Sentiment Analysis of YouTube comments in Dravidian Code-Mixed Language using Deep Neural Network

N Muhammad Fadil[a], Lavanya S K[b]

[a]*Department of Computational Intelligence,* [b]*Faculty of Computational Intelligence*
SRM Institute of Science and Technology, Tamil Nadu, India

Abstract

Sentiment analysis is a method for determining the positivity, negativity, or neutrality of a text block. The purpose of sentiment analysis is to study public sentiment in a manner that promotes company growth. This study seeks to classify the feelings of a dataset of comments/posts into pre-defined classifications for the code-mixed languages Tamil, Malayalam, and Kannada. The Sequential Deep Learning model is used to the code-mixed dataset to identify sentiments. The experiment was carried out using the dataset from the Codalab 2022 competition "Shared Task on Sentiment Analysis and Homophobia Detection of YouTube Comments in Code-Mixed Dravidian Languages", which included social media comments in Tamil, Malayalam, and Kannada code-mixed languages.

Keywords

Sentiment, Code-Mixed, YouTube, LSTM, Keras

1. Introduction

The objective of the challenge is to classify attitudes from YouTube comments in a dataset containing many languages. Therefore, this challenge is a classification problem with multiple classes. Multi-class classification involves the classification of more than two classes. Numerous successful studies involving the use of sentiment analysis to monolingual texts have previously been done. However, many fewer experiments have been undertaken regarding the sentiment analysis of code-mixed languages.

Our study's purpose is to categorize YouTube comments into the following categories: positive, negative, neutral, mixed, and code-mixed language if the word is not in the relevant language. The Sequential model has been trained for this specific purpose. From the development datasets, the accuracy, precision, recall, and F1 score are then measured.

2. Related Works

For the purpose of protecting social media users from cyberbullying, social media companies have always been required to fund/contribute to sentiment analysis research. There have been a number of studies examining models for sentiment analysis. Different fields utilize different methodologies and models. However, few academic studies have examined the use of Emoji characters on social media [1]. When used out of context, emojis can drastically alter a message. To successfully classify the comments in this experiment, we remove all emoji from the dataset provided. Sentiment analysis at the word level studies the orientation of individual words and phrases and how it influences the overall tone, whereas sentiment analysis at the sentence level analyses sentences that reflect a single perspective and seeks to discern its orientation. A lexicon-based method relies on a corpus or list of words with a particular polarity. Then, an algorithm searches for these words, counts or estimates their weight, and measures the text's overall polarity [2, 3, 4].

Recent works on sentiment analysis of mixed-code formats have expanded in number [5, 6, 7, 8, 9, 10]. In a nation where multiple languages are spoken, code-mixing becomes widespread. People in multilingual nations use code-mixed discourse when communicating online and in person [11]. Sequence models are machine learning models that accept or produce data sequences as input or output. Sequential data consists of text streams, audio and video fragments, time-series data, and other types. Recurrent neural networks (RNNs) are commonly employed in sequence modeling [12]. The study of discrete sequential data, including time series, text phrases, and other sequential data, inspired the development of Sequence Models. These models are better suited to manage sequential data, whereas Convolutional Neural Networks are better suited to manage spatial data.

Dravidian languages have been code-mixed with English in the current study, such as "Tamil-English", "Malayalam-English", and "Kannada-English" [13, 14, 15]. This dataset is part of Task A of "Sentiment Analysis and Homophobia detection in YouTube comments." This study classifies each YouTube comment into one of the following message-level categories: "Positive," "negative," "not-tamil/malayalam/kannada," "unknown state," and "mixed-feelings." The experimental results on the Sequential model for the supplied dataset revealed an accuracy of 0.53 for "Malayalam-English".

3. Data

The dataset utilized in this study is provided by Task A of "Sentiment Analysis and Homophobia detection in YouTube comments." It includes YouTube comments written in Tamil, Malayalam, and Kannada (data for all 3 from [8, 10, 11]). The training dataset for Tamil includes 35656 instances, the validation/development dataset includes 3962 instances, and the test dataset includes 649 instances. "Positive," "Negative," "unknown state," "Mixed feelings," and "not-Tamil" are the classes. There are 15888 instances in the Malayalam training dataset, 1766 instances in the validation/development dataset, and 1962 instances in the test dataset. "Positive," "Negative," "unknown state," "Mixed feelings," and "not-malayalam" are the classes. There are 6212 instances in the Kannada training dataset, 691 instances in the validation/development dataset, and 768 instances in the test dataset. "Positive", "Negative", "unknown state", "Mixed feelings", and "not-Kannada" are the classes.

4. Methodology

In this paper, a multi-task classification model for sentiment analysis of YouTube comments written in mixed-Dravidian code is developed. Each comment in the dataset must be represented by a numerical feature vector for a supervised classifier to be trained.

4.1 Task

The objective is to classify each YouTube remark into one of five classes: "Positive", "Negative", "unknown state", "Mixed feelings", and "not-Tamil/Malayalam/Kannada".

4.2 Data Preprocessing

Given that the YouTube dataset is code-mixed and defies grammatical standards. To successfully utilize the dataset, the following procedures are implemented.

- The texts are initially transformed to lowercase and stemming and lemmatization are performed.
- In the following phase, all emojis, special characters, numbers, and punctuation must be removed because they serve no use in a statement.
- Sentences of two letters or less were deleted since they had minimal impact on the data set.
- Next, the training dataset was compiled. After cleaning the text, it was tokenized and encoded into a collection of token indexes.
- Finally, padding was used to verify that all texts were of equal length.

4.3 Model

For sentiment analysis tasks, a DNN has been built. The input for these networks came from the embedding vectors. Initially, word embedding was included to the model. The embedding initializer, embedding regularizer, and embedding regularizer were all assigned as "maximum length," "orthogonal," and "L2 Regularizer" After adding the LSTM layer, we wrapped it with Bidirectional. Bidirectionality was added to a Keras layer by implementing tf.keras.layers.bidirectional within the model. We classified the data into five classes using the Dense layer and the'softmax' activation function..

Our model was compiled and the loss function, optimizer, and metrics were defined. We select "Categorical Cross-Entropy" as the loss function since the provided problem involves multi-class categorization. We applied the default optimizer Adam and a learning rate of 0.01 to the provided problem. We had previously used 'accuracy,' 'precision,"recall,' and 'auc' as measures..

Consequently, we must now train our model to fine-tune the parameters in order to provide the required outputs for a given input. This is achieved by feeding inputs into the input layer, receiving an output, calculating the loss function using the output, and then fine-tuning the model parameters via backpropagation. Consequently, the parameters of the model will be fitted and matched to the data. While fitting the model, the batch size was 256 and the number of epochs was 2..

5. Implementation

The notebook file imports all necessary modules and packages, such as TensorFlow, pandas, NumPy, Regular Expression, Natural Language Toolkit, scikit-learn, etc. Python's scikit-learn1 library is utilized for feature extraction and model training. Using scikit-Tfidf learn's Vectorizer, the text input is turned into TF-IDF feature vectors. The Tamil, Malayalam, and Kannada training sets are utilized to train the sequential model. The accuracy of the three languages is calculated. The development set is utilized to determine the accuracy of the model. The following table shows the accuracy, precision, recall, and f1-score for all three languages.

Language	Accuracy	Precision	Recall	F1-Score
Tamil	0.470	0.410	0.520	0.420
Malayalam	0.550	0.520	0.590	0.520
Kannada	0.570	0.470	0.510	0.480

6. Results and Conclusion

The table presents the weighted "macro" averages for each statistic across all three languages. For the term 'Tamil,' a precision of 0.40, an accuracy of 0.47, a recall of 0.515, and a f1-score of 0.41 were observed. 'Malayalam' was given a precision of 0.53, an accuracy of 0.55, a recall of 0.62, and a f1-score of 0.53. 'Kannada' was given a precision of 0.50, an accuracy of 0.57, a recall of 0.53, and a f1-score of 0.51. The same model was utilized for all three languages. Comparatively, the 'Malayalam' dataset had the highest precision, recall, and f1-score, although the 'Kannada' dataset had slightly greater accuracy than the 'Malayalam' dataset. Compared to the other two datasets, the "Tamil" dataset performed poorly with the model. However, it should be noted that the 'Positive' class in the 'Tamil' dataset has a disproportionately large number of instances relative to the other classes. The disparity in the data reduced the precision. The extensive training and development data relative to the other two languages may have also contributed to the low performance on the measures.

As a result, we tested the three languages in this research using the DNN Sequential model. This method can be applied to any language because it is language-independent.

References

[1] Preisendorfer, Matthew. (2018). Social Media Emoji Analysis, Correlations and Trust Modeling. 10.13140/RG.2.2.25466.18888.

[2] Xiang, Rong & Chersoni, Emmanuele & Lu, Qin & Huang, Chu-Ren & Li, Wenjie & Long, Yunfei. (2021). Lexical data augmentation for sentiment analysis. Journal of the Association for Information Science and Technology. 72. 10.1002/asi.24493.

[3] Tan, Chenhao & Lee, Lillian & Tang, Jie & Jiang, Long & Zhou, Ming & Li, Ping. (2011). User-level sentiment analysis incorporating social networks. 10.1145/2020408.2020614.

[4] Azeema Sadia , Fariha Khan and Fatima Bashir. An Overview of Lexicon-Based Approach For Sentiment Analysis. 2018 3rd International Electrical Engineering Conference (IEEC 2018) Feb, 2018 at IEP Centre, Karachi, Pakistan.

[5] Jhanwar, Madan & Das, Arpita. (2018). An Ensemble Model for Sentiment Analysis of Hindi-English Code-Mixed Data.

[6] Ansari, Mohammed Arshad & Govilkar, Sharvari. (2018). Sentiment Analysis of Mixed Code for The Transliterated Hindi and Marathi Texts. International Journal on Natural Language Computing. 7. 10.5121/ijnlc.2018.7202.

[7] Tho, Cuk & Spits Warnars, Harco Leslie Hendric & Soewito, Benfano & Gaol, Ford. (2020). Code-Mixed Sentiment Analysis Using Machine Learning Approach – A Systematic Literature Review. 1-6. 10.1109/ICICoS51170.2020.9299004.

[8] Patra, Braja & Das, Dipankar & Das, Amitava. (2018). Sentiment Analysis of Code-Mixed Indian Languages: An Overview of SAIL_Code-Mixed Shared Task @ICON-2017.

[9] Ansari, Mohammed Arshad & Govilkar, Sharvari. (2018). Sentiment Analysis of Mixed Code for the Transliterated Hindi and Marathi Texts. SSRN Electronic Journal. 10.2139/ssrn.3429694.

[10] Mishra, Pruthwik & Danda, Prathyusha & Dhakras, Pranav. (2018). Code-Mixed Sentiment Analysis Using Machine Learning and Neural Network Approaches.

[11] Roy, Pradeep & Kumar, Abhinav. (2022). Sentiment Analysis on Tamil Code-Mixed Text using Bi-LSTM.

[12] Qaddoura, Raneem & Al-Zoubi, Ala & Faris, Hossam & Almomani, Iman. (2021). A Multi-Layer Classification Approach for Intrusion Detection in IoT Networks Based on Deep Learning. Sensors (Basel, Switzerland). 21. 10.3390/s21092987.

[13] Chakravarthi, Bharathi & Stearns, Bernardo & Arčan, Mihael & Zarrouk, Manel & McCrae, John & Priyadharshini, Ruba & Jayapal, Arun & Sridarane, Sridevy. (2019). Multilingual Multimodal Machine Translation for Dravidian Languages utilizing Phonetic Transcription.

[14] Mishra, Ankit & Saumya, Sunil & Kumar, Abhinav. (2022). Sentiment Analysis of Dravidian-CodeMix Language.

[15] Chakravarthi, Bharathi & Priyadharshini, Ruba & Muralidaran, Vigneshwaran & Suryawanshi, Shardul & Jose, Navya & Elizabeth, Sherly &

McCrae, John. (2020). Overview of the track on Sentiment Analysis for Dravidian Languages in Code-Mixed Text. 21-24. 10.1145/3441501.3441515.