e-03, 5e-04, and the epoch to 15, 3, 2, and 5 for training. For all the BERT-based models, we take 20 % of the data from the training data for the validation process. We analyzed the trained model to set the batch size to 6, 32 and learning rates as1e-5, 2e-5, 3e-5, and the epochs to 5, 6, and 10. With the pre-trained weights, we fine-tune the classifier. Finally, we have used the MBERT model to predict the offensive content and got a weighted-average F1-score of 0.84 with the epochs 10 and the learning rate as 2e-5 for the task1 Tamil code-mixed language.

Table 2
Validation scores of the different models

Task	Accuracy
ULMFiT	0.81
ALBERT	0.81
MBERT	**0.84**
DistilBERT	0.81
NB Count vector	0.80
NB WordLevel TF-IDF	0.80
NB N-gram	0.83
NB CharLevel	0.83
SVM N-gram	0.82
RF, Count vector	0.80
RF, Word level	0.80
XGB, Count vector	0.82
XGB, Word level	0.82
XGB, CharLevel	0.83
NN, Ngram Level TF IDF	0.80
CNN	0.80
RNN	0.80

5. Experimental analysis and Results

This section presents the analysis of different models and provides the details of results that were experimented with in the Tamil code-mixed data.

5.1. Result Analysis

We experimented with the different models and compared the scores based on the evaluation metrics of weighted precision, weighted recall, and weighted average F1 score. Table 2 presents the validation results of different approaches of models. Based on the performance of the validation process, the NB model with the Character level vector, the NB model with the ngram TF-IDF vectors, XGB classifier with the character level vectors got an accuracy of 0.83 which is close to the MBERT.

MBERT model achieved an accuracy of 0.84 and Precision, Recall and an F1score of 0.83, 0.84, and 0.83 respectively which is compared with the performance of the other machine learning approaches, deep learning approaches, and pre-trained language models. The Precision, Recall, and F1score for the Not-offensive comments are 0.89, 0.92, and 0.90 respectively. The Precision, Recall, and F1score for the offensive comments are 0.58, 0.40, and 0.53 respectively. Table 3 presents the test results of different approaches of models. In the machine learning techniques, the NB model with the ngram features achieved the Precision, Recall, and an F1score of 0.83, 0.84, and 0.84 respectively. NB model with the character level vectors achieved the Precision, Recall, and an F1score of 0.83, 0.83, and 0.84 respectively. So, NB ngram and character level performed well which compared with other performance of the machine learning techniques. For BERT-based Models, MBERT performed well with the Precision, Recall, and an F1score

Table 3
Test Results of the different models

Model	Precision	Recall	F1-Score
ULMFiT	0.67	0.82	0.74
ALBERT	0.67	0.82	0.74
MBERT	0.84	0.85	**0.84**
DistilBERT	0.82	0.83	0.82
NB Count vector	0.82	0.81	0.82
NB WordLevel TF-IDF	0.85	0.82	0.74
NB N-gram	0.83	0.84	0.83
NB CharLevel	0.83	0.83	0.83
SVM N-gram	0.84	0.85	0.81
RF, Count vector	0.84	0.83	0.76
RF, Word level	0.82	0.83	0.77
XGB, Count vector	0.82	0.84	0.80
XGB, Word level	0.81	0.83	0.79
XGB, CharLevel	0.84	0.85	0.82
NN, Ngram Level TF IDF	0.67	0.81	0.73
CNN	0.67	0.81	0.73
RNN	0.67	0.81	0.73

Table 4
Category-wise test Results of MBERT

Category	Precision	Recall	F1-Score
NOT	0.90	0.92	0.91
OFF	0.58	0.53	0.55
Accuracy			**0.85**
Macro Average	0.74	0.72	0.73
Weighted Average	0.84	0.85	**0.84**

of 0.84, 0.85, and 0.84 respectively. We observed that the performance of the MBERT model achieved good results compared to the other models.

5.2. Submitted results

This task is part of a shared competition organized in HASOC-Dravidian-CodeMix-FIRE 2021, where we participated as the SSN_NLP_MLRG team. For task1, we submitted the best performing model and the category-wise results are shown in Table 4. For the Tamil code-mixed task, we submitted the MBERT model which achieved a macro F1-score of 0.84 on the test set. We ranked 3rd in task1 shared of offensive language identification in Tamil code-mixed language.

For further analysis, we used the confusion matrix to represent the performance of a classification model on test data for that the true values are known. Figure 2 shows the confusion matrix of the MBERT model and Figure 3 shows the confusion matrix of the DistilBERT model.

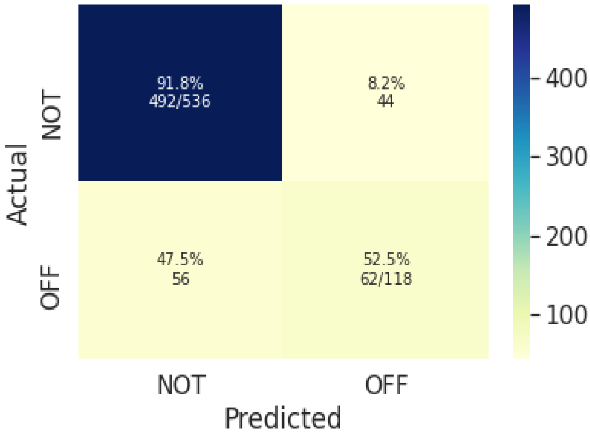

Figure 2: Confusion matrix of MBERT

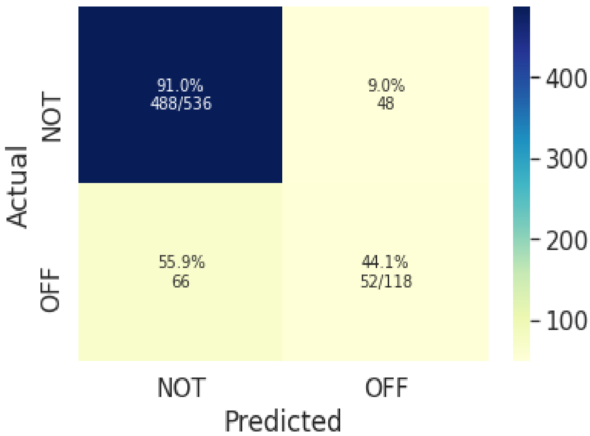

Figure 3: Confusion matrix of DistilBERT

5.3. Error Analysis

For ALBERT Model, The Precision, Recall, and F1score for the Not-offensive comments are 0.82, 1.00, and 0.90 respectively. But the F1 score of the offensive content is 0.00 because the number of not-offensive comments is higher in the overall Tamil code-mixed dataset. We observed the performance of the ULMFiT is the same as like ALBERT model.

From the confusion matrix, we observed that due to an unbalanced dataset many test cases were classified as Not-offensive. For DistilBERT, The Precision, Recall, and F1score for the Not-offensive comments and offensive comments are 0.88, 0.91, 0.90, and 0.52, 0.54, 0.58 respectively. From table Table 4, we observed the F1score for the offensive comments in the MBERT model is 0.55 which comparatively lowers the DistilBERT. So, MBERT performs well in the Not-Offensive category and DistilBERT performs well in the Offensive category. Figure 4 shows the confusion

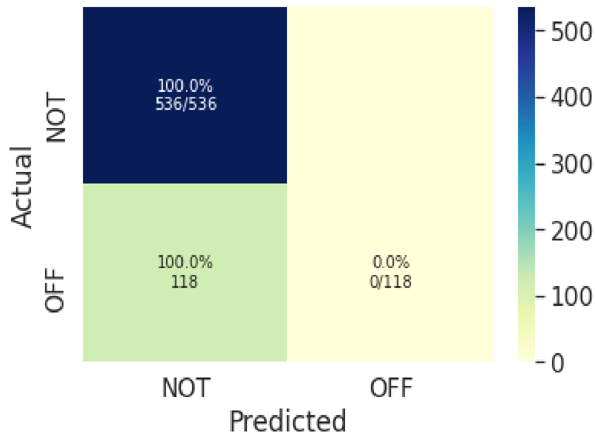

Figure 4: Confusion matrix of ALBERT

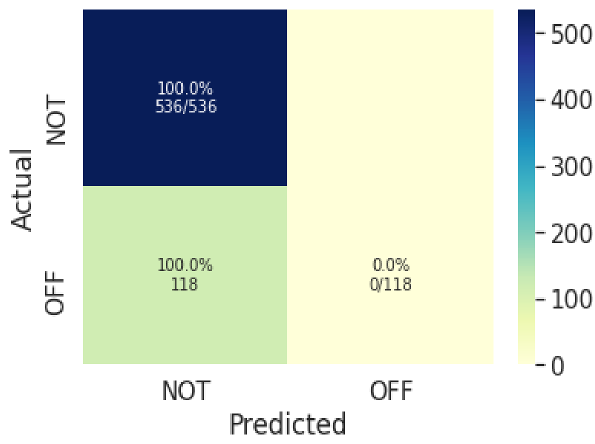

Figure 5: Confusion matrix of ULMFiT

matrix of the ALBERT model and Figure 5 shows the confusion matrix of the ULMFiT model. From the confusion matrix in figure 4 and figure 5, we observed that due to an unbalanced dataset many test cases were classified as Not-offensive.

6. Conclusion and Future enhancements

This paper presents the submitted runs for the offensive language identification for Dravidian Languages in Code-Mixed data in the Forum for Information Retrieval Evaluation (FIRE) 2021. The results show that the Not-offensive class in each dataset receives the highest F1 scores, regardless of the model. This is due to the maximum number of the same as compared to the rest of the class. Comments that were not in the particular Tamil language of their dataset do not receive any classification in the test data. We experimented with different approaches such as machine learning techniques, deep learning approaches, and pre-trained BERT-based models. Based on the evaluation, MBERT performs well. Our team submission had a macro F1-score of 0.84 and achieved the 3rd rank on the final test data in task1 for the Tamil code-mixed language. For future work, we will handle the imbalanced dataset and extend this work into other languages. Further, we will detect the sarcastic feature which helps to avoid misclassification.

References

[1] A. Kalaivani, D. Thenmozhi, Sentimental Analysis using Deep Learning Techniques, International Journal of Recent Technology and Engineering (IJRTE) 7 (2019) 600–606.

[2] 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, CoRR abs/2106.09460 (2021). URL: https://arxiv.org/abs/2106.09460. arXiv:2106.09460.

[3] A. Kalaivani, D. Thenmozhi, Sarcasm Identification and Detection in Conversion Context using BERT, in: Proceedings of the Second Workshop on Figurative Language Processing, Association for Computational Linguistics, Online, 2020, pp. 72–76. URL: https://www.aclweb.org/anthology/2020.figlang-1.10. doi:10.18653/v1/2020.figlang-1.10.

[4] B. R. Chakravarthi, R. Priyadharshini, N. Jose, A. Kumar M, T. Mandl, P. K. Kumaresan, R. Ponnusamy, H. R L, J. P. McCrae, E. Sherly, 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, Kyiv, 2021, pp. 133–145. URL: https://aclanthology.org/2021.dravidianlangtech-1.17.

[5] B. R. Chakravarthi, P. K. Kumaresan, R. Sakuntharaj, A. K. Madasamy, S. Thavareesan, P. B, S. Chinnaudayar Navaneethakrishnan, J. P. McCrae, T. Mandl, Overview of the HASOC-DravidianCodeMix Shared Task on Offensive Language Detection in Tamil and Malayalam, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.

[6] S. Sai, Y. Sharma, Towards offensive language identification for Dravidian languages, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv, 2021, pp. 18–27. URL: https://aclanthology.org/2021.dravidianlangtech-1.3.

[7] K. Yasaswini, K. Puranik, A. Hande, R. Priyadharshini, S. Thavareesan, B. R. Chakravarthi, 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, Kyiv, 2021, pp. 187–194. URL: https://aclanthology.org/2021.dravidianlangtech-1.25.

[8] A. Hande, K. Puranik, K. Yasaswini, R. Priyadharshini, S. Thavareesan, A. Sampath, K. Shanmugavadivel, D. Thenmozhi, B. R. Chakravarthi, Offensive language identification in low-resourced code-mixed Dravidian languages using pseudo-labeling, 2021. arXiv:2108.12177.

[9] A. Kalaivani, D. Thenmozhi, SSN_NLP_MLRG@HASOC-FIRE2020: Multilingual Hate

Speech and Offensive Content Detection in Indo-European Languages using ALBERT, in: Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation, Hyderabad, India, December 16-20, 2020, volume 2826 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 188–194. URL: http://ceur-ws.org/Vol-2826/T2-12.pdf.

[10] S. Saumya, A. Kumar, J. P. Singh, Offensive language identification in Dravidian code mixed social media text, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv, 2021, pp. 36–45. URL: https://aclanthology.org/2021.dravidianlangtech-1.5.

[11] K. Kedia, A. Nandy, indicnlp@kgp at DravidianLangTech-EACL2021: Offensive language identification in Dravidian languages, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv, 2021, pp. 330–335. URL: https://aclanthology.org/2021.dravidianlangtech-1.48.

[12] T. Mandl, S. Modha, G. K. Shahi, A. K. Jaiswal, D. Nandini, D. Patel, P. Majumder, J. Schäfer, Overview of the HASOC track at FIRE 2020: Hate Speech and Offensive Content Identification in Indo-European Languages, 2021. arXiv:2108.05927.

[13] T. Mandl, S. Modha, A. Kumar M, B. R. Chakravarthi, 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, Association for Computing Machinery, New York, NY, USA, 2020, p. 29–32. URL: https://doi.org/10.1145/3441501.3441517. doi:10.1145/3441501.3441517.

[14] 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), CoRR abs/2006.07235 (2020). URL: https://arxiv.org/abs/2006.07235. arXiv:2006.07235.

[15] 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), 2019. arXiv:1903.08983.

[16] D. Thenmozhi, B. Senthil Kumar, S. Sharavanan, A. Chandrabose, SSN_NLP at SemEval-2019 task 6: Offensive language identification in social media using traditional and deep machine learning approaches, in: Proceedings of the 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019, pp. 739–744. URL: https://www.aclweb.org/anthology/S19-2130. doi:10.18653/v1/S19-2130.

[17] A. Kalaivani, D. Thenmozhi, SSN_NLP_MLRG at SemEval-2020 task 12: Offensive language identification in English, Danish, Greek using BERT and machine learning approach, in: Proceedings of the Fourteenth Workshop on Semantic Evaluation, International Committee for Computational Linguistics, Barcelona (online), 2020, pp. 2161–2170. URL: https://aclanthology.org/2020.semeval-1.287.

[18] A. Hande, S. U. Hegde, R. Priyadharshini, R. Ponnusamy, P. K. Kumaresan, S. Thavareesan, B. R. Chakravarthi, Benchmarking multi-task learning for Sentiment analysis and offensive language identification in under-resourced Dravidian languages, arXiv preprint arXiv:2108.03867 (2021).

[19] B. R. Chakravarthi, R. Priyadharshini, S. Banerjee, R. Saldanha, J. P. McCrae, A. K. M, P. Krishnamurthy, M. Johnson, 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, Kyiv, 2021, pp. 119–125. URL: https://aclanthology.org/2021.dravidianlangtech-1.15.

[20] A. Kalaivani, D. Thenmozhi, SSN_NLP_MLRG@Dravidian-CodeMix-FIRE2020: Sentiment Code-Mixed Text Classification in Tamil and Malayalam using ULMFiT, in: FIRE (Working Notes), 2020, pp. 528–534.