Sentiment Classification on Bilingual Code-Mixed Texts for Dravidian Languages using Machine Learning Methods

Rashmi K.B., Guruprasad H.S., Shambhavi B.R.

Department of ISE, BMS College of Engineering, Bangalore, Karnataka, India

Abstract

Sentiment classification is a process of detecting the polarity of emotions. With the increased use of social media, people from all walks of life started communicating by using their local languages and, English as the common language resulted in an enormous amount of code-mixed data. Therefore, Code-mixed sentiment analysis is the trending research topic. This paper describes the Forum for Information Retrieval Evaluation (FIRE) 2021 shared task for message-level polarity classification. The system has to label it into positive, negative, neutral, mixed emotions, or not in the intended languages for the given code-mixed Dravidian dataset. The proposed work implements various machine learning classifiers namely, Logistic Regression, Balanced Random Forest, eXtreme Gradient Boosting (XGBoost), Random Forest, Support Vector Machine (SVM) as baseline algorithms for further ensemble learning. The proposed work achieved an accuracy of 0.57, 0.60, and 0.63 for code mixed Malayalam-English, Kannada-English, and Tamil-English test datasets respectively for the final ensembled voting classifier.

Keywords

Natural Language Processing, Sentiment Classification, Machine Learning, Ensemble Learning, Code-mixed

1. Introduction

Natural language processing (NLP) is a domain in which machines can understand natural languages. Humans express their views or emotions which are called sentiments. Sentiment classification is a field of NLP that refers to automatically classifying emotional states and subjective information. This has wide application in the area of social media, online marketing, and service-related businesses. The rise of Internet users and technology resulted in a huge amount of unstructured data over the Web for Indian languages. Multilingual users are inclined to mix multiple languages especially their native language and English while expressing their views, this resulted in the generation of code-mixed content. Therefore, the drift is towards the code-mixed Indian Languages sentiment classification.

Patra et al. [1] discussed the challenges regarding code-mixed sentiment classification. The difficulties arise due to noisy code-mixed data which needs to be cleaned and preprocessed, language identification and part-of-speech tagging becoming preliminary tasks, non-availability of annotated code-mixed sentiment lexicon, the existing dataset not being sufficient to perform unsupervised learning.

The FIRE 2021 shared task [2,3] is about sentiment classification for code mixed Dravidian languages. The combination of language includes Tamil-English (TA-EN), Kannada-English (KA-EN), and Malayalam-English (MA-EN). The proposed methodology implements several machine learning algorithms to achieve better performance for the given task.

FIRE 2021: Forum for Information Retrieval Evaluation, December 13–17, 2021, India
EMAIL: rashmikb@bmsce.ac.in (Rashmi K.B.); guru.ise@bmsce.ac.in (Guruprasad H.S.); shambhavibr.ise@bmsce.ac.in (Shambhavi B.R.)

2. Related Work

Gazi Imtiyaz Ahmad et al. [4] provided an exhaustive review about sentiment classification for Indian Languages with special attention to code mixed content. They listed out the reasons for the hardness in code-mixed sentiment classification as the data contains noise, requires preprocessing, language identification, POS tagging, and lack of labeled dataset. The study demonstrates most of the research was carried for Hindi, Tamil, and Bengali languages and there is a scope for other local languages.

Pravalika A et al. [5] focused on code mixed sentiment classification for Hindi-English language pair from Facebook. The proposed approach presented domain-based lexicon and machine learning methods. The lexicon method achieved better accuracy compared to the machine learning approach. They intend to address domain-independent and other multilingual data. Mohammed Arshad Ansari et al. [6] designed a system for code mixed Romanized Hindi and Marathi text sentiment classification. They implemented and compared K-NN, Naïve Bayes, and SVM which are supervised learning models. They created Marathi Wordnet in Python and stress the importance of SentiWordnet. T.Y.S.S. Santosh et al. [7] identified hate language in social media Hindi-English bilingual data. They implemented Long Short-Term Memory (LSTM) at a sub-word level and Hierarchical LSTM for available code-mixed datasets.

S. Padmaja et al. [8] created and annotated a code-mixed Telugu-English dataset by extracting movie-related tweets. Sentiment classification for the created dataset followed machine learning and lexicon-based methods. A lexicon-based method included language identification and back transliteration. The machine learning approach included SVM n-gram features.

Bharathi Raja Chakravarthi et al. [9,10] has given a summary of the FIRE 2020 shared task for sentiment classification for Dravidian code-mixed languages which included Malayalam and Tamil mixed with English. They discussed the challenges related to class imbalance; less accuracy related to sentiment classification on rich resource languages without code-switching. Yashvardhan et al. [11] implemented language-specific preprocessing, sub-word level representation, an LSTM network for the Dravidian code-mixed datasets. Fazlourrahman Balouchzahi et al. [12] proposed a Hybrid Voting Classifier which combined deep learning and machine learning classifiers. The machine learning approach included n-gram features and word embeddings. The deep learning approach included sub-word embedding features for BiLSTM. One of the methods in ensemble learning is a voting classifier based on the principle of majority voting by baseline algorithms for final prediction.

Thomas Mandl et al. [13] have summarized the FIRE 2020 Hate Speech and Offensive Content Identification (HASOC) track which included identifying the offensive language and detecting the hate speech for German, English, Hindi, Malayalam, and Tamil languages. Bharathi Raja Chakravarthi et al. [14] has given an overview of the shared track of identifying the offensive language for Tamil and Malayalam code mixed languages. They collected data from YouTube comments, tweets, and Helo App comments. Bharathi Raja Chakravarthi et al. [15] created a shared task and dataset for detecting the offensive languages for code-mixed Dravidian languages.

Sajeetha Thavareesan et al. [16] proposed Part of Speech tagger for Tamil data. They collected the data from various social media platforms and movie websites. Shardul Suryawanshi et al. [17] provided the Tamil memes dataset and created a shared task to identify whether the meme is a troll or not. Sajeetha Thavareesan et al. [18] expanded the sentiment lexicon for Tamil languages by using word embedding approaches for further sentiment classification. Bharathi Raja Chakravarthi et al. [19] presented a summary of the shared task for machine translation for English to Tamil, Tamil to Telugu, English to Malayalam, and English to Telugu language pairs. They collected the dataset from the 2018 released Open subtitles repository. Sajeetha Thavareesan et al. [20] performed sentiment classification for Tamil data by various machine learning approaches and feature representations. They concluded ensemble classifiers may give better accuracy.

Bharathi Raja Chakravarthi et al. [21,22] have discussed identifying the hope speech for Malayalam and Tamil and English languages. They collected the data from YouTube comments and annotated the data manually.

3. Task Description

The shared task Dravidian-Codemix - FIRE 2021 provided the datasets for code-mixed sentiment classification for Dravidian languages. It included the YouTube comments from Kannada-English [23], Tamil-English [24], and Malayalam-English [25] language pairs. The aim is to detect the polarity of the sentiment at the message level. The dataset included three types of code-mixing – "Tag", "Inter-Sentential", and "Intra-Sentential". The polarity labels are "Positive", "Neutral", "Mixed feeling", "Negative", and "not in the intended languages". The dataset included comments in native script as well as Latin script. The detailed description is depicted in Table 1. Example Sentences from Training dataset of Kannada-English dataset is shown in Table 2.

Table 1
Data Statistics

Languages	Training data	Development data	Testing data
Tamil-English	35,657	3,963	4,403
Kannada-English	6213	692	768
Malayalam-English	15,889	1,767	1,963

Table 2
Examples of Kannada-English code- mixed comments

Comments	Polarity
ಇದು ಇದು actually ಚೆನ್ನಾಗಿರೋದು.😍😍😍	Positive
Telugu lyrics super	Not-Kannada
@Vinay Vn kannadalli helu bro gothagalla english idu remake a?	Negative
ಚಿತ್ರ ಗೆದ್ದಿದೆ ಆದರೆ nಪ್ರೇಕ್ಷಕ ಸೋತಿದ್ದಾನೆ	Mixed feelings
ಕೆಲಸ ಬೇಕಾnCall to 8546903696	Unknown state

4. Methodology

The proposed methodology to perform code mixed sentiment classification task is as shown in Figure 1. The phases include cleaning and normalizing the given comments, splitting them into words, extracting the features, training, and predicting with the baseline models further choosing the voting classifier as the final model.

4.1. Preprocessing

The comments in the dataset contain emojis which are important for sentiment analysis therefore those are replaced with related sentiment English text. The dataset included most sentences in Roman script and few sentences in the native script, to get uniformity those are transliterated to Roman script. Preprocessing includes removal of special characters, numbers and converting sentences into lower case. Furthermore, comments are tokenized and feature vectors are extracted by a term frequency-inverse document frequency (TF-IDF) measure.

4.2. Baseline Classifiers

In this section, various machine learning classifiers are explained which are implemented for the task as baseline classifiers for further learning. The classifiers are Logistic Regression, Random Forest, Balanced Random Forest, XGBoost and, SVM.

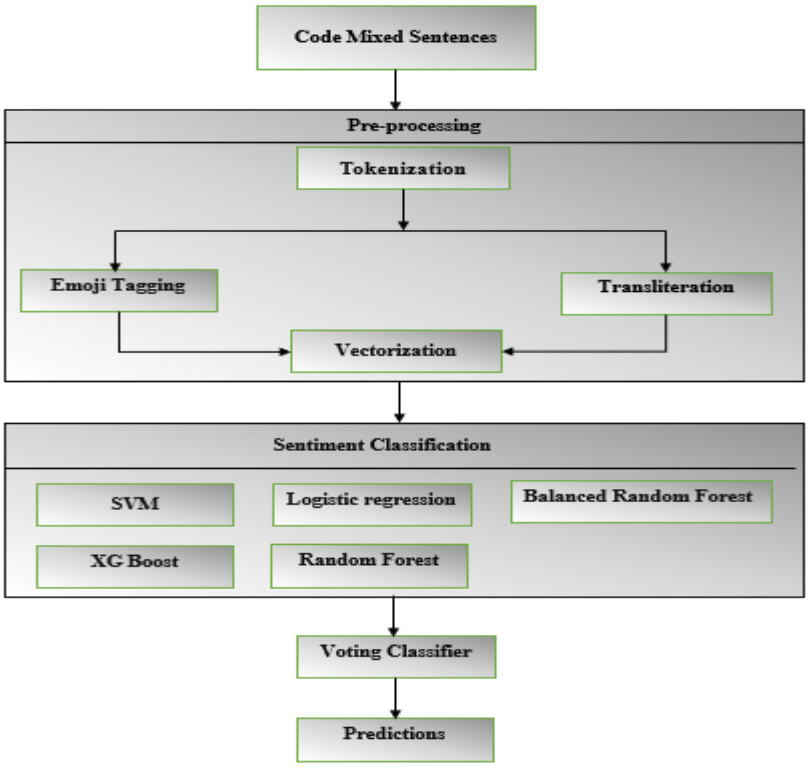

Figure 1: System Architecture

4.2.1. SVM

SVM are supervised learning models more suitable for classification and regression problems [26]. Data elements are plotted as points in n-dimensional space. The dimensions are the features count in the dataset. Further classification can be accomplished by discovering the hyperplane which discriminates the classes. It supports a binary classification. Multiclass classification is achieved by dividing the problem into subproblems and applying the basic principle.

4.2.2. Logistic Regression

It is a supervised learning model for classification which finds the probability of classes. The types of logistic regression are binomial, multinomial, and ordinal. The multinomial logistic regression classifier is suitable for the problem at hand as the dataset contains unordered classes.

4.2.3. Balanced Random Forest

These are the type of Random Forest specifically to handle class imbalance problems. This works on the principle of using a random under-sampling strategy on the majority class within a bootstrap sample to balance the two classes.

4.2.4. XGBoost

XGBoost is eXtreme Gradient Boosting which are ensemble learning models based on Gradient Boosted decision trees used for classification, regression, and prediction problems. It is extreme as it is a faster and accurate version of Gradient Boosting. Decision trees are created in linear patterns. Each class assigned with the weights those are fed into the decision tree for prediction. The weights of wrong

predictions are increased and fed to the second decision tree. These are ensembled to provide more precise results.

4.2.5. Random Forest

It is a supervised ensemble learning method where independent decision trees are built for each training sample and prediction for the test sample is based on the classes selected by maximum decision trees. The selection of the subsample from the training set is random hence avoiding overfitting.

4.3. Ensemble Learning

It is a predictive technique that improves the overall performance by combining the results from multiple classifiers [27]. One such method is Maximum voting which is more suitable for classification. Maximum voting can be soft and hard. The soft voting classifier uses the predicted probabilities of the labels whereas the hard voting classifier uses the class labels from the baseline algorithms.

5. Implementation

The indic-transliteration[1] tool is used to transliterate text from native script to Roman script. The textblob[2] is used for tokenization. Features are extracted and models are trained by using the scikit-learn[3] python module. TF-IDF feature vectors are obtained from the text data by using TfidfVectorizer from the scikit-learn feature extraction model. Baseline and ensemble methods from scikit-learn are used for the task. All baseline classifiers mentioned in the above section are trained by using the given training dataset for each language. The parameters for these machine learning models are as shown in Table 3. The accuracy is calculated for each model using the test dataset. SVM and Logistic Regression performs better compared to other classifiers. Balanced Random Forest overall accuracy is less but it improves the prediction for minority classes. Further to improve the performance Voting Classifier is used. Therefore, an ensemble soft voting classifier is used for the validation and test dataset. The code is given in the Github repository[4].

Table 3
Parameters for machine learning models

Model	Parameters
SVC	kernel='linear',random_state=0, probability=True, tol=1e-6
Logistic Regression	random_state=0,max_iter=500, solver = 'lbfgs', multi_class = 'multinomial'
Balanced Random Forest	n_estimators=1000,max_depth=30,n_jobs=3, random_state=0
XGB	n_estimators=1000,max_depth=3,use_label_encoder=False,eval_metric='mlogloss'
Random Forest	n_estimators=1000, random_state=0, max_depth=10

[1] https://pypi.org/project/indic-transliteration/
[2] https://pypi.org/project/textblob/
[3] https://scikit-learn.org/
[4] https://github.com/Rashmi-KB/FIRE2021.git

6. Result Analysis

To analyze the achieved results, the classification report tool is used from the scikit learn metrics module[1]. The performance of all the models is measured by a weighted-F1 score and accuracy. The classification report for MA-EN, KA-EN, and TA-EN with Baseline classifiers is as shown in Table 4. SVM and Logistic Regression results are higher and very nearer, the least being the Balanced random forest. The F1 score and accuracy calculations are as follows

$$\text{Precision} = \frac{\text{TP}}{\text{TP+ FP}} \quad (1)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+ FN}} \quad (2)$$

$$F1 = 2 * \frac{precision*recall}{precision+recall} \quad (3)$$

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \quad (4)$$

Table 4
Classification report for each language pair (Baseline Classifiers)

Languages	Metric	SVM	Logistic Regression	Balanced Random Forest	XGBoost	Random Forest
MA-EN	Weighted F1-score	0.55	0.55	0.37	0.54	0.34
	Accuracy	0.56	0.56	0.36	0.55	0.45
KA-EN	Weighted F1-score	0.57	0.57	0.30	0.54	0.36
	Accuracy	0.58	0.58	0.30	0.56	0.51
TA-EN	Weighted F1-score	0.57	0.58	0.35	0.57	0.42
	Accuracy	0.63	0.62	0.32	0.62	0.58

Class-wise result analysis for the final ensemble soft voting classifier for test data is as shown in Table 5. Results are improved, compared with the baseline algorithms. The overall results are shown for both validation and test dataset in Table 6. The F1-score and accuracy attained for Malayalam are 0.56 and 0.57. The F1-score and accuracy attained for Kannada are 0.58 and 0.60. The F1-score and accuracy attained for Tamil are 0.56 and 0.63.

Table 5
Class- wise result analysis for each language pair (Soft Voting Classifier)

Languages	Class	Precision	Recall	F1-Score
MA-EN	Mixed_feelings	0.55	0.24	0.33
	Negative	0.61	0.26	0.36
	Positive	0.65	0.57	0.61
	not-malayalam	0.61	0.59	0.60
	Unknown_state	0.51	0.77	0.61
KA-EN	Mixed feelings	0.33	0.09	0.14
	Negative	0.68	0.57	0.62
	Positive	0.63	0.79	0.70
	not-Kannada	0.52	0.53	0.52
	unknown state	0.36	0.24	0.29
TA-EN	Mixed_feelings	0.44	0.04	0.07
	Negative	0.51	0.22	0.31

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

	Positive	0.65	0.95	0.77
	not-Tamil	0.72	0.40	0.52
	Unknown_state	0.55	0.23	0.32

Table 6
Weighted F1-score and Accuracy for validation and test data (Soft Voting Classifier)

Languages	Weighted F1-score		Accuracy	
	Validation	Test	Validation	Test
MA-EN	0.68	0.56	0.69	0.57
KA-EN	0.60	0.58	0.63	0.60
TA-EN	0.58	0.56	0.63	0.63

7. Conclusion

With the increased users in social media and online platforms, sentiment classification for code mixed Indian languages plays a vital role from research, marketing, and customer viewpoint. The paper describes the implementation of various machine learning classifiers for the classification of code-mixed Kannada, Malayalam, and Tamil Tasks provided by the shared task FIRE 2021. The machine learning methods included Logistic Regression, Balanced Random Forest, XGBoost, Random Forest, and SVM as baseline algorithms. The results are improved by the ensemble voting method. A soft voting classifier is used for both validation and test data. For future work, aspect-based sentiment classification could be considered.

References

[1] Patra, Braja Gopal, Dipankar Das, and Amitava Das, "Sentiment Analysis of Code-Mixed Indian Languages: An Overview of SAIL_Code-Mixed Shared Task@ ICON-2017." arXiv preprint arXiv:1803.06745 (2018).

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

[3] Priyadharshini, R., Chakravarthi, B., Thavareesan, S., Chinnappa, D., Durairaj, T., & Sherly, E. (2021). Overview of the DravidianCodeMix 2021 Shared Task on Sentiment Detection in Tamil, Malayalam, and Kannada. In Forum for Information Retrieval Evaluation. Association for Computing Machinery

[4] Ahmad, Gazi Imtiyaz, Jimmy Singla, and Nikita Nikita. "Review on sentiment analysis of indian languages with a special focus on code mixed indian languages." 2019 International Conference on Automation, Computational and Technology Management (ICACTM). IEEE, 2019.

[5] Pravalika, A., Oza, V., Meghana, N. P., & Kamath, S. S. (2017, July). "Domain-specific sentiment analysis approaches for code-mixed social network data." 2017 8th international conference on computing, communication and networking technologies (ICCCNT). IEEE, 2017.

[6] Ansari, Mohammed Arshad, and Sharvari Govilkar. "Sentiment analysis of mixed code for the transliterated hindi and marathi texts." International Journal on Natural Language Computing (IJNLC) Vol 7 (2018).

[7] Santosh, T. Y. S. S., and K. V. S. Aravind. "Hate speech detection in Hindi English code-mixed social media text." Proceedings of the ACM India Joint International Conference on Data Science and Management of Data. 2019.

[8] Padmaja, S., Sasidhar Bandu, and S. Sameen Fatima. "Text Processing of Telugu–English Code-Mixed Languages." International Conference on Emerging Trends in Engineering. Springer, Cham, 2019.

[9] 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: Proceedings of the 12th Forum for Information Retrieval Evaluation, FIRE '20, 2020.
[10] 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 the Forum for Information Retrieval Evaluation (FIRE 2020). CEUR Workshop Proceedings. In: CEUR-WS. org, Hyderabad, India, 2020.
[11] Sharma, Yashvardhan, and Asrita Venkata Mandalam. "Bits2020@ Dravidian-CodeMix-FIRE2020: Sub-Word Level Sentiment Analysis of Dravidian Code Mixed Data." FIRE (Working Notes). 2020.
[12] Balouchzahi, Fazlourrahman, and H. L. Shashirekha. "MUCS@ Dravidian-CodeMix-FIRE2020: SACO-SentimentsAnalysis for CodeMix Text." FIRE (Working Notes). 2020.
[13] Mandl, T., Modha, S., Kumar M, A., & Chakravarthi, B. (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 (pp. 29–32).
[14] Chakravarthi, B., Kumaresan, P., Sakuntharaj, R., Madasamy, A., Thavareesan, S., B, P., Chinnaudayar Navaneethakrishnan, S., McCrae, J., & Mandl, T. (2021). Overview of the HASOC-DravidianCodeMix Shared Task on Offensive Language Detection in Tamil and Malayalam. In Working Notes of FIRE 2021 - Forum for InformationRetrieval Evaluation. CEUR.
[15] Chakravarthi, B.R., Priyadharshini, R., Jose, N., Mandl, T., Kumaresan, P.K., Ponnusamy, R., Hariharan, R.L., McCrae, J.P. and Sherly, E., 2021, April. 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 (pp. 133–145). Association for Computational Linguistics.
[16] Thavareesan, S. and Mahesan, S., 2020, November. Word embedding-based Part of Speech tagging in Tamil texts. In 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS) (pp. 478-482). IEEE.
[17] Suryawanshi, S., & Chakravarthi, B. R. (2021, April). Findings of the Shared Task on Troll Meme Classification in Tamil. In Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages (pp. 126–132). Association for Computational Linguistics.
[18] Thavareesan, S. and Mahesan, S., 2020, July. Sentiment Lexicon Expansion using Word2vec and fastText for Sentiment Prediction in Tamil texts. In 2020 Moratuwa Engineering Research Conference (MERCon) (pp. 272-276). IEEE.
[19] Chakravarthi, B. R., Priyadharshini, R., Banerjee, S., Saldanha, R., McCrae, J. P., Krishnamurthy, P., & Johnson, M. (2021, April). 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 (pp. 119–125). Association for Computational Linguistics.
[20] Thavareesan, S. and Mahesan, S., 2019, December. Sentiment analysis in Tamil texts: A study on machine learning techniques and feature representation. In 2019 14th Conference on Industrial and Information Systems (ICIIS) (pp. 320-325). IEEE.
[21] Chakravarthi, B.R., 2020, December. 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 (pp. 41-53).
[22] Chakravarthi, B.R. and Muralidaran, V., 2021, April. Findings of the shared task on Hope Speech Detection for Equality, Diversity, and Inclusion. In Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion (pp. 61-72).
[23] Hande, Adeep, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. "KanCMD: Kannada CodeMixed Dataset for Sentiment Analysis and Offensive Language Detection." Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media. 2020.
[24] B. R. Chakravarthi, V. Muralidaran, R. Priyadharshini, J. P. McCrae, 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), European Language Resources association, Marseille, France, 2020, pp. 202–210. URL: https://www. aclweb.org/anthology/2020.sltu-1.28.

[25] 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 1st 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://www.aclweb.org/anthology/ 2020.sltu-1.25.
[26] Joachims, Thorsten. "Text categorization with support vector machines: Learning with many relevant features." European conference on machine learning. Springer, Berlin, Heidelberg, 1998.
[27] Saleena, Nabizath. "An ensemble classification system for twitter sentiment analysis." Procedia computer science 132 (2018): 937-946.