CoSaD- Code-Mixed Sentiments Analysis for Dravidian Languages

Fazlourrahman Balouchzahi[1], Hosahalli Lakshmaiah Shashirekha[2] and Grigori Sidorov[1]

[1] *Instituto Politécnico Nacional (IPN), Centro de Investigación en Computación (CIC), Mexico City, Mexico*
[2] *Department of Computer Science, Mangalore University, Mangalore, India*

Abstract

Analyzing sentiments or opinions in code-mixed languages is gaining importance due to increase in the use of social media and online platforms especially during the Covid-19 pandemic. In a multilingual society like India, code-mixing and script mixing is quite common as people especially the younger generation are quite familiar in using more than one language. In view of this, the current paper describes the models submitted by our team MUCIC for the shared task in 'Sentiments Analysis (SA) for Dravidian Languages in Code-Mixed Text'. The objective of this shared task is to develop and evaluate models for code-mixed datasets in three Dravidian languages, namely: Kannada, Malayalam, and Tamil mixed with English language resulting in Kannada-English (Ka-En), Malayalam-English (Ma-En), and Tamil-English (Ta-En) language pairs. N-grams of char, char sequences, and syllables features are transformed into feature vectors and are used to train three Machine Learning (ML) classifiers with majority voting. The predictions on the Test set obtained average weighted F1-scores of 0.628, 0.726, and 0.619 securing 2nd, 4th, and 5th ranks for Ka-En, Ma-En, and Ta-En language pairs respectively.

Keywords

Code-Mixing, Sentiments Analysis, Dravidian Languages, n-grams, Machine Learning

1. Introduction

The task of analyzing the opinions, feelings, and reviews posted on social media or online markets to identify the sentiments of users about a given topic, movie, song, product, etc. is called as Sentiments Analysis (SA). For example, a video on Instagram or a product in e-markets can be viral and popular based on its reviews and sentiments posted by the customers/users [1, 2]. Lately, the demand for SA of social media data has increased both in academia and industry, especially for the code-mixed data [3]. Code-mixed data are common in multilingual communities such as India where people use more than one languages' words, grammar, and phrases in their communication/ posts/ comments in social media or reviews in online shopping websites [4].

Code-mixed content in Dravidian languages is usually a combination of a native language

FIRE 2021, Forum for Information Retrieval Evaluation, December 13-17, 2021, India

frs_b@yahoo.com (F. Balouchzahi); hlsrekha@gmail.com (H. L. Shashirekha); sidorov@cic.ipn.mx (G. Sidorov)
https://mangaloreuniversity.ac.in/dr-h-l-shashirekha (H. L. Shashirekha); http://www.cic.ipn.mx/~sidorov/ (G. Sidorov)
0000-0003-1937-3475 (F. Balouchzahi); 0000-0003-3901-3522 (G. Sidorov)

Table 1
Samples of code-mixed text in Dravidian languages

Language pair	Single script	Multi-scripts
Ta-En	Handsome hunk keri vaa thalaivaa	Trailer wow nu நினைக்கிறவங்க மட்டும் லைக் பண்ணுங்க.....
Ma-En	Gopichettante BGM um mammookayum ishtapedunnavar like!?	Trailer pwolichuuuu ഓണത്തിന് വന്നങ്ങ്ങു തകർത്തക്കേക്
Ka-En	Sari nivu video na roast madi adre madvaga pubg atava free fire games hakondo madbedi a games kooda Chinese	ಚಾಲೆಂಜಿಂಗ್ ಸ್ಟಾರ್ ದರ್ಶನ್ ಅಭಿಮಾನಿಗಳ ಕಡೆಯಿಂದ All the best

such as Kannada, Tamil or Malayalam and English language at different linguistic units such as sentence, phrase, word, morpheme and sub-word. The code-mixed text will either be in a single script which is usually a Roman script or in multi-script i.e., a combination of Roman and native script may be with few words of the native language in Roman script. Table 1 presents some examples of single and multi-scripts code-mixed contents in Ta-En, Ma-En, and Ka-En language pairs from the datasets used in the shared task.

Dravidian languages in general are under-resourced languages and code-mixing adds a further dimension mainly due to the problems with collecting and annotating code-mixed data for various applications. 'Sentiment Analysis for Dravidian Languages in Code-Mixed Text' is a shared task in Dravidian-CodeMix-FIRE2021[1] with the aim of promoting SA of code-mixed texts in Ka-En, Ma-En, and Ta-En language pairs [5, 6]. This shared task is an extension of previous shared task of SA in Ta-En and Ma-En in FIRE 2020 [3] with the addition of Ka-En language pair [6].

The objective of the shared task is to identify the opinion/sentiment of the comments posted by the users on a given topic and classifying them further into one of the following categories:

- **Positive:** comments contain positive contents or justify that speaker is in a positive state
- **Negative:** comments contain negative contents or justify that speaker is in a negative state
- **Mixed_Feelings:** comments contain positive as well as negative contents and hence cannot be explicitly categorized into one of the two classes mentioned earlier
- **Unknown_state:** emotional state of a speaker is not clear or comments does not contain positive or negative contents explicitly
- **Not in indented language:** comments are not written in the intended language

In the earlier works, i) Balouchzahi et al. [1] experimented various features such as Skipgram word embedding, BPEmb[2] sub-word embedding, and a combination of word and char n-grams to train ML classifiers for SA, and ii) Balouchzahi et al. [2] also explored and compared different learning approaches such as ML, Deep Learning (DL), and Transfer Learning (TL) for SA. In continuation of these works in SA in Dravidian languages, this paper describes the models

[1] https://dravidian-codemix.github.io/2021/index.html
[2] https://nlp.h-its.org/bpemb/

Table 2
Statistics of the datasets used in Dravidian-CodeMix-FIRE2020

Class		Positive	Negative	Mixed_ Feelings	Unknown_ state	Other languages	Total
Dataset	Ta-En	10,559	2,037	1,801	850	497	15,744
	Ma-En	2,811	738	403	1,903	884	6,739

submitted by our team MUCIC to the Dravidian-CodeMix-FIRE2021 shared task. Three different feature sets, namely: char, char sequences, and syllables are explored to check the effectiveness of char level (characters) and sub-word level (char sequences and syllables) n-grams for code-mixed SA task. Each feature set is individually used to train three ML classifiers, namely: Linear Support Vector Machine (LSVM), Logistic Regression (LR) and Multi-Layer Perceptron (MLP) and the majority voting of the predictions of all the classifiers is used to classify the given sentiment. The code of the proposed methodology is available in our GitHub link[3].

The rest of paper is organized as follows: Section 2 gives a summary of the best models submitted to the Sentiment Analysis for Dravidian Languages in Code-Mixed Text in Dravidian-CodeMix-FIRE2020[4] shared task and the Methodology is described in Section 3. Section 4 describes the results obtained and the paper concludes in Section 5.

2. Related Work

Researches had submitted several models to ’Sentiment Analysis for Dravidian Languages in Code-Mixed Text’ shared task in Dravidian-CodeMix-FIRE2020 organized by Chakravarthi et al. [3, 7]. The shared task consists of similar sentiments categories (as mentioned in Section 1) in two language pairs, namely: Ta-En and Ma-En. Authors collected the Youtube comments to develop datasets consisting of 15,744 and 6,739 comments in Ta-En and Ma-En language pairs respectively, and provided the same to the participants of the shared task as Train, Dev and Test set. The label distribution of the comments in the dataset shown in Table 2 (borrowed from [2]) illustrates that the dataset is imbalanced for both the language pairs.

Participants were supposed to train and evaluate their models locally on Train and Dev set respectively and then predict the class label of the Test set. These predictions were submitted to the shared task organizers for final evaluation and ranking which is based on average weighted scores. The brief descriptions of the models which exhibited good performance in this shared task are given below:

Most of successful teams have utilized Multilingual BERT (mBERT[5]) [8] and XLM-Roberta [9] - the multilingual transformer based models for SA similar to that of code-mixed Offensive Language Identification (OLI) in Dravidian languages [10]. With the objective of using Masked Language Modeling (MLM), mBERT was trained on the top 104 languages that have largest Wikipedia including Kannada, Malayalam, and Tamil. Pires et al. [11] describe that mBERT can be employed for cross-lingual generalization. Moreover, based on the authors’ experiments,

[3] https://github.com/fazlfrs/CoSaD
[4] https://dravidian-codemix.github.io/2020/index.html
[5] https://github.com/google-research/bert/blob/master/multilingual.md

despite the high lexical overlap among different languages, mBERT is capable of transfering between languages with different scripts by capturing multilingual representations. XLM-Roberta also relay on MLM objective and cross-lingual transfer. By using the large-scale multilingual pre-training model trained on 2.5 TB of clean CommonCrawl data in 100 languages [12], XLM-Roberta has overcome the limitation of XLM [13] and mBERT in learning useful representations for under-resourced languages.

Sun et al. [12] proposed a XLM-Roberta based model by extracting the abundant semantic information from the hidden layer state of XLM-Roberta, which is then fed as input into convolution and max pooling. Further, they concatenated the top hidden states and pooler to improve performances and reported that the proposed model without any pre-processing obtained better results. The proposed model outperformed all other models submitted to the shared task by securing 1st ranks (for both the language pairs) with average weighted F1-scores of 0.74 and 0.65 for Ma-En and Ta-En language pairs respectively.

Ou et al. [14] developed a XLM-Roberta based model similar to the work of Sun et al. [12]. Here, the authors obtained the pooler output and the sequence of hidden states of the last layer of XLM-Roberta and concatenated the pooler output with the average-pooling and max-pooling of hidden-states of XLM-Roberta into a classifier. They merged and shuffled the Train and Dev sets and used k-fold cross validation to enhance the performances of the system. They obtained 1st rank for Ma-En language pair and average weighted F1-scores of 0.74 and 0.63 for Ma-En and Ta-En language pairs respectively. In a simple way, Sun et al. [15] proved the efficiency of multilingual transformers by fine-tuning the pre-trained multilingual BERT adopted from multi_cased_L12_H-768_A-12[6]. They secured 2nd and 4th ranks with average weighted F1-scores of 0.73 and 0.62 for Ma-En and Ta-En language pairs respectively.

Huang et al. [16] proposed a multi-step integration of fine-tuned XLM-Roberta and mBERT transformers for the shared task and obtained average weighted F1-scores of 0.73 and 0.63 for Ma-En and Ta-En language pairs respectively. They used mBERT as binary classifier and XLM-Roberta as quaternary classifier and intertwined both model's predictions for final decision. Zhu et al. [17] experimented an mBERT-based model along with BiLSTM by feeding the hidden state of the last layer of mBERT model to BiLSTM. Further, they set weights for each hidden state layer in BiLSTM and the weighted sum of hidden states is concatenated with the original output of mBERT. The results reported in leaderboard shows 2nd rank for both language pairs with average weighted F1-scores of 0.73 and 0.64 for Ma-En and Ta-En language pairs respectively.

In addition to transformers, several models based on ML classifiers have also obtained promising results in the shared task. Kanwar et al. [18] adopted under-sampling technique from TOMEK [19] to train several ML classifiers with various syntax based n-grams features. The best performance obtained was using LR classifier with word and char n-grams features for Ma-En and Ta-En language pairs with 0.71 and 0.62 average weighted F1-scores respectively. Balouchzahi et al. [2] submitted a majority voting of ML classifiers (Multinomial Naïve Bayes trained on Skipgram word embedding and Multi-Layer Perceptron (MLP) trained on the combination of word and char n-grams) and BiLSTM model (with training a sub-word embedding using BPEmb library that is used as weight later in BiLSTM) for the shared task. The proposed model obtained 0.68 and 0.62 average weighted F1-scores for Ma-En and Ta-En language pairs

[6] https://github.com/google-research/bert

respectively.

Researchers have explored several models based on ML and DL approaches with a combination of different embeddings and feature sets. Balouchzahi et al. [1] explored ML, DL and TL approaches by proposing (i) a ML-based voting classifier trained on a feature set of char sequences along with BPEmb sub-words ngrams and syntactic ngrams [20, 21] with three estimators, namely: LR, MLP, and eXtreme Gradient Boosting (XGB); (ii) A Keras sequential classifier trained on earlier feature set; and (iii) A Universal Language Model Fine-Tuning (ULMFiT) for SA. They used Dakshina[7] dataset as raw text to train a tokenizer, universal Language Model (LM) for fine tuning and fast.ai[8] library for training LM and SA classification model. Using ML-based voting classifier they obtained 0.72 and 0.62 average weighted F1-scores for Ma-En and Ta-En language pairs respectively.

3. Methodology

The proposed methodology contains:

- pre-processing texts
- extracting char, char sequences and syllables as features from the texts
- obtaining the corresponding n-grams from the n-gram generator
- vectorizing the n-grams using TfidfVectorizer
- training the ML classifiers
- predict the labels of the Test set

The pre-processing module adopted from Balouchzahi et al. [2] includes converting Emojis to text, removing punctuation, numbers, unnecessary characters and words of length less than 2 and lower casing the words written in Roman script. Words are split by a simple strategy of using attributes of string data type in a 'for' loop to obtain char features. Char sequences are extracted as sub-word level features using everygrams[9] library from NLTK. Syllable which comprises of vowels and consonants [22] is a smallest unit used to organize sequences of sounds and are considered as the building blocks in Text To Speech (TTS) tasks. Sidorov [23] proposed using syllables as features for Text Classification (TC) tasks. Syllable features are extracted using the Syllablizer[10] library. Though the library works better for native scripts, results for code-mixed texts are also encouraging.

The n-gram generator accepts a list of chars/char sequences/syllables of a word as input and will generate the corresponding n-grams which are vectorized using TfidfVectorizer[11] to train the ML classifiers.

The overview of feature engineering which includes the procedures to pre-process, extract features, generate n-grams, and obtaining TFIDF vectors is shown in Figure 1 and the range of n-grams for each feature type is given in Table 3.

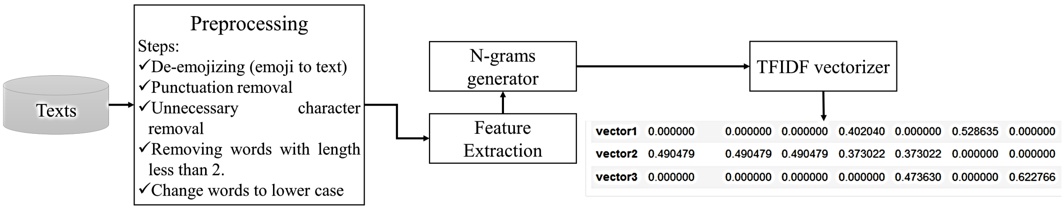
[7] https://github.com/google-research-datasets/dakshina

[8] https://nlp.fast.ai

[9] https://tedboy.github.io/nlps/generated/generated/nltk.everygrams.html

[10] https://github.com/libindic/syllabalizer.git

[11] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html


Texts
Preprocessing
Steps:
✓De-emojizing (emoji to text)
✓Punctuation removal
✓Unnecessary character removal
✓Removing words with length less than 2.
✓Change words to lower case
N-grams generator
Feature Extraction
TFIDF vectorizer

vector1	0.000000	0.000000	0.000000	0.402040	0.000000	0.528635	0.000000
vector2	0.490479	0.490479	0.490479	0.373022	0.373022	0.000000	0.000000
vector3	0.000000	0.000000	0.000000	0.000000	0.473630	0.000000	0.622766

Figure 1: Feature Engineering

Table 3
n-grams range for feature sets

Feature	Char	Char sequences	Syllables
n-grams range	(1, 5)	(1, 6)	(1, 8)

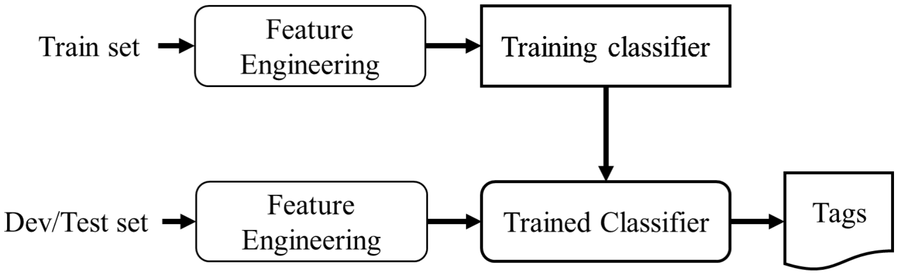


Figure 2: Training the classifiers

The parameters of LSVM and LR classifiers are set to default and that of MLP classifier are set as: hidden_layer_sizes = (150, 100, 50), max_iter = 300, activation = 'relu', solver = 'adam', random_state = 1. Each classifier is trained separately with the three feature sets mentioned earlier. The best performing feature and classifier pairs are selected manually based on their performances on Dev set and majority voting of the predictions on the Test set were submitted for final evaluation to the shared task organizers. Figure 2 presents the steps for training the individual classifier for each feature set.

4. Experiments and Results

A post/comment in each language pair should be classified into one of the five categories as described in Section 1. The dataset provided by the shared task organizers [5, 6] includes a collection of code-mixed text from social media in three language pairs, namely: Ma-En, Ta-En, and Ka-En. These datasets were split into Train, Dev and Test sets and provided to the participants of the shared task to train and evaluate the models. The statistics of the datasets are given in Table 4. Similar to Dravidian-CodeMix-FIRE2020 shared task, the label distribution over the datasets illustrate that the datasets are highly imbalanced. The observation of the datasets in Table 4 illustrate that, for each class, Ta-En language pair consists of more samples and Ka-En language pair consists of less samples and this could affect the performance of the

Table 4
Statistics of the Datasets

Class	Datasets								
	Ta-En			Ma-En			Ka-En		
	train	dev	test	train	dev	test	train	dev	test
Positive	20,070	2,257	2,546	6,421	706	780	2,823	321	374
Negative	4,271	480	477	2,105	237	258	1,188	139	157
Mixed_Feelings	4,020	438	470	926	102	134	574	52	65
Unknown_state	5,628	611	665	5,279	580	643	711	69	62
Other languages	1,667	176	244	1,157	141	147	916	110	110
Total	35,657	3,962	4,402	15,888	1,766	1,962	6,212	691	768

Table 5
Results for the Development set

Feature set	Language pair	Classifiers	F1-score
Char n-grams	Ta-En	LR	0.59
		LSVM	**0.60***
		MLP	0.58
	Ma-En	**LR**	**0.75***
		LSVM	0.74
		MLP	0.71
	Ka-En	LR	0.65
		LSVM	**0.66***
		MLP	0.63
Char sequences n-grams	Ta-En	**LR**	**0.59**
	Ma-En	LR	0.73
		LSVM	**0.74**
		MLP	0.69
	Ka-En	LR	0.63
		LSVM	**0.64**
		MLP	0.60
Syllable n-grams	Ta-En	**LR**	**0.60***
	Ma-En	**LR**	**0.74**
		LSVM	0.73
		MLP	0.70
	Ka-En	**LR**	**0.66***
		LSVM	0.64
		MLP	0.62

classifiers for Ka-En language pair.

The predictions on the Test set submitted by the participants were evaluated based on the average weighted F1-scores. Organizers had encouraged the teams to evaluate the models locally on Dev set and then to submit the predictions on the Test set. Table 5 gives the performances of proposed methodology on the Dev set for all the three feature sets using the three classifiers for

Table 6
Results for the Test set

Language	Feature set and Classifier	Precision	Recall	F1-score	Rank
Ta-En	Char + LSVM	0.620	0.655	0.616	-
	Char seq. + LR	0.598	0.609	0.603	-
	Syllable + LR	0.602	0.622	0.609	-
	Majority Voting	**0.612**	**0.644**	**0.619**	**5**
Ma-En	Char + LR	0.723	0.728	0.721	-
	Char seq. + LSVM	0.719	0.725	0.720	-
	Syllable + LR	0.715	0.720	0.712	-
	Majority Voting	**0.726**	**0.733**	**0.726**	**4**
Ka-En	Char + LSVM	0.622	0.650	0.622	-
	Char seq. + LSVM	0.614	0.652	0.624	-
	Syllable + LR	0.615	0.634	0.618	-
	Majority Voting	**0.622**	**0.655**	**0.628**	**2**

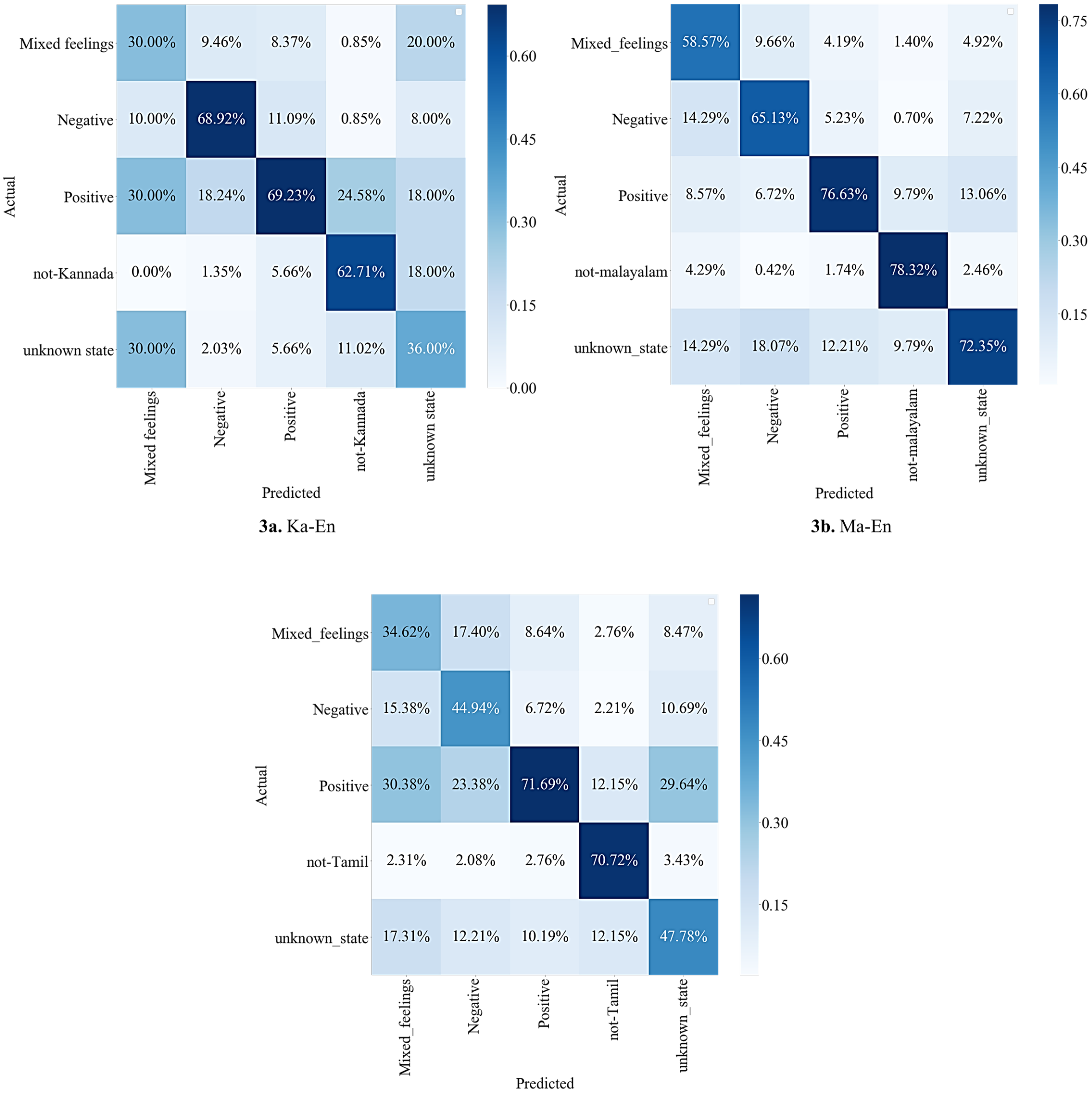
all the three language pairs. Observation of the results on the Dev set shows that LR and LSVM outperform each other for various feature sets and language pairs, while MLP always obtained the lowest results for all feature sets and all language pairs. The highlighted content in Table 5 correspond to the best performing classifier and the starred (*) score indicates the best feature set and classifier pair for the language pair. It can be seen that most of high performances are obtained with char n-grams followed by syllable n-grams. However, results using char sequences are interesting as well.

The good performance of syllable n-grams reveals that they can be effectively used as features in TC tasks as well and it is expected that they perform much better for native scripts as compared to code-mixed texts. Due to hardware resource constraints, only LR classifier was trained with char sequences and syllable n-grams for Ta-En language pair.

According to the performances of the models on the Dev set (highlighted scores in Table 5), the best feature set and classifier pair are selected and applied on the Test sets. The results of the best individual classifier and feature set pairs and their majority voting on the Test sets are given in Table 6. It can be observed that the performance of the majority voting of the predictions outperformed the performances of the individual classifiers. The results released by the shared task organizers in the leaderboard[12] reveals that our proposed methodology using majority voting of the predictions obtained 2nd, 4th, and 5th ranks with average weighted F1-scores of 0.628, 0.726, and 0.619 for Ka-En, Ma-En and Ta-En language pairs respectively.

The confusion matrix for each language pair based on the best performances as mentioned in Table 6 are presented in Figure 3. For both Ka-En (Figure 3a) and Ta-En (Figure 3c) language pairs, the weakest performances are for predicting "Mixed_feelings" comments and the best performances are for predicting "Positive" comments. Similarly, for Ma-En (Figure 3b) language pair, the weakest performance is for predicting "Mixed_feelings" comments and the good performance is for predicting "not-Malayalam" comments along with "Positive" comments. Though predicting "Mixed_feelings" comments exhibits weakest performance in all the three language pairs, the results of Ma-En language pair are higher compared to that in other two

[12] https://dravidian-codemix.github.io/2021/proceedings.html



3a. Ka-En

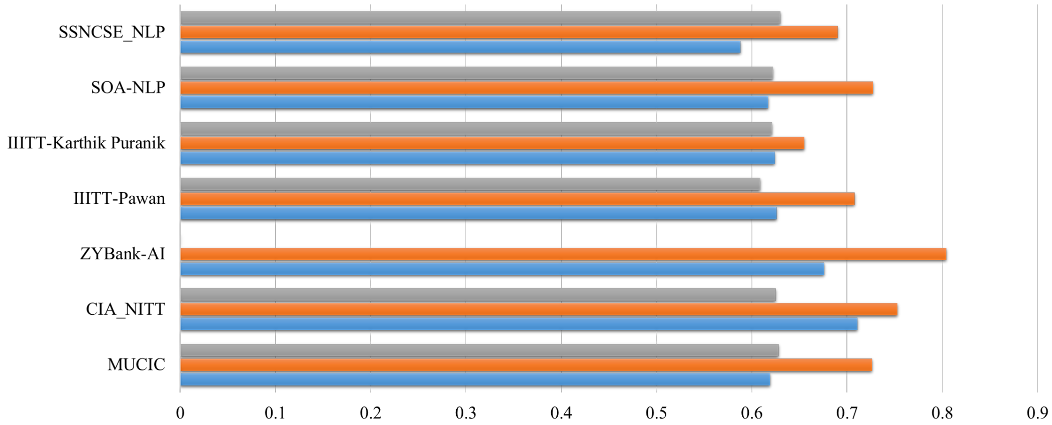
3b. Ma-En

3c. Ta-En

Figure 3: Confusion matrix

language pairs.

The comparison of the performances of the proposed methodology with that of the top performing models in the shared task shown in Figure 4 illustrates that the performances are quite competitive for all the language pairs. Ka-En language pair which has a smaller dataset compared to other language pairs also has given good performance.



	MUCIC	CIA_NITT	ZYBank-AI	IIITT-Pawan	IIITT-Karthik Puranik	SOA-NLP	SSNCSE_NLP
■ Ka-En	0.628	0.625	0	0.609	0.621	0.622	0.63
■ Ma-En	0.726	0.753	0.804	0.708	0.655	0.727	0.69
■ Ta-En	0.619	0.711	0.676	0.626	0.624	0.617	0.588

Figure 4: Comparison of the performances of the proposed models with the top performing models in shared task

5. Conclusion and Future Work

This paper describes the participation of our team MUCIC in SA shared task at Dravidian-CodeMix-FIRE2021. Three types of features, namely: char, char sequences and syllables are extracted from the given texts. These features are used to generate corresponding n-grams which are then transformed to TFIDF vectors for training the classifiers. According to the performances of the models on the Dev set, the best feature set and classifier pair are selected and applied on the Test sets and the majority voting of their predictions were submitted to the shared task organizers for evaluation. The results on the leaderboard reveals that our proposed strategy obtained promising results and secured 2nd, 4th, and 5th ranks with average weighted F1-scores of 0.628, 0.726, and 0.619 for Ka-En, Ma-En and Ta- En language pairs respectively. Other features and feature selection algorithms will be explored further for code-mixed low resource Dravidian languages.

Acknowledgments

Team MUCIC sincerely appreciate the organizers for their efforts to conduct this shared task.

References

[1] F. Balouchzahi, H. Shashirekha, LA-SACo: A Study of Learning Approaches for Sentiments Analysis in Code-Mixing Texts, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, 2021, pp. 109–118.

[2] F. Balouchzahi, H. L. Shashirekha, MUCS@Dravidian-CodeMix-FIRE2020: SACO-Sentiments Analysis for CodeMix Text, in: P. Mehta, T. Mandl, P. Majumder, M. Mitra (Eds.), 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. 495–502.

[3] 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, in: Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation, Hyderabad, India, December 16-20, 2020, volume 2826 of *CEUR Workshop Proceedings*, 2020, pp. 21–24.

[4] N. Jose, B. R. Chakravarthi, S. Suryawanshi, E. Sherly, J. P. McCrae, A Survey of Current Datasets for Code-Switching Research, in: 2020 Sixth International Conference on Advanced Computing and Communication Systems (ICACCS), IEEE, 2020, pp. 136–141.

[5] B. R. Chakravarthi, R. Priyadharshini, S. Thavareesan, D. Chinnappa, D. Thenmozhi, E. Sherly, J. P. McCrae, A. Hande, R. Ponnusamy, S. Banerjee, C. Vasantharajan, Findings of the Sentiment Analysis of Dravidian Languages in Code-Mixed Text, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.

[6] R. Priyadharshini, B. R. Chakravarthi, S. Thavareesan, D. Chinnappa, T. Durairaj, E. Sherly, Overview of the Dravidian CodeMix 2021 Shared Task on Sentiment Detection in Tamil, Malayalam, and Kannada, in: Forum for Information Retrieval Evaluation, FIRE 2021, Association for Computing Machinery, 2021.

[7] 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 First 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. 177–184. URL: https://aclanthology.org/2020.sltu-1.25.

[8] 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), 2019, pp. 4171–4186.

[9] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, É. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised Cross-lingual Representation Learning at Scale, in: Proceedings of the 58$^{\text{th}}$ Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8440–8451.

[10] B. R. Chakravarthi, R. Priyadharshini, N. Jose, T. Mandl, P. K. Kumaresan, R. Ponnusamy, R. Hariharan, J. P. McCrae, E. Sherly, et al., 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, 2021, pp. 133–145.

[11] T. Pires, E. Schlinger, D. Garrette, How Multilingual is Multilingual BERT?, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 4996–5001.

[12] R. Sun, X. Zhou, SRJ@ Dravidian-CodeMix-FIRE2020: Automatic Classification and

Identification Sentiment in Code-mixed Text, in: FIRE (Working Notes), 2020, pp. 548–553.

[13] A. Conneau, G. Lample, Cross-lingual Language Model Pretraining, Advances in Neural Information Processing Systems 32 (2019) 7059–7069.

[14] X. Ou, H. Li, YNU@ Dravidian-CodeMix-FIRE2020: XLM-RoBERTa for Multi-language Sentiment Analysis, in: FIRE (Working Notes), 2020, pp. 560–565.

[15] H. Sun, J. Gao, F. Sun, HIT_SUN@ Dravidian-CodeMix-FIRE2020: Sentiment Analysis on Multilingual Code-Mixing Text Base on BERT, in: FIRE (Working Notes), 2020, pp. 517–521.

[16] B. Huang, Y. Bai, LucasHub@ Dravidian-CodeMix-FIRE2020: Sentiment Analysis on Multilingual Code Mixing Text with M-BERT and XLM-RoBERTa, in: FIRE (Working Notes), 2020, pp. 574–581.

[17] Y. Zhu, K. Dong, YUN111@ Dravidian-CodeMix-FIRE2020: Sentiment Analysis of Dravidian Code Mixed Text, in: FIRE (Working Notes), 2020, pp. 628–634.

[18] N. Kanwar, M. Agarwal, R. K. Mundotiya, PITS@ Dravidian-CodeMix-FIRE2020: Traditional Approach to Noisy Code-Mixed Sentiment Analysis, in: FIRE (Working Notes), 2020, pp. 541–547.

[19] I. TOMEK, Two Modifications of CNN, IEEE Trans. Systems, Man and Cybernetics 6 (1976) 769–772.

[20] J. P. Posadas-Durán, I. Markov, H. Gómez-Adorno, G. Sidorov, I. Batyrshin, A. Gelbukh, O. Pichardo-Lagunas, Syntactic N-grams as Features for the Author Profiling Task, in: CEUR Workshop Proceedings, 2015.

[21] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, L. Chanona-Hernández, Syntactic Dependency-based n-grams: More Evidence of Usefulness in Classification, in: International Conference on Intelligent Text Processing and Computational Linguistics, Springer, 2013, pp. 13–24.

[22] K. De Jong, Temporal Constraints and Characterising Syllable Structuring, Phonetic Interpretation. Papers in Laboratory Phonology VI (2003) 253–268.

[23] G. Sidorov, Automatic Authorship Attribution using Syllables as Classification Features, Rhema. (2018).