

MINI PROJECT REPORT

ON

ABUSIVE COMMENT IDENTIFICATION IN

MALAYALAM LANGUAGE



School of Digital Sciences
Kerala University of Digital Sciences, Innovation and Technology
(Digital University Kerala)

Abusive Comment Identification in Malayalam Language

Submitted by

ABHISHNAV M (Register Number: 223003)

GAYATHRI ANIL (Register Number: 223027)

HARSHIN RONALDO P R (Register Number: 223029)

In partial fulfilment of the requirements for the award of

Master of Science in COMPUTER SCIENCE WITH DATA ANALYTICS of



School of Digital Sciences

Kerala University of Digital Sciences, Innovation and Technology

(Digital University Kerala)

Technocity Campus, Thiruvananthapuram, Kerala – 695317

BONAFIDE CERTIFICATE

This is to certify that the project report entitled “**Abusive Comment Identification in Malayalam Language**” Submitted by ----- (Reg. No:2230--) in partial fulfillment of the requirements for the award of **Master of Science in Computer Science with Specialization in Data Analytics** is a bonafide record of the work carried out at “**Kerala University of Digital Sciences, Innovation and Technology**”.



Supervisor

Dr. ANOOP V.S

School of Digital Sciences

DUK

Course Coordinator

Dr. ANOOP V.S

School of Digital Sciences

DUK

Head of Institution

Prof. SAJI GOPINATH

Vice Chancellor

DUK

DECLARATION

I, -----, a student of **MSc Computer Science with Specialization in Data Analytics**, hereby declare that this report is substantially the result of my own work, except where explicitly indicated in the text, and has been carried out during the period **October 2023 – January 2024**.

Place: Trivandrum

Date: 08-01-2024

.....

Student's signature

ACKNOWLEDGEMENT

I want to express my deep gratitude to **Dr. Anoop V.S**, Professor and Course Coordinator, School of Digital Sciences at Digital University Kerala, Trivandrum, for his invaluable guidance and mentorship, which were instrumental in the successful completion of this project. I extend my appreciation to **Prof. Saji Gopinath**, Vice Chancellor, Digital University Kerala, for granting access to the university's facilities and resources. Furthermore, I am deeply grateful to my friends and family for their steadfast support, inspiration, and aid throughout the project's journey.

ABSTRACT

YouTube is a video-sharing and social media platform where users create profiles and share videos for their followers to view, like, and comment on. Abusive comments on videos or replies to other comments may be offensive and detrimental for the mental health of users on the platform. It is observed that often the language used in these comments is informal and does not necessarily adhere to the formal syntactic and lexical structure of the language. With the increasing presence of abusive language and toxicity in online platforms, particularly YouTube, the need for efficient and accurate identification of such content in regional languages like Malayalam has become imperative. Therefore, creating a rule-based system for filtering out abusive comments is challenging.

This project aims to utilize natural language processing and deep learning approaches for identifying abusive comments posted to the YouTube that are written in Malayalam, which is one of the agglutinative languages spoken in the state of Kerala. For this, we use datasets of abusive comments in Malayalam and code-mixed Malayalam-English languages that are extracted from YouTube videos. Different machine learning approaches with pre-trained language models will be used to implement the classifier. Overall, this project may help in detection of abusive comments in Malayalam and may help in creation of comment-filters for Malayalam language on YouTube.

CONTENTS

CHAPTER 1 INTRODUCTION	2
1.2 PROBLEM DEFINITION	3
CHAPTER 2: METHODOLOGY	4
2.2 EXPERIMENTAL ANALYSIS.....	9
CHAPTER 3: RESULTS AND INSIGHTS	13
3.2 Streamlit App : Comment Classification App.....	16
3.3 CONCLUSION	18
REFERENCE.....	19

TABLE OF FIGURES

Figure 1: Abusive comment identification process	7
Figure 2: Comment Hierarchy for Binary classification	7
Figure 3: Comment Hierarchy for fine-grained classification	8
Figure 4: Annotation of scraped comments using Label Studio	10
Figure 5: Splitting into training and testing sets	10
Figure 6: Tokenizing	11
Figure 7: Tokenizing X_train	11
Figure 8: Tokenizing X_test	12
Figure 9: Decision Tree Classifier	12
Figure 10: Classification report for first phase	13
Figure 11: Predicted labels and Actual labels for first phase	14
Figure 12: Classification report of second phase	14
Figure 13: Predicted labels and Actual labels for second phase	15
Figure 14: Entering the comment in Malayalam or code-mixed Malayalam-English language	16
Figure 15: Streamlit Demo Output 1	16
Figure 16: Streamlit Demo Output 2	17

CHAPTER 1

INTRODUCTION

Social Media is an integral part of our life in this digital age. Online Social Networks are gaining more and more importance and their usage are high more than ever. They are the primary and go-to source for day to day information, news, and so on. They are also widely used for entertainment purposes. The amount of digital information distributed through social media are also having an exponential increase. However, though social network has numerous benefits, there are evidences of high increase of malevolent activities in these networks. Exploitation of Online social networks to spread hate speech over other individuals and groups, causing mental harm and breaking the online guidelines of particular social media are getting more common over time.

YouTube, Facebook, Instagram, Twitter are some of the common social media networks that help people connect on a global scale. These platforms allows users to share their posts, photos, videos, thoughts etc. These are accompanied by comment sections and sometimes there can be unpleasant interactions. YouTube is one of the most popular social media for sharing videos. YouTube doesn't require specific permission granting procedure like other social media, which allows anyone to comment on videos and this often leads to posting of offensive comments on videos targeting individuals or groups. This can lead to triggering of more such activities. Such offensive comments could affect mental well-being of others. Hate-speeches are very common in such comment sections of YouTube videos. This can lead to depression, lacking of interest to socialize or interact.

Since the instances of use of abusive language on social media are growing, our project aims at detection and understanding of online abusive contents in low-resourced Malayalam language. By abusive content, we refer to any sort of comment that is insulting, abusive, discrimination against individuals or groups based on their origin, ethnicity, gender, sex or religion. People motivated by racism, sexism, homophobia etc. causes incidents that are violent both online and offline.

We scraped comments from different YouTube videos to create required dataset. Then created a dataset of fine-grained annotation of offensive comments in Malayalam and Malayalam-English code-mixed data. There are no significant attempts to tackle this problem for Malayalam and code-mixed Malayalam-English data. Abusive comments were detected by classifying comments on YouTube by using machine learning models like Decision tree.

We did a conventional Binary classification of abusive content as not offensive or offensive as well as a fine-grained classification as not offensive, Offensive Untargeted, Offensive Targeted Individual, Offensive Targeted Group.

1.2 PROBLEM DEFINITION

The problem addressed in this report is the development of a comprehensive machine learning model for abusive comment detection on YouTube. The objective is to create a robust and accurate model that can classify comments into offensive or not offensive and then further fine-grained classification of offensive comments.

CHAPTER 2

METHODOLOGY

LIBRARIES

Pandas:

Pandas is an open-source Python library. It is primarily designed for effortlessly and intuitively handling relational or labeled data. It offers diverse data structures and operations to manipulate numerical data and time series. Known for its speed and impressive performance, Pandas enhances productivity for users dealing with data analysis tasks.

NumPy:

NumPy is a package for processing arrays. It offers high-performance multidimensional array objects and corresponding tools. Serving as the core package for scientific computing in Python, it is also effective as a resourceful multi-dimensional container for generic data.

Transformers:

The Transformers library is like a helpful tool for computers that makes it easier for them to understand and use human language. It's a useful resource for tasks like understanding the meaning of sentences or generating text that sounds natural. It simplifies working with language on computers and is widely used for various language-related applications.

Joblib:

It is a python library used for efficiently saving and loading large data structures like machine learning models, Numpy arrays etc.

Scikit-learn:

Scikit-learn is an open-source Python library that implements a range of machine learning, pre-processing, cross-validation, and visualization algorithms using a unified interface.

Keras:

Keras is a deep learning API written in python language, which is running on the top of the machine learning platform TensorFlow. It provides industry-strength performance and scalability. It is mainly focused on ease of use, elegance of code, speed of debugging and maintainability.

DATASETS

The following datasets were used to build the machine learning model.

mal_full_offensive_train, **mal_full_offensive_test** and **mal_full_offensive_dev** datasets were available online. These dataset included comments from YouTube in Malayalam and Malayalam-English code-mixed language. The comments were categorized into Not_offensive, Offensive_Untargetede, Offensive_Targeted_Insult_Indivitual, Offensive_Targeted_Insult_Group and not_malayalam.

Collectively they included more than 16000 data points out of which more than 14000 where annotated not offensive. So we trimmed the number of not offensive comments to the required number.

Additional dataset was created specifically for the purpose of this project since the given dataset was not enough and biased.

YouTube comments were scraped using a scrape tool and then annotation was done, labelling into the specified categories using label studio.

All the datasets were integrated into a single dataset.

YOUTUBE VIDEO COMMENTS SCRAPING

YouTube videos comments were collected. We used videos of famous YouTubers about LGBTQ+, videos related to politics, women rights. All of them had significant amount of offensive comments related transphobia, hate speech, misogony etc. We required comments that in Malayalam and code-mixed Malayalam English language. The Comments were scraped. The raw data was produced in csv format for annotation.

ANNOTATION

Annotation is the process of annotating the raw document, in our project the comments that were scraped from YouTube. These comments were annotated into Not_offensive, Offensive_Untargetede, Offensive_Targeted_Insult_Indivitual, Offensive_Targeted_Insult_Group and not_malayalam using Label Studio.

BERT ALGORITHM

BERT, which stands for Bidirectional Encoder Representations from Transformers, is a sophisticated natural language processing (NLP) algorithm. Unlike traditional language models that read text in one direction (either left to right or right to left) BERT does both direction. It analyzes the entire context of a word by considering both the words that comes before it and after it. We use a variant of BERT known as bert-base-multilingual-cased.

BERT is based on the Transformer architecture. It is a type of neural network designed for sequence-to-sequence tasks. What sets BERT apart is its pre-training process on large amount of unlabelled text data. This allows BERT to learn the relationships between words and their meanings.

The knowledge BERT got during pre-training makes it a powerful tool for various NLP tasks. When question answering or sentiment analysis, BERT can understand and generate human-like responses, making it a crucial component in improving the understanding of natural language by computers.

LSTM MODEL

LSTM (Long Short-Term Memory) model is a type of recurrent neural network (RNN) architecture. It is designed to address the vanishing gradient problem in traditional RNNs. LSTMs are well-suited particularly for the process and prediction of sequences of data. The information can be stored and retrieved with the help of memory cells, over a long period. This enables them to capture dependencies and patterns in sequential data effectively.

DECISION TREE

A popular machine learning approach for classification and regression applications is the decision tree classifier. This supervised learning algorithm divides the training data into subsets according to the feature values in a recursive manner. It creates a structure like a tree, with nodes standing in for features, branches for choices, and leaves for numerical values or class labels.

Abusive comment identification process

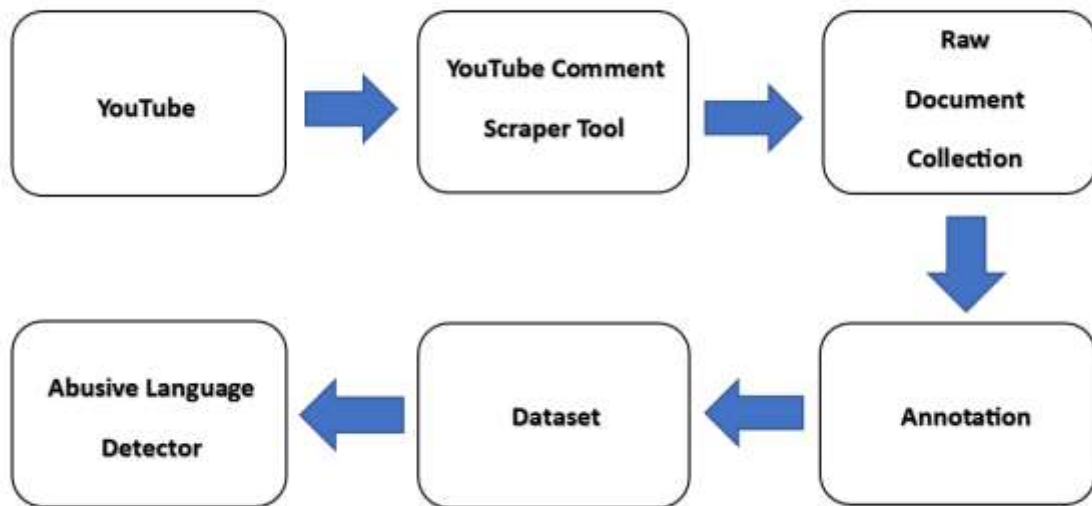


Figure 1: Abusive comment identification process

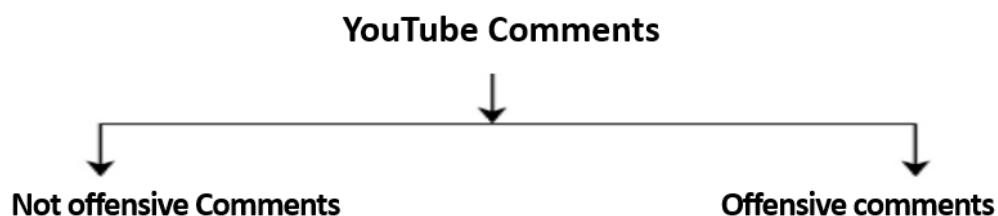


Figure 2: Comment Hierarchy for Binary classification

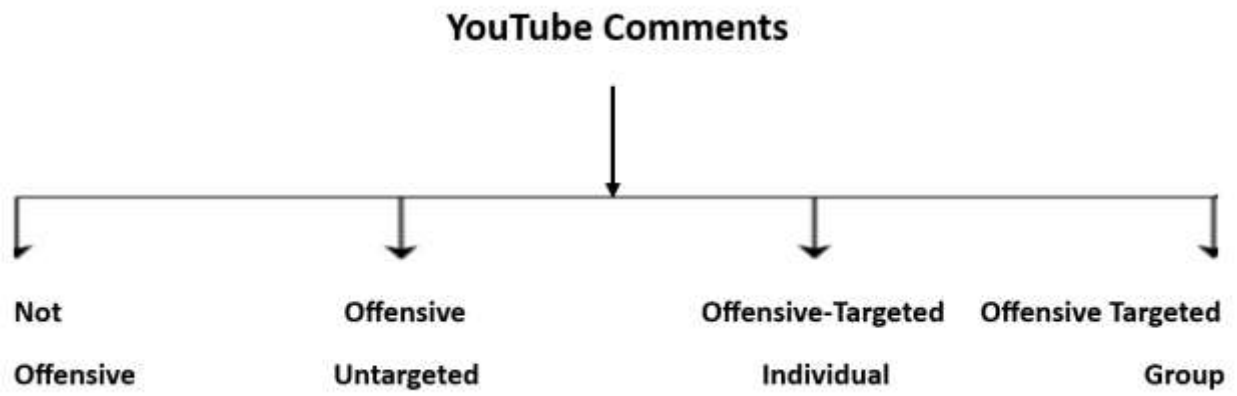


Figure 3: Comment Hierarchy for fine-grained classification

2.2 EXPERIMENTAL ANALYSIS

Partial dataset was available online, the remaining dataset was specifically created for this project. All the dataset were integrated into a single dataset. Entire dataset was annotated into five classes 'Not_offensive', 'Offensive_Untargetede', 'Offensive_Targeted_Insult_Indivitual', 'Offensive_Targeted_Insult_Group' and 'not_malayalam'. Initial dataset was imbalanced due to high number of not offensive classes and less number of offensive classes. So SMOTE was tried.

SMOTE (Synthetic Minority Over-sampling technique):

SMOTE is a method used in machine learning for addressing class imbalance in datasets. Our dataset was imbalanced due to high number of not offensive comments and very low number of offensive samples.

This method was not enough to rectify our dataset imbalance.

Scrapping:

Our initial database contained very high amount of not offensive comments and very low amount of offensive comments. It was necessary to have a significant amount of offensive comments as well. YouTube video comment scrapping was done for creating more required dataset specifically for our project.

Annotation:

Label Studio was used for annotation. In the Label Studio, Text Classification method in Natural Language Processing (NLP) Labelling Setup was used for our annotation process. Text sentiment choices were choosen as 'Not_offensive', 'Offensive_Untargetede', 'Offensive_Targeted_Insult_Indivitual', 'Offensive_Targeted_Insult_Group' and 'not_malayalam'. Comments were annotated one of these five choices one by one and new dataset was downloaded as csv file in the end. Comments that were not either in Malayalam or code-mixed Malayalam-English language was annotated "not_malayalam".

Aa chendakkarute kaariyathil oru Theerumaanamaayii 🤔🤔 Paavagal chendayum kolumaayii eni police station kayari iraggendi varumo aavo 🤔🤔🤔

Choose text sentiment

☒ Not_offensive^[1]
☐ Offensive_Untargeted^[2]
☐ Offensive_Targeted_Insult_Group^[3]

☐ Offensive_Targeted_Insult_Individual^[4]
☐ not_malayalam^[5]

Figure 4: Annotation of scraped comments using Label Studio

New annotated data set was then integrated with the previous dataset which is already trimmed. We trimmed the original dataset for reducing significant amount of not offensive comments so that the dataset imbalance can be rectified.

Data Preprocessing:

Our dataset contained comments that were annotated 'not_malayalam'. We needed to remove those comments. For this we replaced those comments with null values and then removed those null values. We used Label Encoding for converting categorical data into numerical format i.e, convert not offensive and offensive to numerical format

Splitting the dataset for machine learning:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Figure 5: Splitting into training and testing sets

We need to split our dataset into training and testing sets. Accurate predictions can be done on new and unseen dataset by proper data splitting which help access generalization performance of machine learning model.

The process of splitting data involves specifying various parameters. Typically, 'X' comprises the independent variables or features utilized by the model for predictions, while 'Y' holds the corresponding target values or labels the model aims to predict. The random state parameter plays a crucial role in reproducibility, as it establishes a seed for the random number generator, ensuring consistent random splits when the code is rerun with the same seed, set to 42 in this instance. The test size parameter determines the proportion allocated to the testing set; here, 20% of the data serves for testing, with the remaining 80% utilized for training.

BERT Algorithm:

Here we use 'bert-base-multilingual-cased'

```
tokenizer = BertTokenizer.from_pretrained('bert-base-multilingual-cased')
model = BertModel.from_pretrained('bert-base-multilingual-cased')
```

Figure 6: Tokenizing

This code instantiates a tokenizer for BERT from the Hugging Face model hub. This tokenizer will preprocess the text data into tokens suitable for BERT input.

```
for text in X_train:
    tokens = tokenizer(text, padding=True, truncation=True, return_tensors='pt')
    with torch.no_grad():
        model_output = model(**tokens)
    embeddings = model_output['last_hidden_state'].mean(dim=-1).squeeze().numpy()
    X_train_embeddings.append(embeddings)
```

Figure 7: Tokenizing X_train

Here, the loop iterates through X_train considering it as a list of text samples. For each text in X_train, the tokenizer encodes it into tokens suitable for BERT. It performs padding and truncation to ensure consistent input length and returns the tokens in Pytorch tensors.

```

for text in X_test:
    tokens = tokenizer(text, padding=True, truncation=True, return_tensors='pt')
    with torch.no_grad():
        model_output = model(**tokens)
    embeddings = model_output['last_hidden_state'].mean(dim=1).squeeze().numpy()
    X_test_embeddings.append(embeddings)

```

Figure 8: Tokenizing X_test

Similar for X_test happens in this code.

Abusive Comment Identification:

Abusive comment identification comprises two phases. In the first phase we need to train a model to classify the comments into not offensive and offensive comments. We employed some models mainly Linear SVC, Logistic regression and Decision tree. After testing the models, we selected Decision tree as it provided more accuracy. It was 66.84%.

In the next phase, first we replaced 'not offensive' comments with null values and removed them. This was done to get only offensive comments so that we can further classify the offensive comments into sub classes - 'offensive untargeted', 'offensive targeting individual' and 'offensive targeting group'. Label encoding was done to convert labels into numerical values. We employed models like LSTM and Decision tree classifier for this.

LSTM showed less accuracy of 49.02% and 'offensive targeting group' comments were not predicted.

So we implemented Decision tree classification which predicted all the classes. It showed an accuracy of 60.78%.

```

from sklearn.tree import DecisionTreeClassifier
d_classifier = DecisionTreeClassifier(random_state=42)
d_classifier.fit(X_train_embeddings, y_train)

```

```

y_pred = d_classifier.predict(X_test_embeddings)

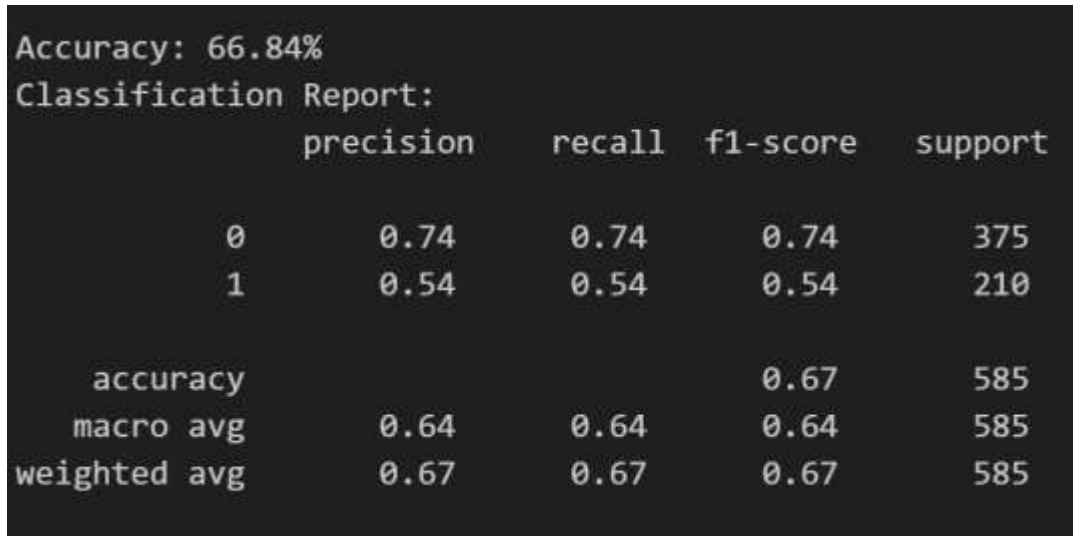
```

Figure 9: Decision Tree Classifier

CHAPTER 3

RESULTS AND INSIGHTS

The classification report obtained for first phase (Binary Classification of comments) is given below:



```
Accuracy: 66.84%
Classification Report:

```

	precision	recall	f1-score	support
0	0.74	0.74	0.74	375
1	0.54	0.54	0.54	210
accuracy			0.67	585
macro avg	0.64	0.64	0.64	585
weighted avg	0.67	0.67	0.67	585

Figure 10: Classification report for first phase

From classification report, we can see that the model predicts the not offensive comments with a precision of 0.74 and this is maximum precision value in the report. This means that most of the not offensive comments are predicted accurately. Precision for offensive comments is 0.54. Moreover, recall value and F1-score of the emotion are also same as precision for both.

Accuracy of the model is calculated by the given formula:

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FN} + \text{FP})$$

TP indicates True Positive

TN indicates True Negative

FN indicates False Negative

FP indicates False Positive

Here, the accuracy obtained is 66.84%.

	Actual_Labels	Predicted_Labels
0	offensive	not offensive
1	not offensive	not offensive
2	offensive	offensive
3	offensive	not offensive
4	offensive	offensive
5	not offensive	not offensive
6	offensive	not offensive
7	not offensive	not offensive
8	not offensive	offensive
9	not offensive	offensive

Figure 11: Predicted labels and Actual labels for first phase

The classification report obtained for phase two (Multi-class Classification) is given below:

Accuracy: 60.78%				
Classification Report:				
	precision	recall	f1-score	support
0	0.41	0.48	0.44	44
1	0.71	0.70	0.71	100
2	0.60	0.55	0.57	60
accuracy			0.61	204
macro avg	0.58	0.58	0.57	204
weighted avg	0.62	0.61	0.61	204

Figure 12: Classification report of second phase

From this classification report, we can see that the model predicts the ‘offensive targeting group’ comments with a precision of 0.71 and this is maximum precision value in the report. This means that most of these comments are predicted accurately.

Here, the accuracy obtained is 66.84%.

	Actual_Labels	Predicted_Labels
0	Offensive_Targeted_Insult_Individual	Offensive_Untargetede
1	Offensive_Targeted_Insult_Individual	Offensive_Targeted_Insult_Individual
2	Offensive_Targeted_Insult_Group	Offensive_Targeted_Insult_Group
3	Offensive_Targeted_Insult_Group	Offensive_Targeted_Insult_Individual
4	Offensive_Targeted_Insult_Individual	Offensive_Targeted_Insult_Individual
5	Offensive_Targeted_Insult_Individual	Offensive_Targeted_Insult_Individual
6	Offensive_Untargetede	Offensive_Untargetede
7	Offensive_Targeted_Insult_Individual	Offensive_Targeted_Insult_Group
8	Offensive_Targeted_Insult_Individual	Offensive_Targeted_Insult_Individual
9	Offensive_Targeted_Insult_Individual	Offensive_Targeted_Insult_Individual

Figure 13: Predicted labels and Actual labels for second phase

3.2 Streamlit App: Comment Classification App

We hosted the app in streamlit. A section for entering comments in Malayalam and code-mixed Malayalam-English language was provided.



The screenshot shows the 'Comment Classification App' interface. It features a title 'Comment Classification App' with a small icon to the left. Below the title is a label 'Enter comment:' followed by a text input field. The input field contains the text 'Trailer കാണാൻ വേണ്ടി urakathe kathu ninnar ലൈക് ഫ്രണ്ട്'. Below the input field is a 'Submit' button. A small hint 'Press Ctrl+Enter to apply' is visible at the bottom right of the input field.

Figure 14: Entering the comment in Malayalam or code-mixed Malayalam-English language



The screenshot shows the 'Comment Classification App' interface after the 'Submit' button has been clicked. The input field still contains the text 'Trailer കാണാൻ വേണ്ടി urakathe kathu ninnar ലൈക് ഫ്രണ്ട്'. Below the input field, the 'Submit' button is now highlighted with a red border. Below the button, the text 'The comment is not offensive' is displayed.

Figure 15: Streamlit Demo Output 1

Comment Classification App

Enter comment:

Ni arda vadha beeshani ondakan..patti..inala vannu korach cinema edthpo ni ethanda vallva kopila producer ayen thonuna..poov chathudeda mayre

Submit

The comment is offensive

The comment is offensive targeting an individual

Figure 16: Streamlit Demo Output 2

Not offensive comments are predicted 'The comment is not offensive' and offensive comments are predicted 'The comment is offensive' and further the type of offensive comment is also predicted if the comment is offensive.

3.3 CONCLUSION

Abusive content in online social media has increased in the past few years. Social media platforms like YouTube is one of the area where people started posting aggressive and offensive comments in response to videos others' post at a high rate. YouTube doesn't have permission granting procedure related to who can comment on videos which often leads to offensive and unpleasant comments. Our project aimed at creating a model that will detect the offensive comments from YouTube. Detection of abusive comments in low resource language of Malayalam and code-mixed Malayalam-English are not so popular. Our project specifically aimed at this. Our project also classifies the offensive comments at a fine-grained level whether it is untargeted or targeted towards a group or an individual. We got better result for the binary classification and fine-grained classification further helping understanding the nature of it.

As part of this project we created a more balanced dataset of comments in Malayalam and code-mixed Malayalam-English language which is multi-class annotated. As a future work there are scope for making a model that will prevent or remove abusive comment from YouTube or any other social media directly which will help maintain the guidelines of the particular platform. Also model was sometimes unable to detect abusive comments delivered in a sarcastic tone, which we hope to rectify in future works.

REFERENCES

- [1] Bhawal, S., Roy, P. K., & Kumar, A. (2022, July 4). *Hate Speech and Offensive Language Identification on Multilingual code-mixed Text using BERT*. ResearchGate.
https://www.researchgate.net/publication/361744588_Hate_Speech_and_Offensive_Language_Identification_on_Multilingual_code-mixed_Text_using_BERT
- [2] Chakravarthi, B. R., Priyadharshini, R., Banerjee, S., Jagadeeshan, M. B., Kumaresan, P. K., Ponnusamy, R., Benhur, S., & McCrae, J. P. (2023, June 1). *Detecting abusive comments at a fine-grained level in a low-resource language*. *Natural Language Processing Journal*.
<https://doi.org/10.1016/j.nlp.2023.100006>
- [3] bert-base-multilingual-cased · Hugging Face. (2023, June 1).
<https://huggingface.co/bert-base-multilingual-cased>
- [4] Label Studio Documentation — Overview of Label Studio. (n.d.).
https://labelstud.io/guide/get_started.html
- [5] *ML Handling Imbalanced Data with SMOTE and Near Miss Algorithm in Python*. (2023, January 11). *GeeksforGeeks*.
<https://www.geeksforgeeks.org/ml-handling-imbalanced-data-with-smote-and-near-miss-algorithm-in-python/>