

# BiLSTM-Sentiments Analysis in Code-Mixed Dravidian Languages

Mudoor Devadas Anusha, Hosahalli Lakshmaiah Shashirekha

*Department of Computer Science, Mangalore University, Mangalore, Karnataka, India*

## Abstract

Understanding the sentiments of a comment/post on social media is a fundamental move in numerous applications and Sentiments Analysis (SA) of a text can be worthy for decision-making process. Over the past few years, SA of texts have received much attention. One such application is to analyze the main-stream sentiments of videos on social media based on viewer comments. Social media text is primarily code-mixed and research on SA in code-mixed low-resourced languages is in its infancy that too for very few language pairs. Non-availability of annotated code-mixed data for low-resourced languages makes the SA task much more complex. Kannada, Malayalam and Tamil languages belonging to the family of Dravidian languages are popular south Indian languages but are low-resourced. Each of these languages' content mixed with English language either in Roman script or as a combination of native script and Roman script are available on social media abundantly. In this paper, we, team MUM, describe the proposed Bidirectional Long Short Term Memory (BiLSTM) model submitted to "Sentiment Analysis of Dravidian Languages in Code-Mixed Text" - a shared task at Forum for Information Retrieval Evaluation (FIRE) 2021 to analyze the sentiments in Kannada-English (Kn-En), Malayalam-English (Ma-En), and Tamil-English (Ta-En) code-mixed texts. In the proposed approach, the code-mixed word embeddings' are constructed using the training set of the respective code-mixed language pairs' and these embeddings are used to build a Deep Learning (DL) model based on BiLSTM. Our proposed model obtained 13<sup>th</sup>, 14<sup>th</sup>, and 14<sup>th</sup> ranks with weighted F1-scores of 0.563, 0.604, and 0.365 for code-mixed Ta-En, Ma-En and Kn-En language pairs respectively.

## Keywords

Machine Learning, BiLSTM, Word Embedding, Code-mixing, Dravidian language

## 1. Introduction

An analysis of users' sentiments can help in understanding users' attitudes and moods which can help to draw insights for future decision-making. Rather than just a fundamental check of notification or comments, sentiments describe the feelings and assessments. An essential aspect of Natural Language Processing (NLP) is SA which involves understanding the polarity of a given text or sentence. SA is the undertaking of subjective impressions or reactions about a given subject and SA via social media uncovers the opinion of users' about whatever they see or listen on social media [1]. SA is an on-going area of research for more than a decade in both academia and industry. However, the increasing online content in the form of code-mixed text in social media is throwing new challenges to the SA research community.

---


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

✉ [anugowda251@gmail.com](mailto:anugowda251@gmail.com) (M. D. Anusha); [hlsrekha@gmail.com](mailto:hlsrekha@gmail.com) (H. L. Shashirekha)

🌐 <https://mangaloreuniversity.ac.in/dr-h-l-shashirekha> (H. L. Shashirekha)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

Code-mixing or Code-switching is a common event in a multi-lingual community where the sentences, phrases, words and morphemes of two or more languages are mixed in speech or writing according to one's convention. As the younger generation is quite familiar in using English language, they tend to mix up their native language with English according to their whims and fancies. India being a multi-lingual country provides much scope for code-mixed text. Kannada, Malayalam and Tamil languages belong to the family of Dravidian languages and are the official languages of Karnataka, Kerala, and Tamilnadu respectively. The scripts of these three Dravidian languages are alpha-syllabic, relating to a group of the Abugida writing methods that are partly alphabetic and partially syllable-based [2] [3]. Despite their popularity, these three languages are digitally low-resourced languages [4]. As users' tend to use a combination of languages to post comments on social media, the majority of the data available on social media for these languages are code-mixed. Code-mixed texts are usually written in non-native scripts especially in Roman script on social media [5] due to the ease of the use of Roman script and also due to the technological limitations of keyboard layouts of native languages on smart phones.

SA systems trained on mono-lingual text are not suitable for code-mixed text due to the complexity of mixing languages at various linguistic levels in the given text [6]. In order to promote research in SA in code-mixed Dravidian languages, "Sentiment Analysis of Dravidian Languages in Code-Mixed Text"<sup>1</sup> - a shared task in FIRE 2021<sup>2</sup> provides an opportunity for researchers to develop and evaluate the working models for SA. The organizers of the shared task provide the code-mixed datasets in Kn-En, Ma-En, and Ta-En language pairs with an objective of identifying the sentiment polarity of a given code-mixed text in these language pairs.

SA task is a typical binary (coarse grained) or multi-class (fine-grained) Text Classification (TC) task of assigning a sentiment polarity to a given text depending on the predefined number of categories. Researchers have developed several models for SA of natural language mono-lingual texts as well as code-mixed texts using conventional Machine Learning (ML) and DL approaches based on Neural Network (NN). DL models are gaining popularity as they provide accurate and effective results for TC by reducing false positives [7]. Majority of DL models use BiLSTM which contains two Long Short Term Memory (LSTM) models: one taking the commitment in a forward direction and the other in a retrogressive way. BiLSTMs are at the core of many NNs that achieve cutting edge performance in NLP tasks [7].

Embedding words in a document is an active research area in which scientists endeavour to find better representations of words is achieved by capturing the contextual, semantic, and syntactic information about words as much as possible [8]. In this approach, the representation of words is based on the notion of distributional hypothesis in which words with similar meanings occur in similar contexts or textual vicinity and each word is represented by a real-valued vector in a predefined vector space. This distributed representation of words is expected to provide a great deal of insight for many NLP applications as it captures the syntactic and semantic information of words in a sufficiently large corpus.

The BiLSTM NN consists of LSTM units that integrate past and future context information

---

<sup>1</sup><https://dravidian-codemix.github.io/2021/index.html>

<sup>2</sup>[https://competitions.codalab.org/competitions/306424#learn\\_the\\_details-overview](https://competitions.codalab.org/competitions/306424#learn_the_details-overview)

because of which they are showing excellent performance for sequential modeling problems as well as for TC [9]. NN models expect numeric values as input. Hence, it is necessary to convert the text data to numeric representation by building an embedding layer before building a BiLSTM model. In this paper, we, team MUM, describe the BiLSTM model submitted to “Sentiment Analysis of Dravidian Languages in Code-Mixed Text” shared task in FIRE 2021 to identify the sentiment polarity of the given code-mixed comment.

The rest of the paper is organized as follows: Few latest works related to SA are described in Section 2 followed by the proposed methodology in section 3. Experimental setup and Results are described in Section 4 and the paper reaches its conclusion throwing light on future work in Section 5.

## 2. Related Work

Researchers have explored different algorithms for SA of monolingual texts as well as code-mixed texts of different language pairs. However, very few works are reported for code-mixing of Indian language texts in general and Dravidian languages in particular.

Chakravarthi et al. [10] created code-mixed benchmarked corpora for SA in Ma-En language pair. Using youtube-comment-scraper tool<sup>3</sup> they collected 116,711 Ma-En code-mixed sentences from YouTube comments posted for Malayalam movie trailers during 2019. The majority of the contents in these comments were written either in English or as a combination of English and Malayalam in Malayalam script and/or Roman script. These comments were tokenized into sentences and the sentences were annotated for SA by volunteers. In addition, the authors also experimented SA using the traditional ML algorithms, namely: Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Multinomial Naïve Bayes (MNB), and K-Nearest Neighbour (kNN), as baselines. The ML algorithms were trained using the Term Frequency-Inverse Document Frequency (TF-IDF) vectors obtained by vectorizing the sentences. Among the baseline classifiers, LR and RF classifiers achieved higher average weighted macro F1-score of 0.66 and 0.61 respectively.

Hande et. al [11] developed KanCMD - a multi-task Ka-En code-mixed dataset for SA and Offensive Language Identification (OLI). This work aims to promote multi-task learning for under-resourced languages in general and Kannada language in particular. The dataset which consists of 7,671 comments and annotated by at least three annotators is benchmarked using computational models. Similar baselines and feature set as used by Chakravarthi et al. [10] are used to evaluate both SA and OLI on KanCMD dataset. Among all the classifiers, the LR classifier obtained the highest weighted F1-score of 0.70 and 0.77 for SA and OLI respectively.

Chakravarthi et al. [12] proposed the most substantial corpus for code-mixed Ta-En (Tanglish) text with sentiment polarity annotations. They reported a high inter-annotator agreement in terms of Krippendorff  $\alpha$  from voluntary annotators on contributions collected using Google form and created gold standard annotated data for code-mixed Ta-En SA. In addition, the authors also evaluated ML classifiers, namely: LR, kNN (with 3, 4, 5, and 9 neighbors), DT, MNB, RF and SVM and DL classifiers, namely: 1D Conv-LSTM, Bidirectional Encoder Representations from Transformers (BERT)-Multilingual, Dynamic Meta Embedding (DME) and Contextual DME, as

---

<sup>3</sup><https://github.com/philbot9/youtube-comment-scraper>

baselines. They trained the ML classifiers on TF-IDF vectors of word n-grams in the range (1, 3) and used word2vec as feature for DL models. Among all the baselines, RF model exhibited the highest performance with a macro f1-score of 0.42 and the weighted f1-score of 0.65.

A total of 1,200 Hindi and 300 Marathi text consisting of chats, Tweets, and YouTube comments were collected by Ansari et al. [13] for SA of code-mixed transliterated Hindi and Marathi Texts. The study involves Language Identification (LI), word transliteration, sentiment scoring, and feature extraction along with using learning methods. They conducted several experiments to classify transliterated Hindi and Marathi text using kNN, NB, SVM and ontology-based classifiers and obtained a weighted F1-score of 0.59 and 0.57 for NB and Linear SVM respectively.

Choudhary et al. [14] proposed a novel approach called Sentiment Analysis of Code-Mixed Text (SACMT) to classify sentences into positive, negative or neutral sentiments using Contrastive learning. This work introduces a basic clustering-based pre-processing method for capturing variations of code-mixed transliterated words and utilizing the shared parameters of Siamese networks to map the sentences of code-mixed and standard languages to a common sentiment space. The proposed approach employs twin BiLSTM networks with shared parameters to capture a sentiment based representation of the sentences which is used in conjunction with a similarity metric to group sentences with similar sentiments together. SACMT's performance for SA of code-mixed text obtained a weighted F1-score of 0.759 and outperformed the existing approaches by 7.6% in accuracy and 10.1% in F1-score.

The system developed by Joshi et al. [15] introduces sub-word-LSTM architecture for learning sub-word level representations instead of character or word level representations for SA and the linguistic prior in their architecture gives them the ability to learn sentiment information about important morphemes. Also, the authors hypothesize that encoding the linguistic prior in the subword-LSTM architecture leads to superior performance. For the dataset containing 3,879 code-mixed English-Hindi (Hi-En) sentences gathered from Facebook, subword-LSTM and char-LSTM obtained F1-scores of 0.658 and 0.511 respectively. Additionally, the model which performed well for heavily noised text containing misspellings was demonstrated in the morpheme-level feature maps.

Vaibhav et al. [16] proposed a hybrid model for SA tasks in Hi-En code-mixed texts using sub-words embedding. They first generate sub-word level representations for the sentences using a Convolutional Neural Network (CNN) architecture and used them as inputs to a Dual Encoder Network consisting of two different BiLSTMs: Collective and Specific Encoder. The Collective Encoder captures the overall sentiment of the sentence, while the Specific Encoder utilizes an attention mechanism in order to focus on individual sentiment-bearing sub-words. This was combined with a feature network consisting of orthographic features and specially trained word embeddings'. On the dataset consisting of 3,879 code-mixed En-Hi messages created by [15], their proposed model achieved state-of-the-art results of 83.54% accuracy and 0.827 F1-score.

### **3. Methodology**

The proposed methodology includes i) Pre-processing - to clean the text data by removing unnecessary data ii) Feature Engineering - to represent the sentences/comments as word vectors

and iii) Model construction - to perform SA. Each of the steps are explained below:

### 3.1. Pre-processing

Text data needs to be pre-processed to remove noise so that the performance of the classifier can be improved. Pre-processing the data is just as important as the model building itself. Text pre-processing procedures may differ depending on the task and the dataset used. The following pre-processing steps were applied in the proposed work:

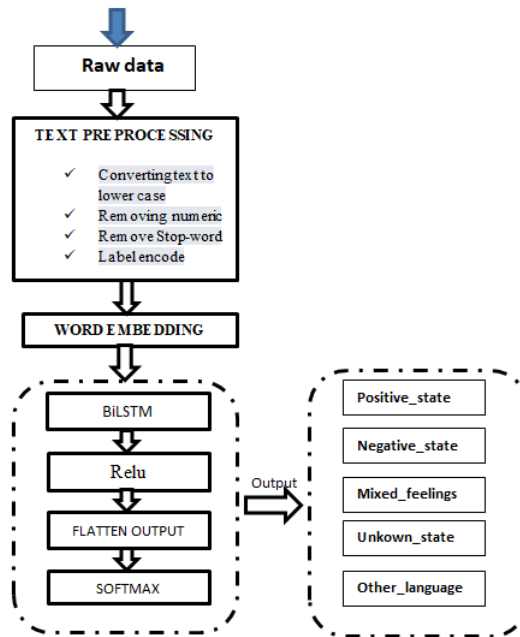
- Converting text to lowercase as the character case does not matter for TC task.
- Removing numeric and punctuation information as they are not important for TC task.
- Eliminating stop words - the frequently occurring words in a language, as they are not the distinguishing features for TC task.
- Label encoding refers to converting the class/category labels into numeric form to make them machine-readable. For example, the class labels of SA task: Mixed feelings, Positive, Unknown\_state, Negative, and Not-Kannada, will be encoded as 0, 1, 2, 3 and 4 respectively.

### 3.2. Feature Engineering

As text data has to be represented as numeric values for any NN model, word embeddings' are used to encode text data into numeric vectors. For learning word embeddings, Thomas Mikolov's [17] word2vec skip-gram model with 300 embedding dimensions and window size 10 is used. The advantage of word2vec is that its vector can be used to learn words' similarities and relationships [18]. Word2vec skip-gram model is first trained with the Train set provided by the organizers of the shared task to obtain the word embeddings'. After training, the word embedding is considered as a lookup table and the representation of the words is obtained from this look-up table. The sentences/comments are represented as vectors by averaging the numeric representations of all the words present in that sentence/comment.

### 3.3. Model Construction

BiLSTM - a Recurrent Neural Network (RNN) which works with two hidden layers taking data in both the directions simultaneously has shown good results for NLP applications [19]. The generated embedding vectors of 300 dimension with activation, optimizer and dropout parameters set to "softmax", "adam" and 20% respectively are used to train a BiLSTM network for dynamic epochs until the loss value gets stabilized (at most 20 times). Input to the BiLSTM layer is fed through the time distributed wrapper to a dense layer with the activation function of a Rectified Linear Unit (ReLU), followed by a dense layer with softmax activation after flattening the output from the previous layer. Output dimensions of the model are configured based on the number of class labels. The structure of the BiLSTM model is shown in Figure 1.



**Figure 1:** Structure of the BiLSTM Model

## 4. Experimental setup and Results

The dataset<sup>5</sup> [20] provided by the shared task organizers for SA of code-mixed Ka-En [11], Ma-En [21] and Ta-En [22] [23] language pairs consists of Train, Development (Dev), and Test sets. A comment/post in the dataset contains more than one sentence, but the average number of sentences in each language pair is one and each comment/post is annotated with one of the sentiment polarities: Positive\_state, Negative\_state, Mixed\_feelings, Neutral, Unkown\_state, and Other\_language (Not Kannada/Tamil/Malayalam). The given dataset has the class imbalance issue which is consistent with how sentiments are expressed in reality and the distribution of labels in the dataset are shown in Table 1.

Scikit-learn<sup>6</sup> and keras<sup>7</sup> - a minimalist library for DL, are used to implement the code in Python. BiLSTM model with word embedding feature applied for the Test set of all three language pairs obtained 13<sup>th</sup>, 14<sup>th</sup>, and 14<sup>th</sup> ranks with weighted F1-scores of 0.563, 0.604, and 0.365 for Ta-En<sup>8</sup>, Ma-En<sup>9</sup>, and Ka-En<sup>10</sup> respectively. The results obtained in terms of Precision, Recall, and F1-score are shown in Table 2. Table 3 shows the performance of the proposed approach on the Development sets of Ta-En, Ma-En, and Ka-En language pairs.

<sup>5</sup><https://competitions.codalab.org/competitions/30642#participate>

<sup>6</sup><https://scikit-learn.org/stable>

<sup>7</sup>[https://www.tensorflow.org/api\\_docs/python/tf/keras/](https://www.tensorflow.org/api_docs/python/tf/keras/)

<sup>8</sup>[https://drive.google.com/file/d/14pKDC5fuRcWoAnn\\_HpBD50pdxGsxxvPT/view](https://drive.google.com/file/d/14pKDC5fuRcWoAnn_HpBD50pdxGsxxvPT/view)

<sup>9</sup><https://drive.google.com/file/d/1nZaQ4fm0h6rIHVtbYwWYVmvM8AFD71pD/view>

<sup>10</sup><https://drive.google.com/file/d/1TkWH9vp89p2Yzza3OS3XfWhWYwudhAdA/view>

**Table 1**

Distribution of labels in the given dataset

Language Pair \Label		Positive_ state	Mixed_ feelings	Unknown_ state	Other_ language	Negative_ state	Total
Malayalam -English	Train	6,421	926	5,279	1,157	2,105	15,888
	Dev	706	102	580	141	237	1,768
	Test	780	134	643	147	258	1,962
Tamil -English	Train	20,070	4,020	5,628	1,667	4,271	35,656
	Dev	2,257	438	611	176	480	3,962
	Test	2,546	470	665	244	477	4,402
Kannada -English	Train	2,823	574	711	916	1,188	6212
	Dev	321	52	69	110	139	691
	Test	374	65	62	110	157	768

**Table 2**

Results of the proposed BiLSTM models on the Test sets

Language pair	Precision	Recall	F1-score	Rank
Malayalam-English	0.583	0.624	0.563	13
Tamil-English	0.621	0.626	0.604	14
Kannada-English	0.407	0.487	0.369	14

**Table 3**

Results of the proposed BiLSTM models on the Development sets

Language pairs	Precision	Recall	F1-score
Malayalam-English	0.662	0.648	0.684
Tamil-English	0.684	0.651	0.714
Kannada-English	0.591	0.546	0.601

## 5. Conclusion and Future Work

In this work, we, team MUM, present the description of the working model for the SA of code-mixed text in Kannada, Malayalam, and Tamil submitted to “Sentiment Analysis of Dravidian Languages in Code-Mixed Text” shared task in FIRE 2021. To tackle the challenge of classifying the given YouTube comments into one of the six predefined categories, we propose a BiLSTM model with word embeddings’ as features. The proposed model obtained F1-scores of 0.563, 0.604, and 0.365 for Ta-En, Ma-En, and Ka-En language pairs respectively. Exploring different features and different learning models such as Transfer Learning for SA of code-mixed Indian languages are in the pipeline.

## References

- [1] B. R. Chakravarthi, R. Priyadharshini, S. Thavareesan, D. Chinnappa, T. Durairaj, 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 2021, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.
- [2] B. R. Chakravarthi, N. Rajasekaran, M. Arcan, K. McGuinness, N. E. O'Connor, J. P. McCrae, Bilingual Lexicon Induction across Orthographically-Distinct Under-Resourced Dravidian Languages (2020).
  - [3] B. Krishnamurti, *The Dravidian Languages*, Cambridge University Press, 2003.
  - [4] B. R. Chakravarthi, R. Priyadharshini, S. Banerjee, R. Saldanha, J. P. McCrae, P. Krishnamurthy, M. Johnson, et al., 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, 2021, pp. 119–125.
  - [5] R. Priyadharshini, B. R. Chakravarthi, M. Vegupatti, J. P. McCrae, Named Entity Recognition for Code-Mixed Indian Corpus using Meta Embedding, in: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), IEEE, 2020, pp. 68–72.
  - [6] K. Bali, J. Sharma, M. Choudhury, Y. Vyas, "I am borrowing ya mixing?" An Analysis of English-Hindi Code Mixing in Facebook, in: Proceedings of the First Workshop on Computational Approaches to Code Switching, 2014, pp. 116–126.
  - [7] R. Bhargava, Y. Sharma, S. Sharma, Sentiment Analysis for Mixed Script Indic Sentences, in: 2016 International Conference On Advances In Computing, Communications And Informatics (ICACCI), IEEE, 2016, pp. 524–529.
  - [8] K. Ghosh, A. Senapati, Technical Domain Identification using Word2vec and BiLSTM, in: 17th International Conference on Natural Language Processing, 2020, p. 21.
  - [9] B. Jang, M. Kim, G. Harerimana, S.-u. Kang, J. W. Kim, Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism, *Applied Sciences* 10 (2020) 5841.
  - [10] B. R. Chakravarthi, N. Jose, S. Suryawanshi, E. Sherly, J. P. McCrae, A Sentiment Analysis Dataset for Code-Mixed Malayalam-English, arXiv preprint arXiv:2006.00210 (2020).
  - [11] A. Hande, R. Priyadharshini, B. R. Chakravarthi, KanCMD: Kannada CodeMixed Dataset for Sentiment Analysis and Offensive Language Detection, in: Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media, 2020, pp. 54–63.
  - [12] B. R. Chakravarthi, V. Muralidaran, R. Priyadharshini, J. P. McCrae, Corpus Creation for Sentiment Analysis in Code-Mixed Tamil-English Text, arXiv preprint arXiv:2006.00206 (2020).
  - [13] M. A. Ansari, S. Govilkar, Sentiment Analysis of Mixed Code for the Transliterated Hindi And Marathi Texts, *International Journal on Natural Language Computing (IJNLC)* Vol 7 (2018).
  - [14] N. Choudhary, R. Singh, I. Bindlish, M. Shrivastava, Sentiment Analysis of Code-Mixed Languages Leveraging resource rich Languages, arXiv preprint arXiv:1804.00806 (2018).
  - [15] A. Joshi, A. Prabhu, M. Shrivastava, V. Varma, Towards Sub-Word Level Compositions for Sentiment Analysis of Hindi-English Code Mixed Text, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 2482–2491.
  - [16] Y. K. Lal, V. Kumar, M. Dhar, M. Shrivastava, P. Koehn, De-Mixing Sentiment from



- Code-Mixed Text, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, 2019, pp. 371–377.
- [17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed Representations of Words and Phrases And their Compositionality, in: Advances In Neural Information Processing Systems, 2013, pp. 3111–3119.
- [18] A. Khatua, A. Khatua, E. Cambria, A Tale of Two Epidemics: Contextual Word2Vec for Classifying Twitter Streams during Outbreaks, *Information Processing & Management* 56 (2019) 247–257.
- [19] K. S. Tai, R. Socher, C. D. Manning, Improved Semantic Representations from Tree-Structured Long Short-Term Memory Networks, arXiv preprint arXiv:1503.00075 (2015).
- [20] B. R. Chakravarthi, R. Priyadharshini, N. Jose, A. Kumar M, T. Mandl, P. K. Kumaresan, R. Ponnusamy, H. R L, J. P. McCrae, E. Sherly, Findings of The Shared Task on Offensive Language Identification in Tamil, Malayalam, and Kannada, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, 2021.
- [21] 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), 2020.
- [22] R. Priyadharshini, B. R. Chakravarthi, S. Thavareesan, D. Chinnappa, T. Durairaj, E. Sherly, Overview of the DravidianCodeMix 2021 Shared Task on Sentiment Detection in Tamil, Malayalam, and Kannada, in: Forum for Information Retrieval Evaluation, FIRE 2021, Association for Computing Machinery, 2021.
- [23] 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), 2020.