MUCS@LT-EDI-EACL2021:CoHope-Hope Speech Detection for Equality, Diversity, and Inclusion in Code-Mixed Texts

F. Balouchzahi, B. K. Aparna, H. L. Shashirekha
Department of Computer Science,
Mangalore University
frs_b@yahoo.com,aparnabk14@gmail.com,hlsrekha@gmail.com

Abstract

This paper describes the models submitted by the team MUCS for "Hope Speech Detection for Equality, Diversity, and Inclusion-EACL 2021" shared task that aims at classifying a comment / post in English and code-mixed texts in two language pairs, namely, Tamil-English (Ta-En) and Malayalam-English (Ma-En) into one of the three predefined categories, namely, "Hope_speech", "Non_hope_speech", and "other_languages". Three models namely, CoHope-ML, CoHope-NN, and CoHope-TL based on Ensemble of classifiers, Keras Neural Network (NN) and BiLSTM with Conv1d model respectively are proposed for the shared task. CoHope-ML, CoHope-NN models are trained on a feature set comprised of char sequences extracted from sentences combined with words for Ma-En and Ta-En code-mixed texts and a combination of word and char ngrams along with syntactic word ngrams for English text. CoHope-TL model consists of three major parts: training tokenizer, BERT Language Model (LM) training and then using pre-trained BERT LM as weights in BiLSTM-Conv1d model. Out of three proposed models, CoHope-ML model (best among our models) obtained 1st, 2nd, and 3rd ranks with weighted F1-scores of 0.85, 0.92, and 0.59 for Ma-En, English and Ta-En texts respectively.

1 Introduction

The recent wave of using social media especially during the outbreak of Covid-19 has increasingly affected the amount of user-generated data and text over the internet that has provided immense opportunities in automated text analysis and Computational Linguistics (Bohra et al., 2018). Most of tools and systems to analyze social media texts are designed to handle them in their native script. However, social media texts are often code-mixed, i.e., written in Roman script mixing English words rather than in the native script of language due to difficulty in using tools provided to pen the comments in native script (Jose et al., 2020; Priyadharshini et al., 2020) . Further, users may prefer using Roman scripts even though the language has its own standardized written form and script (Sitaram and Black, 2016). The analysis of Romanized and code-mixed texts is 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).

Hope speech detection is defined as analysis and detection of inspirational talk and comments/posts with positive vibes, etc. against people with not straight desires such as Lesbian, Gay, and Transgender or positive suggestion for Covid-19 guidelines, etc. (Chakravarthi, 2020). Even though a couple of studies and workshops are focused on analyzing code-mixed texts in tasks such as Sentiments Analysis (SA) and Offensive Language Identification (OLI) it has been rarely experimented on Hope Speech Detection even in native scripts. In this direction, the "Hope Speech Detection for Equality, Diversity, and Inclusion"[1] shared task aims at classifying a comment/post in English and code-mixed texts in two language pairs, namely, Tamil-English (Ta-En) and Malayalam-English (Ma-En) into one of the three predefined categories, namely, "Hope_speech", "Non_hope_speech", and "other_languages". The details of the datasets provided by organizers are given in (Chakravarthi, 2020).

In this paper, we, team MUCS describe the three models CoHope-ML, CoHope-NN and CoHope-TL submitted for "Hope Speech Detection for Equality, Diversity, and Inclusion" shared task. The char sequences extracted from sentences com-

[1] https://sites.google.com/view/lt-edi-2021

Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion, pages 180–187

bined with words in the sentences are used to train CoHope-ML and CoHope-NN models for code-mixed Ma-En and Ta-En texts whereas a combination of char and word ngrams along with syntactic ngrams are used to train the same models for English texts. CoHope-TL model is comprised of three major steps: (i) training tokenizer, (ii) training BERT LM using raw texts from Dakshina Dataset[2] [5], for Ma-En and Ta-En code mixed texts and pre-trained BERT LM from Kaggle[3] for English, and (iii) transferring obtained weights and building BiLSTM-Conv1d model.

The rest of paper is organized as follows: while Section 2 describes the recent literature on code-mixed text processing, Section 3 focuses on the description of the models submitted to the shared task followed by experiments and results in Section 4. Conclusion and future plans are included in Section 5.

2 Related Work

Researchers have developed a vast range of datasets, tools and models for Text Classification (TC). However, comparatively very less work has been done on the classification of code-mixed texts and the available literature focus on SA and OLI tasks for several languages pairs. Hope Speech detection is a new challenge that has been explored rarely. Some of recent studies on TC tasks for code-mixed texts are given below:

(Chakravarthi et al., 2020b) presents an overview of OLI shared task on code-mixed texts in Dravidian languages[4] consisting of two subtasks A and B to classify a given text into "offensive" or "not-offensive" categories. While Subtask A is to classify code-mixed Ma-En YouTube comments, SubTask B is to classify Romanized Malayalam and Romanized Tamil texts from YouTube or Twitter comments. Datasets used in this shared tasks are described in (Chakravarthi et al., 2020c) and (Chakravarthi et al., 2020a). Two models based on different configurations of LSTM proposed by (Renjit and Idicula, 2020) for the OLI shared task obtained a weighted F1-score of 0.53 for Romanized Malayalam text in Subtask B. A Universal LM has been trained for Ma-En code-mixed texts from Wikipedia articles in native script combined with translated and transliterated versions by (Arora, 2020). The authors transferred the obtained LM to TC model from fastai library to classify code-mixed texts in Ma-En and obtained 0.91, 0.74 weighted F1-score for Subtask A and Romanized Malayalam text of Subtask B respectively.

"Sentiment Analysis of Dravidian Languages in Code-Mixed Text"[5] which focuses on SA of code-mixed texts in Ta-En and Ma-En language pairs (Chakravarthi et al., 2020d) is another shared task on Dravidian languages. Datasets described in (Chakravarthi et al., 2020c) and (Chakravarthi et al., 2020a) are used in this shared task and they include five categories, namely, "Positive", "Negative", "Unknown_state", "Mixed-Feelings", and "Other_languages" for each language pairs. The overall results of this shared task reported in leaderboard illustrates that XLM-Roberta model proposed by (Sun and Zhou, 2020) with a weighted F1-score of 0.65 and 0.74 for Ta-En and Ma-En language pairs respectively obtained first rank for both subtasks. The proposed XLM-Roberta model uses extracted output of Convolution Neural Networks (CNN) which enables it to utilize the semantic information from texts. Another XLM-Roberta model proposed by (Ou and Li, 2020) ensembles pre-trained multi-language models and K-folding method to classify code-mixed texts. The proposed model with 0.63 and 0.74 weighted F1-score obtained third and first ranks on Ta-En and Ma-En language pairs respectively.

3 Methodology

The proposed models are described in terms of feature engineering to extract the required features followed by description of the classifiers.

3.1 Feature Engineering

Framework of the proposed methodology for CoHope-ML and CoHope-NN consists of a step of preprocessing the train and test data followed by feature engineering module to extract features and use them to train and test the models.

Preprocessing steps includes converting emojis to corresponding text (using emoji library[6]), removing punctuations, words of length less than 2, unwanted characters (such as !()-[];:'" ¡¿./?$=%+@*_ ', etc.) and converting text to lowercase.

Feature engineering module uses everygrams[7]

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

[3]https://www.kaggle.com/christofhenkel/pytorchpretrainedbert

[4]https://competitions.codalab.org/competitions/25295#learn_the_details

[5]https://dravidian-codemix.github.io/2020/index.html

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

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

Input text	Extracted features
“yuvanvera level ya.” (in 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: Examples of input text and extracted features for code-mixed texts

function from NLTK library to extract char sequences of length 3 to 6 from texts along with tokenized words for Ma-En and Ta-En language pairs as features. For English texts SNgramExtractor[8] library is used to extract syntactic ngrams of length 2 to 3 (Sidorov et al., 2013) (Posadas-Durán et al., 2015) in addition to traditional char ngrams of length 3 to 5 and word ngrams of length 1 to 3 as features. The extracted features are represented as TFIDF vectors for further processing. Tables 1 and 2 give samples of input texts and features extracted from the corresponding texts.

3.2 Models Description

The proposed models are described below:

3.2.1 CoHope-ML

There are various notions of ensemble learning such as bagging, stacking, etc. Due to simplicity and efficiency of bagging method, CoHope-ML model is developed as a hard voting classifier based on bagging by ensembling three sklearn[9] classifiers, Logistic Regression (LR), eXtreme Gradient Boosting (XGB) (Chen and Guestrin, 2016) and Multi-Layer Perceptron (MLP)[10]. Idea behind ensembling simple classifiers as estimators is to build a robust classifier utilizing the strength of each classifier. Parameters used for each estimator are given in Table 3. CoHope-ML model is trained on TFIDF vectors obtained in feature engineering module. The framework of CoHope-ML is shown in Figure 1.

[8]https://pypi.org/project/SNgramExtractor/

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

[10]https://scikit-learn.org/stable/modules/neural_networks_supervised.html

Input text	Extracted features
“Economic news have little effect on financial markets.” (in English)	Economic, news, have, little, effect, on, financial, markets., Economic news, news have, have little, little effect, effect on, on financial, financial markets., Economic news have, news have little, have little effect, little effect on, effect on financial, on financial markets., _Ec, Eco, con, ono, nom, omi, mic, ic_, _ne, new, ews, ws_, _ha, hav, ave, ve_, _li, lit, itt, ttl, tle, le_, _ef, eff, ffe, fec, ect, ct_, _on, on_, _fi, fin, ina, nan, anc, nci, cia, ial, al_, _ma, mar, ark, rke, ket, ets, ts., s._, _Eco, Econ, cono, onom, nomi, omic, mic_, _new, news, ews_, _hav, have, ave_, _lit, litt, ittl, ttle, tle_, _eff, effe, ffec, fect, ect_, _on_, _fin, fina, inan, nanc, anci, ncia, cial, ial_, _mar, mark, arke, rket, kets, ets., ts._, _Econ, Econo, conom, onomi, nomic, omic_, _news, news_, _have, have_, _litt, littl, ittle, ttle_, _effe, effec, ffect, fect_, _fina, finan, inanc, nanci, ancia, ncial, cial_, _mark, marke, arket, rkets, kets., ets._, news_Economic, have_news, effect_little, have_effect, effect_on, markets_financial, on_markets, have_.., effect_on_markets, on_markets_financial

Table 2: Examples of input text and extracted features for English texts

Estimators	Parameters
XGB	max_depth=20, n_estimators=80, learning_rate=0.1, colsample_bytree=.7, gamma=.01, reg_alpha=4, objective=’multi: softmax’
MLP	hidden_layer_sizes= (150,100,50), max_iter=300,activation = ’relu’, solver=’adam’, random_state=1
LR	Default parameters

Table 3: Parameters for estimators in CoHope-ML

3.2.2 CoHope-NN

The framework of CoHope-NN model is shown in Figure 2. It makes use of a Keras[11] dense Neural Network (NN) architecture adopted from

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

CoHope-NN model is trained for 40 epochs with a batch size of 128 on TFIDF vectors obtained from feature engineering module.

3.2.3 CoHope-TL

Based on TL, CoHope-TL adopts the architecture described in

https://huggingface.co/blog/how-to-train

to train Tokenizers and LMs using transformers for Ta-En and Ma-En language pairs. Tokenizer and LM for English are publicly available at:

https://www.kaggle.com/christofhenkel/torch-bert-weights

The steps involved in designing CoHope model are described below:

Training Tokenizer: Romanized text from Dakshina dataset (Roark et al., 2020) combined with code-mixed texts from (Chakravarthi et al., 2020c) and (Chakravarthi et al., 2020a) are preprocessed and used to train a byte-level Byte-pair encoding tokenizer[12] with a vocab size of 52000 words and min frequency of 2 (separately for each language pairs Ma-En and Ta-En). The resulting tokenizer is later used in training BERT LM.

Training BERT LM: BERT LM is trained using the trained tokenizer and raw texts used in previous step and transformers library[13] with following configurations:

- vocab_size=52_000
- max_position_embeddings=514
- num_attention_heads=12
- num_hidden_layers=6
- type_vocab_size=1

[11]https://keras.io/

[12]https://huggingface.co/transformers/tokenizer_summary.html

[13]https://pypi.org/project/transformers/

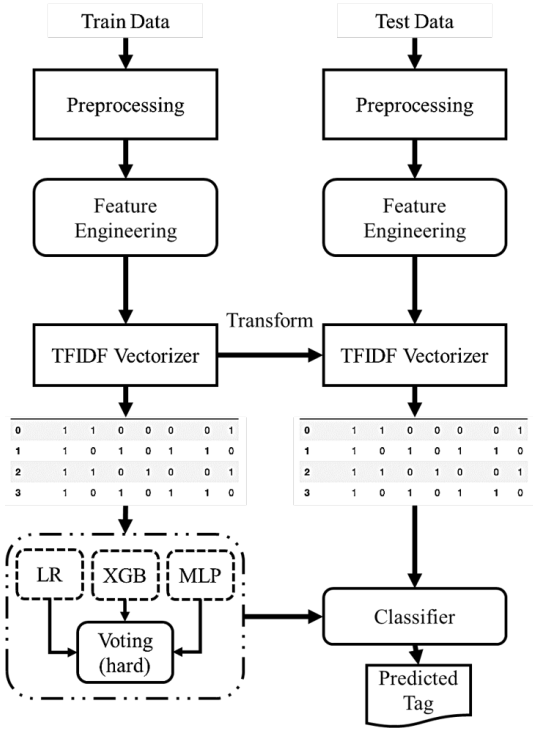

Figure 1: Framework of CoHope-ML model

The resulting LM is in turn trained for Ta-En and Ma-En language pairs separately and the weights are transferred for the construction of the classifier.

Model Construction: a BiLSTM-Conv1D architecture which is a BiLSTM model over convolutional layers with :

- Kernel size of 3
- Filter = 32
- MaxPooling1D with pool size of 2
- Length of words sequences = 250 with padding for short sentences

is used to train CoHope-TL model for 50 epochs with a batch size of 126. Table 4 gives summary of the layers in BiLSTM-Conv1D model and the frame work of CoHope-TL is shown in Figure 3.

4 Experimental Results

4.1 Datasets

Datasets used in this study include unannotated Romanized text from Dakshina (Roark et al., 2020) combined with texts from (Chakravarthi et al., 2020c), (Chakravarthi et al., 2020a) and annotated datasets provided by organizers which are

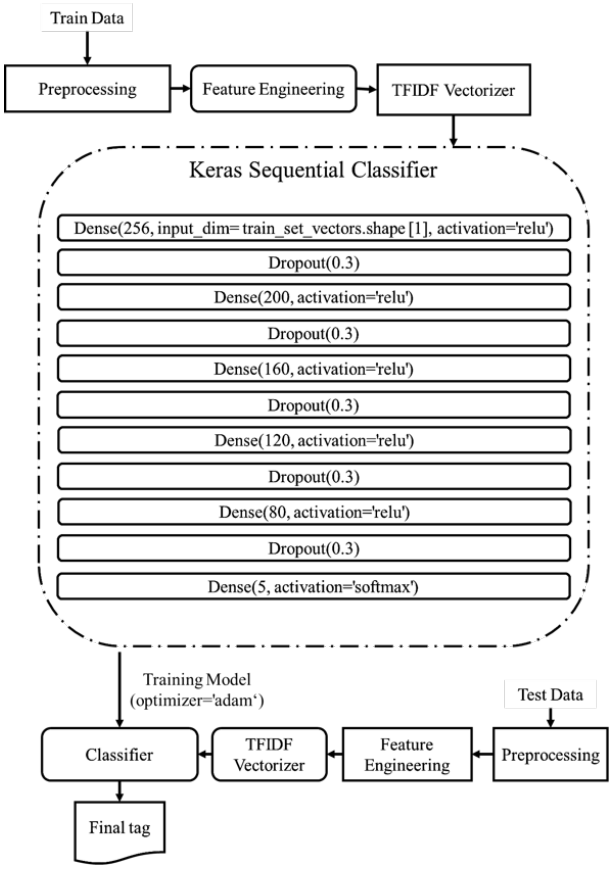

Figure 2: Framework of CoHope-NN model

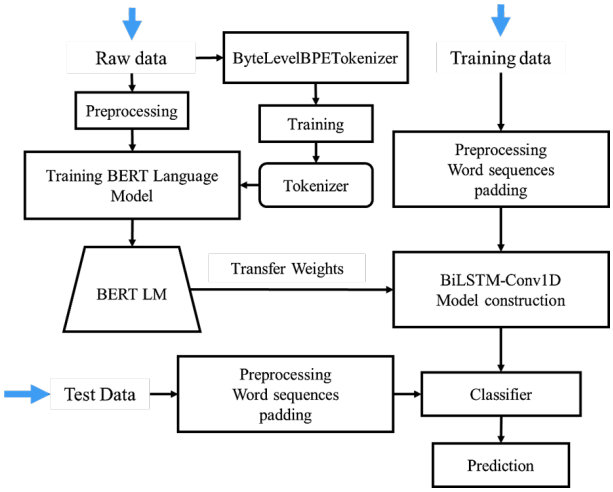

Figure 3: Framework of CoHope-TL

Layer(Type)	Output shape	
	Ta-En and Ma-EN	English
Embedding	(None,250,768)	(None,250,1024)
Conv1D	(None, 250, 32)	(None, 250, 32)
MaxPooling1D	(None, 125, 32)	(None, 125, 32)
Bidirectional	(None, 600)	(None, 600)
Dense	(None, 3)	(None, 3)

Table 4: Layers in BiLSTM-Conv1D

Set	LP	NO	HS	OL
Train	Ma-En	6205	1668	691
	Ta-En	7872	6327	1961
	English	20778	1962	22
Dev.	Ma-En	784	190	96
	Ta-En	998	757	263
	English	2569	272	2
Test	Ma-En	776	194	101
	Ta-En	946	815	259
	English	2593	250	3

Table 5: Label distribution over annotated datasets

described in (Chakravarthi, 2020). Statistics of the Tamil-English[14] (TaCo) and Malayalam-English[15] (MaCo) code-mixed raw texts are shown in Figure 4. It can be observed that MaCo code-mixed texts are noticeably less than TaCo code-mixed texts.

Annotated datasets include two code-mixed datasets Ta-En and Ma-En along with English datasets. Texts in the datasets for each Language Pairs (LP) are distributed in three categories namely, "Hope_speech (HS)", "Non_hope_speech (NO)", and "other_languages (OL)". Statistics of labels distribution in train, development (Dev.) and test sets and given in Table 5. It can be observed that as Ma-En code-mixed texts include significant number of samples in Malayalam native script and English text includes more samples, the proposed models are expected to perform better for Ma-En code-mixed texts and English texts compared to Ta-En code-mixed texts.

4.1.1 Results

Out of three proposed models, the results reported by organizers in leaderboard obtained 1st, 2nd, and 3rd ranks for Ma-En, English and Ta-En texts respectively for CoHope-ML model (best among our models). Comparison of weighted scores of all the models proposed by MUCS is shown in Table 6. As it is illustrated in Table 6, both CoHope-ML and CoHope-NN models utilizing char sequences, traditional n-grams and syntactic ngrams features outperformed the CoHope-TL model. The results also illustrate that models performed better for texts with more native scripts.

	P	R	F1	Rank
LP	CoHope-ML			
Ma-En	0.85	0.85	0.85	1
Ta-En	0.59	0.59	0.59	3
English	0.92	0.93	0.92	2
	CoHope-NN			
Ma-En	0.83	0.83	0.83	
Ta-En	0.57	0.57	0.56	
English	0.91	0.92	0.91	
	CoHope-TL			
Ma-En	0.79	0.76	0.77	
Ta-En	0.55	0.54	0.54	
English	0.90	0.90	0.90	

Table 6: Results of the proposed models

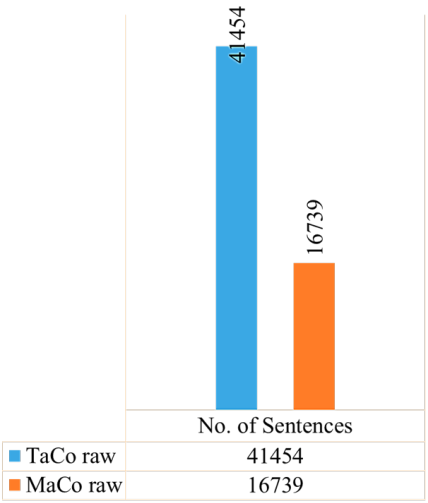

Figure 4: Statistics of raw texts

The Confusion Matrix (CM) for Ma-En, Ta-En, and English texts using CoHope-ML model are shown in Figures 5, 6 and 7 respectively. The confusion matrices illustrates that CoHope-ML model rarely gets confused between other languages and the intended language in Malayalam and English since both datasets are having significant number of samples in native scripts.

5 Conclusion and Future Work

In this paper we, team MUCS, present the description of three proposed models for the task of "Hope Speech Detection for Equality, Diversity,

[14]TaCo: texts from combination of Tamil-English with Romanized Tamil (Dakshina) datasets

[15]MaCo: texts from combination of Malayalam-English with Romanized Malayalam (Dakshina) datasets

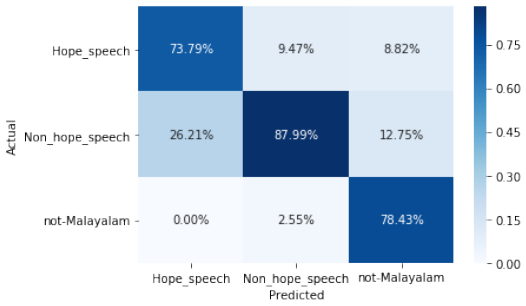

Figure 5: CM for Ma-En texts using CoHope-ML model

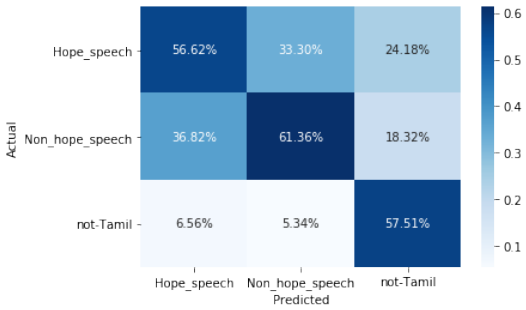

Figure 6: CM for Ta-En texts using CoHope-ML model

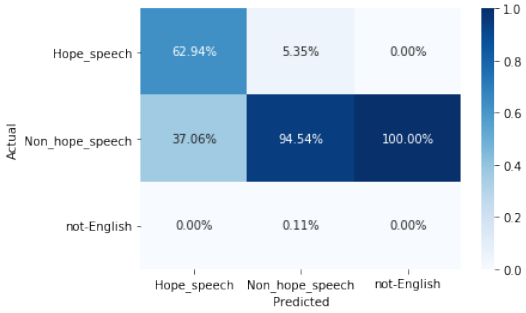

Figure 7: CM for English texts using CoHope-ML model

and Inclusion-EACL 2021". Proposed models includes a ML voting classifier - CoHope-ML, a DL NN model - CoHope-NN and a TL based model - CoHope-TL. The first two models are trained on a combination of char sequences and words for Ta-En and Ma-En code-mixed texts and combination of traditional char and word ngrams with syntactic word ngrams for English. CoHope-TL model utilizes BERT LM as weights in a BiLSTM-Conv1D architecture. Out of three proposed models, CoHope-ML model (best among our models) obtained weighted F1-scores of 0.85, 0.92 and 0.59 and 1, 2, 3 ranks for Malayalam-English, English, and Tamil-English texts. As future work, we planned to explore syntactic ngrams features for code-mixed texts and improve CoHope-NN architecture by experimenting on different NN layers and configurations. We also would like to compare different approaches based on TL for code-mixed texts from low resource languages.

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*.

Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A dataset of Hindi-English code-mixed social media text for hate speech detection. In *Proceedings of the second workshop on computational modeling of people's opinions, personality, and emotions in social media*, pages 36–41.

Bharathi Raja Chakravarthi. 2020. 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*, pages 41–53, Barcelona, Spain (Online). Association for Computational Linguistics.

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, M Anand Kumar, John Philip McCrae, Premjith B, Soman KP, and Thomas Mandl. 2020b. Overview of the track on HASOC-Offensive Language Identification-

DravidianCodeMix. In *Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2020). CEUR Workshop Proceedings. In: CEUR-WS. org, Hyderabad, India*.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John Philip McCrae. 2020c. 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, Vigneshwaran Muralidaran, Shardul Suryawanshi, Navya Jose, Elizabeth Sherly, and John P. McCrae. 2020d. 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.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

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.

X. Ou and H. Li. 2020. YNU@Dravidian-CodeMix-FIRE2020: XLM-RoBERTa for Multi-language Sentiment Analysis. In *Forum for Information Retrieval Evaluation*. CEUR Workshop Proceedings.

Juan-Pablo Posadas-Durán, Ilia Markov, Helena Gómez-Adorno, Grigori Sidorov, Ildar Batyrshin, Alexander Gelbukh, and Obdulia Pichardo-Lagunas. 2015. Syntactic n-grams as features for the author profiling task.

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.

Brian Roark, Lawrence Wolf-Sonkin, Christo Kirov, Sabrina J Mielke, Cibu Johny, Isin Demirsahin, and Keith Hall. 2020. Processing South Asian languages written in the Latin script: the Dakshina dataset. *arXiv preprint arXiv:2007.01176*.

Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona-Hernández. 2013. Syntactic dependency-based n-grams: More evidence of usefulness in classification. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 13–24. Springer.

Sunayana Sitaram and Alan W Black. 2016. Speech synthesis of code-mixed text. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3422–3428.

R. Sun and X. Zhou. 2020. SRJ @ Dravidian-CodeMix-FIRE2020:Automatic Classification and Identification Sentiment in Code-mixed Text. In *Forum for Information Retrieval Evaluation*. CEUR Workshop Proceedings.