Sentiment Analysis and Homophobia detection of YouTube comments in Code-Mixed Dravidian Languages using machine learning and Transformer models

Josephine Varsha[1], B Bharathi[2] and A. Meenakshi[3]

[1] [2] *Department of CSE, Sri Siva Subramaniya Nadar College of Engineering,Tamil Nadu, India*

[3] *Department of Computer Science and Application, SRM Institute of Science and Technology,Tamil Nadu, India*

Abstract

Sentiment Analysis is the task of identifying the emotions underlying the subjective opinions or emotional responses pertaining to a given topic, be it positive, negative or neutral. Sentiment Analysis is done with the use of natural language processing. Homophobia speech is a type of hate speech directed towards LGBT+ people. This research work presents Sentiment Analysis and Homophobia detection in Youtube comments in Code-Mixed Dravidian Languages with different embeddings using machine learning algorithms. The goal of Task- A is to identify sentiment polarity of the code-mixed dataset of comments, posts in Tamil-English, Malayalam-English, and Kannada-English collected from social media. The goal of Task-B is to identify if the comment is homophobic/transphobic in nature. Our team srmnlp worked with code-mixed form of Tamil, Malayalam and Kannada text provided by the FIRE 2022 organizers. Pre-trained models such as bert, xlm, MPNet were used along with classifiers such as SVM,MLP, Random Forest under the feature extraction techniques like, Count Vectorizer, and TF-IDF. The rankings for sentiment analysis task are, rank 1 in Tamil dataset, rank 6 in Malayalam dataset, rank 7 in Kannada dataset. The highest F1-score of 0.63 was obtained for sentiment analysis in Malayalam dataset, similarly 0.95 was obtained for homophibia detection task in Malayalam dataset. The performance of the proposed system is compared with various machine learning algorithms.

Keywords

Count Vectorizer, TF-IDF, Random Forest, Adaboost, Xlnet, Sentiment, Homophobia, Transphobia, LGBT+

1. Introduction

A person's or a group's reputation can change significantly thanks to a large part of social media. Social media plays a major role in online communication, facilitating users to freely post and share material and express their opinions and thoughts on anything at any time. With the freedom of speech prevailing on social media, there are a few voices that send intentional hate messages toward LGBT+ people. People who identify as LGBT+ are routinely mistreated,

treated unfairly, tortured, and even executed around the world because of the way they appear, the people they love, and who they are. Even with the increasing awareness of the importance of media representation, the amount of general toxicity on the internet remains unchanged McInroy and Craig [1].

The usage of social media has increased dramatically in recent years, yet there are fundamental standards of behaviour that limit free expression in order to preserve a positive environment and prevent online abuse which was mentioned by Chakravarthi et al. [2]. Utilizing the special features of the Internet, such as anonymity, the user is able to have a big impact on other people's lives. Unfortunately, homophobic or transphobic attacks also target LGBT+ individuals who seek consolation online. Because of this, LGBT+ individuals seeking support online are assaulted or mistreated, which has a serious impact on their mental healthMcConnell et al. [3].

Sentiment analysis is the process of determining the sentiments like emotions, that may or may not affect others, in the given text or sentence, or paragraph Kalaivani and Thenmozhi [4]. A text mining task such as Sentiment Analysis helps companies and researchers to extract personal information from source material, to try to understand the social sentiment of a brand or service better Hande et al. [5]. There are many monolingual datasets for Dravidian languages that can be utilised for various types of study in order to discover and extract the emotions from a text.

Sentiment multilingual code-mixed language is an important challenge research area in sentiment analysis research. The organizers proposed the shared task of Dravidian–CodeMix-Fire to classify the sentiment polarity as positive, negative, neutral, mixed emotions, unknown state or not in the Tamil-English and Malayalam-English languages. Based on the sentiment analysis, the task is to detect the mixed feelings of online users and to prevent unusual activities, depression, and criminal activities.

To make it simpler to ban non-LGBT+ content and drive the Internet toward equality, diversity, and inclusion, it is urgently necessary to find and filter homophobic and transphobic materials online. It is also crucial to evaluate and discern the sentiment of a text. Our team worked on the shared task of Homophobia detection and Sentiment Analysis of Dravidian-CodeMix-Fire 2022. For this proposed work, we used the datasets for code-mixed Tamil, code-mixed Malayalam, code- mixed Kannada text for the sentiment analysis task, comprising comments from YouTube. Similarly, we used 3 datasets, Tamil, Malayalam, and code-mixed Tamil text for the Homophobia Detection which also comprised comments from YouTube. We have used feature extraction methods namely, Count Vectorizer and TF-IDF to extract the feature vector, and classifier models like SVM, MLP, AdaBoost, and transformer models like Bert, XLM. In this study, we examine the effectiveness of various learning models in identifying homophobia and sentiment analysis, under the working notes of FIRE 2022, in the proceedings of dravidian code-mix 2022 Shanmugavadivel et al. [6].

There are five sections in the paper. Section 2, the related works on sentiment analysis and homophobia detection for Tamil and other languages, in the field of artificial intelligence. In Section 3 of this study, the methodology suggested for the model and the techniques used are thoroughly detailed. Results and observations are discussed in section 4. The paper is concluded in Section 5.

2. Related Work

To recognize the emotions included in the text sequences, The Seq2Seq deep neural network was used by the authors of Thenmozhi et al. [7] to build various models by adjusting settings like the number of layers, units, and attention wrappers, with and without delimiter string, and train-validation split. Research on a development set shows that the two-layered LSTM with the Normed Bahdanau attention mechanism, delimiter string, and train-validation split outperforms all alternatives.

To enhance study in the under-resourced Dravidian languages, Chakravarthi et al. [8] created an annotation scheme and obtained a high level of inter-annotator agreement in terms of Krippendorff from volunteer annotators on contributions gathered using Google Form. For each class, baselines with gold standard annotations for recall, precision, and F-Score were constructed and presented.

To classify the emotions in Tamil, Sampath et al. [9] produced two additional datasets with fine-grained emotions. They plan to increase the dataset's size and include more exact emotional descriptors in order to boost the system's efficacy.

A method for performing sentiment analysis in Tamil texts using the k-means clustering and k-nearest neighbor algorithm was proposed in Thavareesan and Mahesan [10]. For all k values, class-wise clustering with m-folds of the training set beat the alternative strategies and the baseline method. This method's ability to produce improved accuracy with fewer k-nearest neighbor classifier training samples is another enhancement.

Chakravarthi et al. [11] performed the first joint effort for classifying YouTube comments using the Tamil, English, and Tamil-English (code-mixed) dataset to detect homophobia and transphobia. To deal with data imbalance and multilingualism, the most effective solution used XLM, RoBERTa pre-trained language models for zero-shot learning.

The first dataset on homophobia and transphobia in multilingual comments in Tamil, English, and Tamil-English was produced by Chakravarthi et al. [2]. This study offered a dataset with high-quality, expert homophobic and transphobic content classification from multilingual YouTube comments as well as a hierarchical granular homophobic and transphobic taxonomy.

The authors of S et al. [12], a synthesis of two knowledge bases of words and emojis, made use of an Emotion Word Ontology. A list of emotional terms that are matched to the appropriate emotion class can be found in the Word Knowledge Base. Similar to this, the Emoji Knowledge Base includes emotion icons that correspond to the associated emotions.

A model that initially learns to extract the sub-elements (holders, targets, and expressions) using sequence labelers was offered by Anantharaman et al. [13] in their article.

The authors of [14] [15] [16], uses machine learning algorithms and transformer models for sentiment analysis and homophobia detection tasks.

3. Proposed approach

In this section, we have described our implementation of feature extraction and machine learning algorithms. Further, we will evaluate the performance of the various algorithms we've employed along with the feature extraction procedure. The architecture of the proposed model

is illustrated down below along with the steps involved in Fig. 1 and Fig. 2
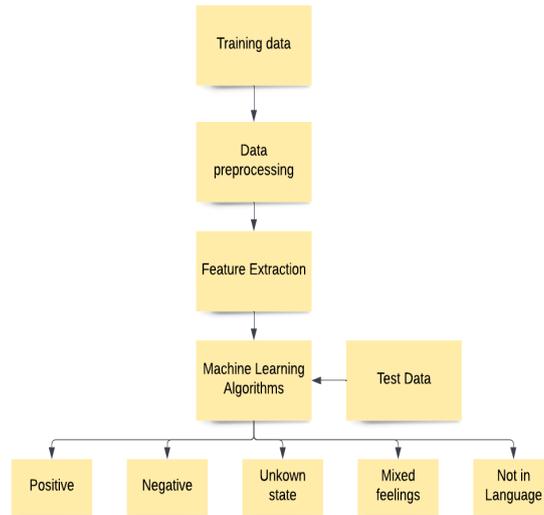
Figure 1: The workflow of the proposed work for Sentiment Analysis
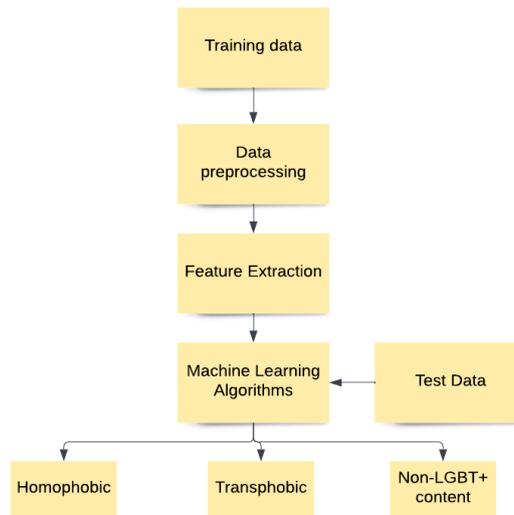
Figure 2: The workflow of the proposed system for Homophobia Detection

The datasets provided by the FIRE 2022 organizers for the Sentiment Analysis, and Homophobic Detection [2] consisted of code-mixed text in Tamil, Malayalam, and Kannada each consisting of Youtube comments. The details regarding the datasets are provided in Table 1 and Table 2.

Language	Training	Development	Test
TamilEnglish	35657	3963	650
MalayalamEnglish	15889	1767	1963
KannadaEnglish	6213	692	769

Table 1
Sentiment Analysis dataset description

Language	Training	Development	Test
Tamil	2663	667	650
Malayalam	3115	867	1214
TamilEnglish	3862	967	1208

Table 2
Homophobia detection dataset description

3.1. Data-set Analysis

Determine whether a particular comment has an emotion and the sentiment it represents is the aim of the first job. This assignment involves polarity categorization at the message level. Systems must categorize a YouTube comment into positive, negative, neutral, or mixed emotions.

The dataset issued by the FIRE 2022 organizers, consists of the 3 datasets, namely, training set, development set, and test set, each consisting of 15889, 1767, and 1963 instances respectively for the Malayalam code-mixed text, 35657, 3963, and 650 instances respectively for the Tamil code-mixed text, and 6213, 692, and 769 instances respectively for the Kannada code-mixed text. It contained the sequence of texts that include user utterances along with the context, followed by the sentiment class label. The task was to identify and label them under any of the following: Positive, Negative, Mixed Feelings, Unknown State, Not in Language

The goal of the second task is to check whether a specific comment contains homophobic, or transphobic speech and if not those comments should be labeled, Non-LGBTQ+. We were provided with comments extracted from social media platforms and developed submit systems to predict whether it is homophobic/transphobic in nature. The seed data for this task is the Homophobia/Transphobia Detection dataset, a collection of comments from YouTube. This dataset consists of manually annotated comments indicating whether the text is homophobic/-transphobic or not.

The dataset provided by FIRE 2022 organizers, consisted of the training set, development set, and test set of 2663, 667, and 650 instances respectively for the Tamil text, 3115, 867, and 1214 instances respectively for the Malayalam text, and 3862, 967, and 1208 instances respectively for the Tamil-English code-mixed text. It contained text sequences that include user utterances along with the context, followed by the homophobic, or transphobic class label. The task was to identify and label them under any of the following: Homophobia, Transphobia, Non-LGBT content.

3.2. Data Pre-processing

Since real-world data frequently contains noise, and missing values, and may be in an unusable format that cannot be directly used for machine learning models, data pre-processing is crucial for any machine learning challenge. Data preprocessing is necessary to clean the data and prepare it for a machine learning model, which also improves the model's accuracy and effectiveness. Before categorising, the dataset must first be cleaned and processed.

The Natural Language Toolkit, often known as the NLTK package, which was created to interact with the NLP, has been used to implement data processing (Natural Language Processing). Different text-processing libraries are provided for categorization, tokenization, parsing, semantic reasoning, etc. Functions were used to clean and scrape the text, remove URLs, numerals, and tags.

We were able to extract the tokens from the string using the RegexpTokenizer() function and the tokenize. regexp() module. Tokenizing is an essential step When it comes to cleaning the text. It is employed to divide the text into words or sentences, dividing it into more manageable chunks while maintaining its meaning.

3.3. Methodology

The datasets were used with machine learning models that had various embeddings, namely TF-IDF, count vectorizer, BERT. Classifiers namely Random Forest, Support Vector Machine, and Multilayer Perceptron were used to build the baseline models with the above embeddings. After removing the essential features from the processed data, these classifier models were used. These models were fine-tuned after being trained on the training dataset using the development set. By speculating on the labels for the held-out test set, the model's effectiveness was assessed. The models considered and their efficiency are mentioned in table 3 and 4 for the Tamil dataset, 5 and 6 for the Malayalam dataset, and 7 and 8 for the Kannada dataset, where the performance metrics of all the datasets in the sentiment analysis task have been tabulated.

A similar approach was chosen for the Homophobia Detection task. The same feature extraction techniques were employed to decrease the number of features in the input, along with the classifier models mentioned above for vector classification. The performance of the dataset has been tabulated below to measure the efficiency. The models under consideration and the performance along with the development dataset have been tabulated in 9 and 10 for the Tamil dataset, 11 and 12 for the Malayalam dataset. Similarly, 13 and 14 show the tabulations for the code-mixed Tamil dataset.

Feature Extraction aims to reduce the number of features in the datasets by creating new features from the existing ones and then discarding the original features. The feature extraction methods we followed are

- Count Vectorizer:
  The Count Vectorizer feature extractor breaks down a sentence or any text into smaller words by performing preprocessing tasks. This approach converts text into a vector form that is dependent on the frequency in which each word occurs in the text.
- TF-IDF:
  TF-IDF stands for term frequency-inverse document frequency. For each word in the

Feature Extraction	Classifier	Precision	Recall	F1-score	Accuracy
Count Vectorizer	SVM	0.60	0.63	0.55	0.63
Count Vectorizer	MLP	0.58	0.61	0.59	0.61
Count Vectorizer	Random Forest	0.59	0.61	0.52	0.61
TF-IDF	SVM	0.32	0.57	0.41	0.57
TF-IDF	MLP	0.57	0.61	0.58	0.61
TF-IDF	Random Forest	0.59	0.61	0.51	0.61
TF-IDF	Adaboost	0.48	0.54	0.49	0.31
TF-IDF	Gradient Boosting Classifier	0.51	0.59	0.50	0.59
TF-IDF	Extratrees	0.53	0.55	0.53	0.59

Table 3
Performance analysis of the proposed methodology using development data for Tamil Dataset -Sentiment Analysis

Pre-Trained Model	Precision	Recall	F1-score	Accuracy
XLM feature extraction + MLP classifier + Tamil	0.51	0.51	0.51	0.51
MPNet feature extraction + MLP classifier + Tamil	0.52	0.54	0.53	0.54
bert feature extraction +MLP classifier + Tamil	0.53	0.55	0.54	0.55

Table 4
Performance analysis of the selected transformer model using development data for Tamil Dataset -Sentiment Analysis

corpus relative to the dataset, the TF-IDF score is calculated, and the data is then put into a vector. Each document in the corpus would have its own vector, and each word in the entire collection of documents would have a TF-IDF score in the vector. Typically, this is employed in fields like text mining and information retrieval.

In our proposed approach, TF-IDF, Count vectorizer, and BERT embeddings were extracted from the dataset. Then the extracted features were trained with multiple machine learning models such as SVM classifier, MLP classifier, random forest classifier, Ada boost classifier, Gradient Boosting classifier, and ExtraTrees classifier. The experiments were conducted for Tamil code-mixed, Malayalam code-mixed, and Kannada code-mixed data sets for the Sentiment Analysis task, and Tamil, Malayalam, and Tamil-English data sets for the Homophobia Detection task, and the models that obtained the best results were used to generate the scores for the test dataset.

4. Results and Analysis

In this section, we will discuss the performance of the techniques, and models implemented and chose the best accurate model that will generate the test labels.

Feature Extraction	Classifier	Precision	Recall	F1-score	Accuracy
Count Vectorizer	SVM	0.62	0.61	0.60	0.61
Count Vectorizer	MLP	0.63	0.63	0.63	0.63
Count Vectorizer	Random Forest	0.62	0.60	0.58	0.60
TF-IDF	SVM	0.41	0.41	0.26	0.41
TF-IDF	MLP	0.62	0.62	0.62	0.62
TF-IDF	Random Forest	0.62	0.60	0.57	0.60
TF-IDF	Adaboost	0.54	0.53	0.52	0.53
TF-IDF	Gradient Boosting Classifier	0.55	0.58	0.56	0.58
TF-IDF	Extratrees	0.57	0.57	0.56	0.57

Table 5
Performance analysis of the proposed methodology using development data for Malayalam Dataset
-Sentiment Analysis

Pre-Trained Model	Precision	Recall	F1-score	Accuracy
XLM feature extraction + MLP classifier + Malayalam	0.54	0.55	0.54	0.55
MPNet feature extraction + MLP classifier + Malayalam	0.54	0.54	0.54	0.54
bert feature extraction + MLP classifier + Malayalam	0.54	0.54	0.54	0.54

Table 6
Performance analysis of the selected transformer model using development data for Malayalam Dataset
-Sentiment Analysis

4.1. Sentiment Analysis

4.1.1. TamilEnglish Dataset

Feature Extraction techniques namely, Count vectorizer, TFIDF, and BERT embeddings were employed to extract the necessary features from the youtube comments specified in Tamil-English code mixed text. These extracted features in the form of vectors were trained along with different machine learning algorithms and then evaluated using the development data. The results are tabulated down below, in Table 3 and Table 4. From the table, we see that Count Vectorizer with the Random Forest model fetched the best F1-score of 0.61.

4.1.2. MalayalamEnglish Dataset

Feature Extraction techniques namely, Count vectorizer, TFIDF, and BERT embeddings were employed to extract the necessary features from the youtube comments specified in Malayalam-English code mixed text. These extracted features in the form of vectors were trained along with different machine learning algorithms and then evaluated using the development data. The results are tabulated in Table 5 and Table 6. From the table, we see that Count Vectorizer with the MLP classifier fetched the best F1-score of 0.63.

Feature Extraction	Classifier	Precision	Recall	F1-score	Accuracy
Count Vectorizer	SVM	0.54	0.56	0.49	0.56
Count Vectorizer	MLP	0.59	0.62	0.59	0.62
Count Vectorizer	Random Forest	0.62	0.63	0.61	0.63
TF-IDF	SVM	0.22	0.46	0.29	0.46
TF-IDF	MLP	0.60	0.63	0.60	0.63
TF-IDF	Random Forest	0.62	0.63	0.59	0.63
TF-IDF	Adaboost	0.49	0.53	0.50	0.53
TF-IDF	Gradient Boosting Classifier	0.55	0.58	0.55	0.58
TF-IDF	Extratrees	0.59	0.60	0.58	0.60

Table 7
Performance analysis of the proposed methodology using development data for Kannada Dataset -Sentiment Analysis

Pre-Trained Model	Precision	Recall	F1-score	Accuracy
XLM feature extraction +MLP classifier + Kannada	0.56	0.56	0.56	0.56
MPNet feature extraction +MLP classifier + Kannada	0.58	0.56	0.57	0.56
bert feature extraction +MLP classifier + Kannada	0.58	0.59	0.58	0.59

Table 8
Performance analysis of the selected transformer model using development data for Kannada Dataset -Sentiment Analysis

4.1.3. KannadaEnglish Dataset

Feature Extraction techniques namely, Count vectorizer, TFIDF, and BERT embeddings were employed to extract the necessary features from the youtube comments specified in Kannada-English code mixed text. These extracted features in the form of vectors were trained along with different machine learning algorithms and then evaluated using the development data. The results are tabulated in Table 7 and Table 8. From the table, we see that Count Vectorizer with the Random Forest model fetched the best F1-score of 0.61.

4.2. Homophobia Detection

4.2.1. Tamil Dataset

Feature Extraction techniques namely, Count vectorizer, TFIDF, and BERT embeddings were employed to extract the necessary features from the youtube comments specified in Tamil text. These extracted features in the form of vectors were trained along with different machine learning algorithms and then evaluated using the development data. The results are tabulated in Table 9 and Table 10. From the table, we see that TF-IDF with the Random Forest model fetched the best F1-score of 0.88.

Feature Extraction	Classifier	Precision	Recall	F1-score	Accuracy
Count Vectorizer	SVM	0.77	0.81	0.76	0.81
Count Vectorizer	MLP	0.86	0.86	0.83	0.86
Count Vectorizer	Random Forest	0.89	0.89	0.83	0.89
TF-IDF	SVM	0.62	0.79	0.70	0.79
TF-IDF	MLP	0.85	0.85	0.82	0.85
TF-IDF	Random Forest	0.89	0.89	0.88	0.89
TF-IDF	Adaboost	0.77	0.79	0.76	0.79
TF-IDF	Gradient Boosting Classifier	0.80	0.81	0.80	0.81
TF-IDF	Extratrees	0.87	0.87	0.87	0.87

Table 9
Performance analysis of the proposed methodology using development data for Tamil Dataset - Homophobia Detection

Pre-Trained Model	Precision	Recall	F1-score	Accuracy
XLM feature extraction + MLP classifier + Tamil	0.86	0.87	0.86	0.87
MPNet feature extraction + MLP classifier + Tamil	0.86	0.86	0.86	0.86
bert feature extraction + MLP classifier + Tamil	0.88	0.89	0.88	0.89

Table 10
Performance analysis of the selected transformer model using development data for Tamil Dataset -Homophobia Detection

4.2.2. Malayalam Dataset

Feature Extraction techniques namely, Count vectorizer, TFIDF, and BERT embeddings were employed to extract the necessary features from the youtube comments specified in Malayalam text. These extracted features in the form of vectors were trained along with different machine learning algorithms and then evaluated using the development data. The results are tabulated in Table 11 and Table 12. From the table, we see that Count Vectorizer and TF-IDF with the Random Forest model fetched the best F1-score of 0.96 along with the ExtraTrees classifier.

4.2.3. Tamil-English Dataset

Feature Extraction techniques namely, Count vectorizer, TFIDF, and BERT embeddings were employed to extract the necessary features from the youtube comments specified in Tamil-English code mixed text. These extracted features in the form of vectors were trained along with different machine learning algorithms and then evaluated using the development data. The results are tabulated in Table 13 and Table 14. From the table, we see that Count Vectorizer with the SVM, MLP, and TF-IDF with MLP fetched the best F1-score of 0.86.

The development dataset was used for evaluating the performance of the models after training them. The final performance results for the task are recorded in Table 15 and 16.

Feature Extraction	Classifier	Precision	Recall	F1-score	Accuracy
Count Vectorizer	SVM	0.85	0.84	0.81	0.84
Count Vectorizer	MLP	0.93	0.93	0.93	0.93
Count Vectorizer	Random Forest	0.96	0.96	0.96	0.96
TF-IDF	SVM	0.64	0.80	0.71	0.80
TF-IDF	MLP	0.91	0.91	0.90	0.91
TF-IDF	Random Forest	0.96	0.96	0.96	0.96
TF-IDF	Adaboost	0.77	0.79	0.77	0.79
TF-IDF	Gradient Boosting Classifier	0.89	0.89	0.89	0.89
TF-IDF	Extratrees	0.96	0.96	0.96	0.96

Table 11
Performance analysis of the proposed methodology using development data for Malayalam Dataset
-Homophobia Detection

Pre-Trained Model	Precision	Recall	F1-score	Accuracy
XLM feature extraction + MLP classifier + Malayalam	0.92	0.92	0.92	0.92
MPNet feature extraction + MLP classifier + Malayalam	0.92	0.92	0.92	0.92
bert feature extraction + MLP classifier + Malayalam	0.95	0.95	0.95	0.95

Table 12
Performance analysis of the selected transformer model using development data for Malayalam Dataset
-Homophobia Detection

Feature Extraction	Classifier	Precision	Recall	F1-score	Accuracy
Count Vectorizer	SVM	0.86	0.89	0.86	0.89
Count Vectorizer	MLP	0.86	0.89	0.86	0.89
Count Vectorizer	Random Forest	0.80	0.89	0.84	0.89
TF-IDF	SVM	0.80	0.89	0.84	0.89
TF-IDF	MLP	0.87	0.89	0.86	0.89
TF-IDF	Random Forest	0.80	0.89	0.84	0.89
TF-IDF	Adaboost	0.84	0.88	0.85	0.88
TF-IDF	Gradient Boosting Classifier	0.84	0.86	0.84	0.86
TF-IDF	Extratrees	0.84	0.89	0.85	0.89

Table 13
Performance analysis of the proposed methodology using development data for Code-mixed Dataset
-Homophobia Detection

Pre-Trained Model	Precision	Recall	F1-score	Accuracy
XLM feature extraction + MLP classifer + code-mixed text	0.85	0.87	0.86	0.87
MPNet feature extraction + MLP classifer + code-mixed text	0.86	0.87	0.86	0.87
bert feature extraction + MLP classifer + code-mixed text	0.85	0.88	0.86	0.88

Table 14
Performance analysis of the selected transformer model using development data for Code-mixed Dataset
-Homophobia Detection

No	Dataset	Accuracy	Precision	Recall	F1-score
1	Tamil	0.13	0.39	0.13	0.18
2	Tamil	0.10	0.42	0.10	0.15
3	Tamil	0.11	0.37	0.11	0.16
1	Malayalam	0.63	0.63	0.63	0.63
2	Malayalam	0.63	0.62	0.63	0.62
3	Malayalam	0.57	0.56	0.57	0.56
1	Kannada	0.62	0.59	0.62	0.58
2	Kannada	0.62	0.60	0.62	0.58
3	Kannada	0.55	0.52	0.55	0.52

Table 15
Performance analysis of the proposed methodology using test data for Sentiment Analysis

No	Dataset	Accuracy	Precision	Recall	F1-score
1	Tamil	0.53	0.38	0.53	0.38
2	Tamil	0.53	0.37	0.53	0.39
3	Tamil	0.51	0.42	0.51	0.38
1	Malayalam	0.89	0.90	0.89	0.88
2	Malayalam	0.87	0.87	0.87	0.85
3	Malayalam	0.95	0.95	0.95	0.95
1	Kannada	0.90	0.89	0.95	0.89
2	Kannada	0.90	0.81	0.90	0.85
3	Kannada	0.89	0.86	0.89	0.87

Table 16
Performance analysis of the proposed methodology using test data for Homophobia Detection

5. Conclusion

In this study, we examined the test datasets' baseline accuracy for various models and model variants. The challenge at hand was to determine whether a comment has sentiment and whether it disparages LGBT+ individuals in any way. Sentiment analysis and homophobia detection are in high demand on social media. Our team submitted these findings after taking part in the FIRE 2022 competition. For both tasks, our models performed at a baseline, although performance can be increased by incorporating beneficial features.

References

[1] L. McInroy, S. Craig, Transgender representation in offline and online media: Lgbtq youth perspectives, Journal of Human Behavior in the Social Environment 25 (2015) 1–12. doi:10.1080/10911359.2014.995392.

[2] B. R. Chakravarthi, R. Priyadharshini, R. Ponnusamy, P. K. Kumaresan, K. Sampath, D. Thenmozhi, S. Thangasamy, R. Nallathambi, J. P. McCrae, Dataset for identification of homophobia and transophobia in multilingual youtube comments, 2021. URL: https://arxiv.org/abs/2109.00227. doi:10.48550/ARXIV.2109.00227.

[3] E. McConnell, A. Clifford, A. Korpak, G. Phillips, M. Birkett, Identity, victimization, and support: Facebook experiences and mental health among lgbtq youth, Computers in Human Behavior 76 (2017) 237–244. doi:10.1016/j.chb.2017.07.026, publisher Copyright: © 2017 Copyright: Copyright 2017 Elsevier B.V., All rights reserved.

[4] A. Kalaivani, D. Thenmozhi, Ssn_nlp_mlrg@dravidian-codemix-fire2020: Sentiment code-mixed text classification in tamil and malayalam using ulmfit, in: FIRE, 2020.

[5] A. Hande, S. U. Hegde, R. Priyadharshini, R. Ponnusamy, P. K. Kumaresan, S. Thavareesan, B. R. Chakravarthi, Benchmarking multi-task learning for sentiment analysis and offensive language identification in under-resourced dravidian languages, CoRR abs/2108.03867 (2021). URL: https://arxiv.org/abs/2108.03867. arXiv:2108.03867.

[6] K. Shanmugavadivel, M. Subramanian, P. K. Kumaresan, B. R. Chakravarthi, B. B, S. Chinnaudayar Navaneethakrishnan, L. S.K, T. Mandl, R. Ponnusamy, V. Palanikumar, M. B. J, Overview of the Shared Task on Sentiment Analysis and Homophobia Detection of YouTube Comments in Code-Mixed Dravidian Languages, in: Working Notes of FIRE 2022 - Forum for Information Retrieval Evaluation, CEUR, 2022.

[7] D. Thenmozhi, A. Chandrabose, S. Sharavanan, et al., Ssn_nlp at semeval-2019 task 3: Contextual emotion identification from textual conversation using seq2seq deep neural network, in: Proceedings of the 13th International Workshop on Semantic Evaluation, 2019, pp. 318–323.

[8] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, N. Jose, S. Suryawanshi, E. Sherly, J. P. McCrae, DravidianCodeMix: sentiment analysis and offensive language identification dataset for Dravidian languages in code-mixed text, Language Resources and Evaluation (2022). URL: https://doi.org/10.1007/s10579-022-09583-7. doi:10.1007/s10579-022-09583-7.

[9] A. Sampath, T. Durairaj, B. R. Chakravarthi, R. Priyadharshini, S. Chinnaudayar Navaneethakrishnan, K. Shanmugavadivel, S. Thavareesan, S. Thangasamy, P. Krishnamurthy, A. Hande, S. Benhur, S. Ponnusamy, Kishor Kumar Pandiyan, Findings of the shared task on Emotion Analysis in Tamil, in: Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, 2022.

[10] S. Thavareesan, S. Mahesan, Sentiment analysis in tamil texts using k-means and k-nearest neighbour, in: 2021 10th International Conference on Information and Automation for Sustainability (ICIAfS), 2021, pp. 48–53. doi:10.1109/ICIAfS52090.2021.9605839.

[11] B. R. Chakravarthi, R. Priyadharshini, T. Durairaj, J. McCrae, P. Buitelaar, P. Kumaresan, R. Ponnusamy, Overview of the shared task on homophobia and transphobia detection in social media comments, in: Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 369–377. URL: https://aclanthology.org/2022.ltedi-1.57. doi:10.18653/v1/2022.ltedi-1.57.

[12] V. S, K. Rajan, A. S, R. Sivanaiah, S. M. Rajendram, M. T T,

Varsini_and_Kirthanna@DravidianLangTech-ACL2022-emotional analysis in Tamil, in: Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 165–169. URL: https://aclanthology.org/2022.dravidianlangtech-1.26. doi:10.18653/v1/2022.dravidianlangtech-1.26.

[13] K. Anantharaman, D. K, J. Pt, A. S, R. Sivanaiah, S. M. Rajendram, M. T T, SSN_MLRG1 at SemEval-2022 task 10: Structured sentiment analysis using 2-layer BiLSTM, in: Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), Association for Computational Linguistics, Seattle, United States, 2022, pp. 1324–1328. URL: https://aclanthology.org/2022.semeval-1.184. doi:10.18653/v1/2022.semeval-1.184.

[14] K. Swaminathan, B. Bharathi, G. Gayathri, H. Sampath, Ssncse_nlp@ lt-edi-acl2022: Homophobia/transphobia detection in multiple languages using svm classifiers and bert-based transformers, in: Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion, 2022, pp. 239–244.

[15] B. Bharathi, G. Samyuktha, Machine learning based approach for sentiment analysis on multilingual code mixing text, in: Working Notes of FIRE 2021-Forum for Information Retrieval Evaluation (Online). CEUR, 2021.

[16] N. N. A. Balaji, B. Bharathi, J. Bhuvana, Ssncse_nlp@ dravidian-codemix-fire2020: Sentiment analysis for dravidian languages in code-mixed text., in: FIRE (Working Notes), 2020, pp. 554–559.