MUCS@DravidianLangTech-EACL2021:COOLI-Code-Mixing Offensive Language Identification

F. Balouchzahi
Dept. of Computer Science,
Mangalore University,
Mangalore - 574199, India
frs_b@yahoo.com

B. K. Aparna
Dept. of Computer Science,
Mangalore University,
Mangalore - 574199, India
aparnabk14@gmail.com

H. L. Shashirekha
Dept. of Computer Science,
Mangalore University,
Mangalore - 574199, India
hlsrekha@gmail.com

Abstract

This paper describes the models submitted by the team MUCS for Offensive Language Identification in Dravidian Languages-EACL 2021 shared task that aims at identifying and classifying code-mixed texts of three language pairs namely, Kannada-English (Kn-En), Malayalam-English (Ma-En), and Tamil-English (Ta-En) into six predefined categories (5 categories in Ma-En language pair). Two models, namely, COOLI-Ensemble and COOLI-Keras are trained with the char sequences extracted from the sentences combined with words in the sentences as features. Out of the two proposed models, COOLI-Ensemble model (best among our models) obtained first rank for Ma-En language pair with 0.97 weighted F1-score and fourth and sixth ranks with 0.75 and 0.69 weighted F1-score for Ta-En and Kn-En language pairs respectively.

1 Introduction

Along with the increasing developments on social media, social networks, and Internet, the number of people using these are also increasing. The advantage of social media is that users' have the freedom of expressing their opinions without revealing their identity (Thavareesan and Mahesan, 2019, 2020a,b). This feature is exploited by miscreants for spreading offensive messages targeting an individual or a group. These messages will be usually code-mixed texts in native languages mixed with English words but written in Roman script. The texts on social media do not adhere to the rules of any of the languages in which they are written. Hence, the analysis of code-mixed texts is a more challenging task compared to analysis of texts in native scripts because of the inconsistent Romanization conventions and non-standard grammars in code-mixed texts (Riyadh and Kondrak, 2019). These issues have created a demand for analyzing social media text which is becoming important day by day.

Generally, social media analysis and Offensive Language Identification (OLI) is fathomably important for social media platforms to monitor the texts including hateful or offensive content or advertising violence against people, communities, or religions. Many studies have been carried out in this direction to identify offensive content in texts. But, most of these works focus on rich resource languages such as English, Spanish, etc. giving less or no importance for low resource languages such as Kannada, Tamil, Telugu, Malayalam (Chakravarthi et al., 2020c; Mandl et al., 2020). However, most texts in social media are not written in any one language, but a mix of several languages making the OLI task more challenging (Arora, 2020). Further, the task becomes more complex due to the usage of non-native scripts. Indians usually mix English language with their native language and use Roman script mixed with their native language script to post messages. This type of mixing two or more languages in a text is called Code-Mixing (Chakravarthi, 2020; Jose et al., 2020; Priyadharshini et al., 2020). Due to lack of efficient keyboards for native languages or ease of using Roman script and better adaptation of Roman script in software and smartphones, code-mixing texts are increasing day-by-day. To promote analyzing code-mixing texts to identify offensive language posts in Dravidian languages, "Offensive Language Identification in Dravidian Languages" (Chakravarthi et al., 2021) shared task provides texts in three code-mixing language pairs namely, Kannada-English (Kn-En), Malayalam-English (Ma-En), and Tamil-English (Ta-En).

The datasets provided by DravidianLangTech[1]

[1] https://dravidianlangtech.github.io/

Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pages 323–329

include code-mixing texts labeled into one of six categories namely, "not-offensive, offensive-untargeted, offensive-targeted-individual, offensive-targeted-group, offensive-targeted-other, and not-in-indented-language" ("offensive-targeted-other" is not included for Ma-En language pair). This paper describes the models submitted by team MUCS to "Offensive Language Identification in Dravidian Languages" (Chakravarthi et al., 2021) shared task. The two models COOLI[2]-Ensemble and COOLI-Keras are trained using char sequences extracted from sentences combined with words in the sentences. COOLI-Ensemble is a voting classifier obtained by ensembling three estimators namely, Multi-Layer Perceptron (MLP), eXtreme Gradient Boosting (XGB), and Logistic Regression (LR) that predict a final tag based on a hard majority voting configuration. COOLI-Keras is a simple architecture based on Neural Network (NN) using Keras sequential model.

The rest of paper is organized as follows: Section 2 describes the recent literature on code-mixed text processing, and Section 3 presents details of submitted models followed by the results in Section 4. Conclusion and future plans are given in Section 5.

2 Related Work

Researchers have proposed many models for OLI for several languages. However, very few works are reported for OLI task in code-mixed texts. Some of the recent literature related to OLI in Dravidian code-mixed texts (Mandl et al., 2020) are given here: A shared task on OLI conducted by (Mandl et al., 2020), on Dravidian code-mixed texts consists of two message level subtasks namely, subtask A and subtask B. While the focus of subtask A was to classify YouTube comments in Malayalam-English code-mixed texts subtask B focused on the classification of Romanized Twitter comments in Tamil-English and Malayalam-English code-mixed texts. The datasets used in this shared task are described in (Chakravarthi et al., 2020b) (Chakravarthi et al., 2020a).

(Renjit and Idicula, 2020) proposed a binary classification model for subtask B to classify Malayalam-English code-mixed texts into "offensive" and "not offensive" posts. Their model consists of a text processing step which includes removing English stopwords, hashtags, URLs, and emojis, converting text to lower case, and tokenization. Using Keras embedding they represented text as one-hot encoding with 50D and fed it to two LSTM architectures. The first LSTM architecture comprised of an LSTM layer and recurrent dropout (0.2) followed by a dense layer that was configured with sigmoid activation and binary cross-entropy and the second LSTM architecture is same as the first architecture with three dense layers and Relu activation added in between LSTM and dense layer. They obtained 0.53 and 0.48 macro F1-score using first and second architectures respectively.

A Universal Language Model Fine-Tuning (ULMFiT) based on Transfer Learning (TL) approach submitted by (Arora, 2020) trained a language model synthetically for Malayalam –English code-mixed texts using fastai library from Malayalam –English code-mixed data collected from Wikipedia articles in native script as well as the translated and transliterated versions. Fastai library was also used to build final classifiers after a step of fine-tuning the pre-trained language model using training set. They obtained 0.91 and 0.74 weighted F1-score on subtasks A and B respectively.

Another work in this shared task submitted by (Ghanghor et al., 2021) is based on TL and Machine Learning (ML) approaches for subtask A. In addition to the dataset provided by task organizers they used the OLID dataset (Zampieri et al., 2019a), a well-known dataset that has been used in the SemEval-2019 Task 6 (OffensEval) (Zampieri et al., 2019b) in TL model and explored BERT and XLM-ROBERTA by adopting implementation from HuggingFace's transformer models (Wolf et al., 2019). They trained ML classifiers namely, Multinomial Naive Bayes (MNB), Support Vector Machines (SVM), and Random Forest (RF) on vectors obtained from bag-of-words feature engineering method. The reported results illustrate that RF with a 0.93 weighted F1-score outperformed other models. However, XLM-ROBERTA based on TL obtained 0.89 weighted F1-score and 5thplace in subtask A.

3 Methodology

The methodology includes two major steps for each proposed models, namely, Feature Extraction and Classification Models. Details of the steps are given below:

2021

[2]COOLI stands for Code-Mixing Offensive Language Identification

Input text	Extracted features
"yuvanvera level ya" (Ta-En)	yu, uv, va, an, n_, _v, ve, er, ra, a_, _l, le, ev, ve, el, l_, _y, ya, yuv, uva, van, an_, _ve, ver, era, ra_, _le, lev, eve, vel, el_, _ya, yuva, uvan, van_, _ver, vera, era_, _lev, leve, evel, vel_, yuvan, uvan_, _vera, vera_, _leve, level, evel_, yuvan_, _vera_, _level, level_, yuvanvera, level, ya

Table 1: An example of a text and its char sequences

3.1 Feature Extraction

The feature extraction module pre-processes the input texts and extracts a set of char sequences and words as features which are then converted to vectors using CountVectorizer[3] library.

Pre-processing includes converting emojis to text (using emoji library[4]), removing punctuations, words of length less than 2, unwanted characters (such as !()-[];:'" ¡¿./?$=% +@*_ ', etc.), and finally converting text to lower case.

The everygrams [5] function of NLTK library is used to generate char sequences of length 2 to 6 from texts and then the extracted char sequences are combined with tokenized words of text. An example of a text and its char sequences are shown in Table 1. The combination of features are converted to vectors using CountVectorizer library and fed to the classification models. Figure 1 shows the Feature Extraction steps in the proposed methodology.

3.1.1 Feature Selection

Text is high dimensional in nature as every word is considered as a feature. The char sequences extracted in feature extraction module adds further to this dimensionality increasing the complexity of the algorithms processing these texts (Shashirekha et al., 2020). Hence, reducing the features becomes important to reduce the complexity of the algorithms. Feature selection has gained importance since it reduces the number of features by eliminating redundant and irrelevant features that is expected to improve the performance of the algorithm (Gao et al., 2017).

Inspired by (Giglou et al., 2019), a feature selection module that consists of three feature selection algorithms namely Chi-Square test, Mutual Information (MI), and F test is adopted from

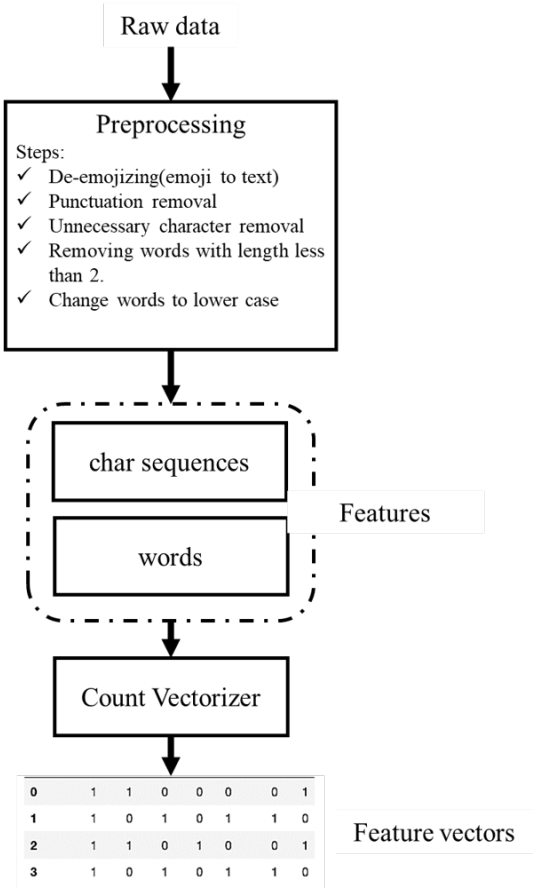

Figure 1: Feature engineering module

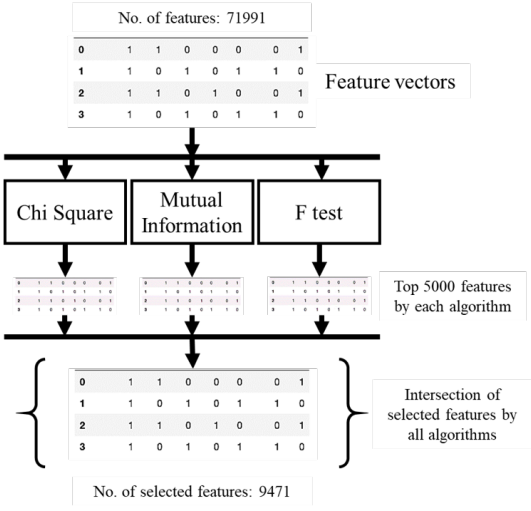

Figure 2: Feature selection module

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

[4]https://pypi.org/project/emoji

[5]https://www.kite.com/python/docs/nltk.everygrams

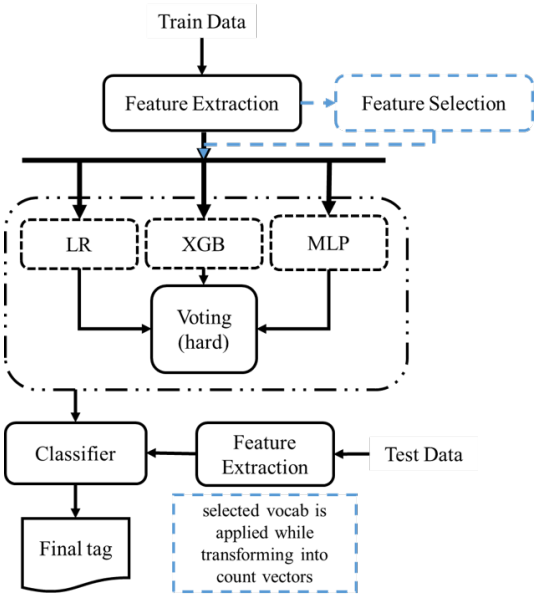

Figure 3: Structure of COOLI-Ensemble model (the dashed lined path is applied only for Kn-En texts)

(Shashirekha et al., 2020). The feature selection module utilizes a statistical measure to assign a score to features and based on these scores features are ranked and top k features are selected (in this case top 5000 features) as most relevant features for further processing.

Each of the three feature selection algorithms are used to select top 5000 ranked features and an intersection of these selected features are considered as final features. Figure 2 gives an overview of feature selection module. This feature selection is applied only for Ka-En code-mixed texts. Figure 2 illustrates that 71991 features get reduced to 9471.

Due to technical problems and limitation on RAM and GPU to process huge datasets in the machine that was used for processing the task, Feature selection algorithms could not be applied for Ta-En and Ma-En texts.

3.2 Classification Models

We propose two classification models namely, COOLI-Ensemble and COOLI-Keras which are trained on the features obtained in feature extraction module. The size of obtained vectors that are fed as input to the models depend on datasets and 71,991, 326,628 and 187,810 features were obtained for Ka-En, Ta-En, and Ma-En texts respectively. The proposed models are described below:

3.2.1 COOLI-Ensemble

It is a Voting Classifier (VC) with three sklearn[6] estimators given below:

Multi-Layer Perceptron[7] (MLP): is one of the Neural Network (NN) models that are widely used in the field of ML due to its simplicity compared to the recent complex NN models. MLP is also defined as a supplement for feed-forward NN that consists of three types of layers namely, the input layer, output layer, and hidden layer [16].In the proposed model, one layer per type (input layer, hidden layer, and output layer) is used and layer sizes are set to (150, 100, 50) and maximum iteration, activation, solver[8](that specifies the algorithm for weight optimization across the nodes), and random state have been set to 300, Relu, Adam, and 1 respectively.

eXtreme Gradient Boosting (XGB): is one of Gradient boosting classifier that use Gradient boosting technique to solve classification and regression problems. The major advantage of "boosting" is producing a robust classifier by converting a set of learners that perform poor (Athanasiou and Maragoudakis, 2017). Adaptive Boosting (AdaBoost), Gradient Boosting (GBM) are other classifiers in boosting family. In this study, XGB classifier from Sklearn with the following configuration is used: max_depth is set to 20, and n_estimators, learning rate, colsample_bytree, gamma, reg_alpha, and objective are set to 80, 0.1, 0.7, 0.01, 4, 'multi: softmax' respectively.

Logistic Regression (LR): is one of the most popular binary classifier that is driven from the field of statistics. However, it utilizes the one-vs-rest (OvR) scheme to deal with multi-class classification tasks. LR classifier has been used with default parameters.

Figure 3 represents the structure of COOLI-Ensemble. Features are extracted from the training set as described in feature extraction module and are used to construct the classifier. Features from test data are also extracted using feature extraction module and only those features in the feature set are considered for classification. In case of Ka-En code-mixed texts, feature selection is applied to get reduced feature set. The COOLI-Ensemble is configured as hard voting.

[6]https://scikit-learn.org/stable/

[7]https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

[8]nnabla.readthedocs.io

3.2.2 COOLI-Keras

The feature extraction and feature selection steps described earlier are used to train COOLI-Keras model which is a Keras[9] dense NN architecture adopted from

https://www.kaggle.com/ismu94/tf-idf-deep-neural-net

COOLI-Keras model has been trained for 40 epochs with a batch size of 128. The main diference between MLP in COOLI-Ensemble and COOLI-Keras model is in layers and configuration of them. Figure 4 describes the overview of COOLI-Keras model with all NN layers and configurations.

4 Experimental Results

4.1 Datasets

Datasets provided by the organizers for this study includes code-mixed texts in three Language Pairs (LP), namely, Ma-En, Ta-En, Ka-En and the details are given in (Chakravarthi et al., 2021). Texts are distributed into 6 categories, namely, "Not-offensive (NO), offensive-untargeted (OU), offensive-targeted-individual (OTI), offensive-targeted-group (OTG), offensive-targeted-other (OTO), and Not-in-indented-language (NIL)". "offensive-targeted-other" is skipped for Ma-En language pair. The distribution of train, development (Dev.) and test sets is given in Table 2.

Statistics of the datasets shown in Table 2 illustrates that Ka-En texts are less than other two language pairs and it can affect the performance of the proposed models. Further, all datasets are imbalanced. However, as the percentage of imbalance sounds to be less in Ma-En dataset and also having more text in Malayalam native script, it is expected that the proposed models performs better for this dataset compared to other two datasets.

4.1.1 Results

The results obtained for each model on the test set of each language pairs are evaluated by weighted Precision, Recall, and F1-score using Sklearn.metrics library. For Ka-En language pair the models are constructed with Feature Selection (with FS) and without Feature Selection (No FS).

The results obtained by COOLI-Ensemble (along with obtained ranks) and COOLI-Keras

[9]https://keras.io/

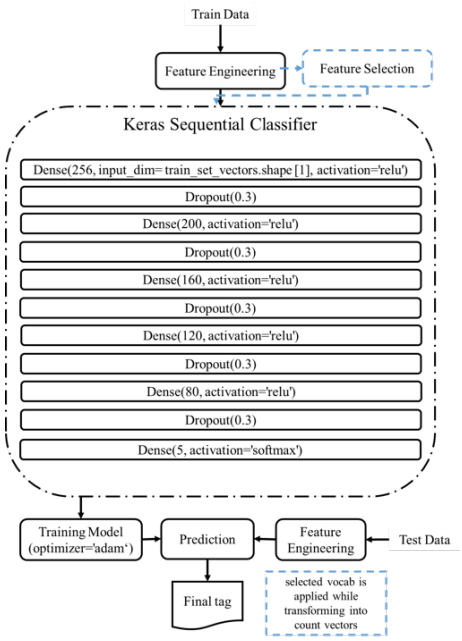

Figure 4: Structure of COOLI-Keras model (the dashed lined path is applied only for Kn-En texts)

Train Set			
tag	Ma-En	Ka-En	Ta-En
NO	14153	3544	25425
OU	1287	1522	1454
OTI	140	329	2557
OTG	239	487	2343
OTO	191	212	2906
NIL	-	123	454
Dev. Set			
tag	Ma-En	Ka-En	Ta-En
NO	1779	426	3193
OU	163	191	172
OTI	13	45	295
OTG	24	66	307
OTO	20	33	356
NIL	-	16	65
Test Set			
tag	Ma-En	Ka-En	Ta-En
NO	1765	427	3190
OU	157	185	160
OTI	23	44	288
OTG	27	75	315
OTO	29	33	368
NIL	-	14	71

Table 2: Datasets for each Language Pairs

COOLI-Ensemble					
Language Pair		P	R	F1	Rank
Ma-En		**0.97**	**0.97**	**0.97**	**1**
Ta-En		**0.74**	**0.77**	**0.75**	**4**
Ka-En	**With FS**	**0.68**	**0.72**	**0.69**	**6**
	No FS	0.68	0.71	0.68	-
COOLI-Keras model					
Language Pairs		P	R	F1	
Ma-En		0.96	0.96	0.96	
Ta-En		0.73	0.74	0.73	
Ka-En	With FS	0.68	0.72	0.69	
	No FS	0.68	0.68	0.68	

Table 3: Results of proposed models

models are shown in Table 3 and the results illustrate that COOLI-Ensemble model (best among our models) outperformed COOLI-Keras model for Ma-En and Ta-En language pairs. However, the performance on Kn-En language pair by both the models remains more or less the same. Further, it can be observed that both the models with feature selection for KA-En dataset performed slightly better compared to other models with all the features. Hence, applying feature selection module for rest of language pairs could improve their performance. As it was expected both models performed better for Ma-En texts due to nature of dataset that includes more number of texts in Malayalam native script that results in distinguishing of not Malayalam texts.

5 Conclusion and Future Work

This paper describes the two models, COOLI-Ensemble and COOLI-Keras submitted for "Offensive Language Identification in Dravidian Languages-EACL 2021" shared task to classify code-mixed text in Tamil-English, Malayalam-English, and Kannada-English. The analysis of results shows that COOLI-Ensemble (best among our models) on a feature set of char sequences, and words outperformed COOLI-Keras model for both Ma-En and Ta-En language pairs and also obtained first rank for Ma-En language pair with 0.97 weighted F1-score and fourth and sixth ranks with 0.75 and 0.69 weighted F1-score for Ta-En and Kn-En language pairs respectively. The results also illustrate that feature selection applied on Ka-En texts slightly improved the performance of models. As a future work, we planned to explore different feature sets and feature selection models along with various learning approaches to process code-mixed texts.

References

Gaurav Arora. 2020. Gauravarora@ HASOC-Dravidian-CodeMix-FIRE2020: Pre-training ULMFiT on Synthetically Generated Code-Mixed Data for Hate Speech Detection. *arXiv preprint arXiv:2010.02094*.

Vasileios Athanasiou and Manolis Maragoudakis. 2017. A novel, gradient boosting framework for sentiment analysis in languages where nlp resources are not plentiful: a case study for modern greek. *Algorithms*, 10(1):34.

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

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

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

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

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

Weihao Gao, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. 2017. Estimating mutual information for discrete-continuous mixtures. *arXiv preprint arXiv:1709.06212*.

Nikhil Kumar Ghanghor, Rahul Ponnusamy, Prasanna Kumar Kumaresan, Ruba Priyadharshini, Sajeetha Thavareesan, and Bharathi Raja Chakravarthi. 2021. IIITK@LT-EDI-EACL2021: Hope Speech Detection for Equality, Diversity, and Inclusion in Tamil, Malayalam and English. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, Online.

Hamed Babaei Giglou, Mostafa Rahgouy, Taher Rahgooy, Mohammad Karami Sheykhlan, and Erfan Mohammadzadeh. 2019. Author profiling: Bot and gender prediction using a multi-aspect ensemble approach. In *CLEF (Working Notes)*.

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

Thomas Mandl, Sandip Modha, Anand Kumar M, and Bharathi Raja Chakravarthi. 2020. Overview of the HASOC Track at FIRE 2020: Hate Speech and Offensive Language Identification in Tamil, Malayalam, Hindi, English and German. In *Forum for Information Retrieval Evaluation*, FIRE 2020, page 29–32, New York, NY, USA. Association for Computing Machinery.

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

Sara Renjit and Sumam Mary Idicula. 2020. CUSATNLP@ HASOC-Dravidian-CodeMix-FIRE2020: Identifying Offensive Language from ManglishTweets. *arXiv preprint arXiv:2010.08756*.

Rashed Rubby Riyadh and Grzegorz Kondrak. 2019. Joint approach to deromanization of code-mixed texts. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 26–34.

H. L. Shashirekha, M. D. Anusha, and Nitin S. Prakash. 2020. Ensemble model for profiling fake news spreaders on twitter. In *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020*, volume 2696 of *CEUR Workshop Proceedings*. CEUR-WS.org.

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

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

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. HuggingFace's Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the type and target of offensive posts in social media. *arXiv preprint arXiv:1902.09666*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.