



Hate Speech Detection from Facebook Social Media Posts and Comments in Tigrigna language

A Thesis Presented

By

Weldemariam Bahre

To

The faculty of informatics

Of

St. Mary's University

In Partial Fulfillment of the Requirements

for the Degree of Master of Science

in

Computer Science

June, 2022

ACCEPTANCE

**Hate speech detection from Facebook social media posts and comments in
Tigrigna language**

By

Weldemariam Bahre

**Accepted by the Faculty of Informatics, St. Mary's University, in partial
fulfillment of the requirements for the degree of Master of Science in
Computer Science**

Thesis Examination Committee:

Alemebante Mulu (PHD)

Internal Examiner

Sileshi Demesie (PHD)



External Examiner

Dean, Faculty of Informatics

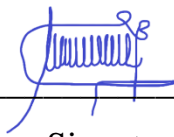
January 27, 2022

DECLARATION

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been duly acknowledged.

Weldemariam Bahre

Full Name of Student



Signature


Addis Ababa

Ethiopia

This thesis has been submitted for examination with my approval as advisor.

Dr. million Meshesha

Full Name of Advisor



Signature

Addis Ababa

Ethiopia

January 27, 2022

ACKNOWLEDGMENT

First and for most, I would like to thank the Almighty GOD for giving me strength, courage, and patience in order to accomplish this research.

Next, I would like to express my appreciation to my advisor Dr. Million Meshesha for his substantial help, persistent advice, constructive comments, suggestions and sharing his knowledge and experience throughout the preparation of this research paper.

I would like also express my great thanks to my teacher of NLP and machine learning, Dr. Michael Melese who taught us not only as a teacher but also as a friend. I would like to say thanks for Dr. Aderajew Mekonnen who support on annotating the dataset and evaluate my research paper.

Thankful to St. Mary's university community for create a chance to get good education and other necessary supports.

The last but not least, thanks to my wonderful wife and parents for their every support and advice in my life.

Table of content

ACCEPTANCE	i
DECLARATION	ii
ACKNOWLEDGMENT.....	iii
List of tables.....	vii
List of figures.....	viii
List of Abbreviations	ix
Abstract.....	x
CHAPTER ONE	1
INTRODUCTION	1
1.1 Background.....	1
1.2 Statement of the problem.....	2
1.3 Research questions.....	4
1.4 Objective of the study	5
1.4.1 General objective	5
1.4.2 Specific objectives	5
1.5 Scope and limitation of the study.....	5
1.6 Significance of the study.....	6
1.7 Methodology of the study	6
1.7.1 Research design.....	6
1.7.2 Data Preparation.....	7
1.7.3 Implementation tools.....	7
1.7.4 Evaluation procedure	9
CHAPTER TWO	10
LITERATURE REVIEW	10
2.1 Overview.....	10
2.2 Hate speech and its characteristics.....	11
2.3 Steps in hate speech detection.....	12
2.4 Machine learning algorithms	14
2.5 Supervised learning algorithms.....	15
2.6 Challenges in hate speech detection.....	18
2.7 Tigrigna language	19
2.7.1 Tigrigna Word classes.....	20

2.7.2	Challenges of Tigrigna language for hate speech detection.....	20
2.8	Related works.....	21
2.8.1	Research gap	23
CHAPTER THREE	25
METHODS AND APPROACHES.....		25
3.1	Overview.....	25
3.2	The proposed architecture.....	25
3.3	Data collection and preparation	27
3.3.1	Data source.....	28
3.4	Data preprocessing.....	29
3.4.1	Removing punctuation marks and other unnecessary characters.....	29
3.4.2	Normalization.....	30
3.4.3	Labeling Tigrigna dataset Facebook Text post	31
3.5	Feature extraction for hate speech detection.....	34
3.5.1	N-gram model	35
3.3.2.	TF*IDF vectorizer.....	36
3.6	Supervised Machine Learning	36
3.6.1	Naive Bayes	36
3.6.2	Support Vector Machine (SVM).....	38
3.6.3	Random Forest.....	41
CHAPTER FOUR.....		44
EXPERIMENTATION AND DISCUSSION.....		44
4.1	Overview.....	44
4.2	Preprocessing implementation	44
4.2.1	Removing unnecessary symbols and punctuations	44
4.2.2	Implementation of post and Comment Tokenization.....	45
4.3	Feature extraction.....	46
4.3.1	Implementation of N-gram.....	47
4.3.2	Implementation of TF*IDF feature extraction.....	47
4.4	Machine learning Models Implementations.....	48
4.4.1	Naïve Bayes	48
4.4.2	Random forest.....	49
4.4.3	Support Vectored machine.....	49
4.5	Model evaluation	50
4.5.1	Model evaluation matrix	50
4.6	Model selection and evaluation.....	59

4.6.1	Evaluating the proposed Hate speech detection Model	60
4.7	Discussion of result.....	61
CHAPTER FIVE		63
CONCLUSION, RECOMMENDATION AND FUTURE WORK		63
5.1	Overview.....	63
5.2	Conclusion	63
5.3	Recommendation	64
5.4	Future work.....	65
Reference		66
APPENDEX.....		70

List of tables

Table 1. 1: Description of the Tools and Python Packages Used During the Implementation [8].....	9
Table 2. 1: Sample alphabets of Ge'ez script for Tigrigna language [8].....	19
Table 2. 2: Common Tigrigna language punctuations [34]	19
Table 3. 1: data source on social media platform	28
Table 3. 2 : preprocessing algorithm to remove the punctuations and foreign characters	29
Table 3. 3: Normalization of Tigrigna (Geez) characters	30
Table 3. 4: Normalization algorithm for Tigrigna language.....	30
Table 3.5: annotation of post and comments for each annotator	31
Table 3. 6: Data source on social media platform.....	33
Table 3. 7: Structure of confusion matrix [8]	51
Table 3. 8: confusion matrix of hate detection model [8].....	51
Table 4. 1 : Results of the Extracted Features Vectors Size	47
Table 4. 2: Models Accuracy for Each Features and each classification model.....	54
Table 4. 3 : model precision, recall and f1-score for Each Features and each classification model.....	55
Table 4. 4 : Models ROC for Each Features and each classification model.....	57
Table 4. 5: summary of best result registered by each algorithm	59
Table 4. 6 : sample output of model evaluation using user entered data	61
Table A- 1: Tigrigna language post and comment used for user test of model	79

List of figures

Figure 2. 1 steps of hate speech detection using machine learning [39].....	13
Figure 2. 2: Most common machine learning algorithms [36]	15
Figure 3. 1 : Proposed architecture of hate speech detection for Tigrigna language [39].....	26
Figure 3. 2 Procedures for building hate speech dataset.....	27
Figure 3. 3 : distribution of class of dataset	32
Figure 3. 4 : sample data set in CSV format	34
Figure 3. 5 : Support vector machine binary classification with hyperplane [20]	41
Figure 3. 6: building random forest algorithms [22].....	42
Figure 3. 7 : Random Forest algorithm decision tree for prediction.....	43
Figure 4. 1:: python code for preprocessing Tigrigna Texts.....	45
Figure 4. 2: Sample code for tokenization sentences.....	46
Figure 4. 3 : sample code for tri-gram feature extraction	47
Figure 4. 4 : sample code for TF*IDF text Vectorization.....	48
Figure 4. 5 : Sample python code to train naive Bayes model	49
Figure 4. 6: sample python code to train random forest algorithm.....	49
Figure 4. 7 : sample code to train support vector machine model	50
Figure 4. 8 : model metric of with TF*IDF vectorizer and Naïve Bayes classifier Algorithms ..	55
Figure 4. 9 : confusion matrix of (a) Naïve Bayes Model; (b) Random Forest Model; (c) SVM Model ...	56
Figure 4. 10 : ROC curve and AUC; (a) of SVM model; (b) Naïve Bayes model; (c) Random Forest model.....	58
Figure 4. 11 : sample python code for model evaluation using test data.....	60
Figure 4. 12: sample output of model evaluation using user entered data.....	61
Figure A- 1 : sample python code for removing unnecessary text from text dataset	73
Figure A- 2 sample python code for Normalization Tigrigna Language Dataset.....	74
Figure A- 3 sample python code for Random Forest classification algorithm	75
Figure A- 4 : sample python code for Random Forest classification algorithm.....	75
Figure A- 5 : sample python code for support vector machine classification algorithm.....	76
Figure A- 6 : sample python code plot Support vector machine	76
Figure A- 7 : sample code for user test models	77

List of Abbreviations

AUC	Area under the Curve
Avg.	Average
BBC	British Broadcast corporation
CNN	Convolutional Neural Network
CSV	Comma Separated value
DNN	Deep neural Network
F1	F-Score
FN	False Positive
GRU	Gate Recurrent unit
HASOC	Hate Speech and Offensive Content
KNN	K-nearest Neighbors
LSTM	Long short –Term memory
ML	Machine learning
NB	Naïve Bayes
NLP	Natural language processing
NLTK	Natural language Tool kit
OLID	Offensive Language Identification Dataset
P	Precision
R	Recall
Regex	Regular expression
RF	Random Forest
ROC	Receiver operating characteristic
SVM	Support Vector Machine
TF	True positive
TF*IDF	Term frequency – inverse document frequency
TN	True Negative
TP	True Positive
URL	Uniform Resource Locator
VOA	Voice of America

Abstract

In recent years, hate speech on social media has become a common phenomenon in the Ethiopian online community particularly due to the substantial growth of users. As part of our country language Tigrigna language Facebook users also increased in recent years. In line with this, the hate speech in Tigrigna language is also increased. The reason could be due to, the political instabilities. Hate speech on social media has the potential to quickly disseminate through the online users that could escalate an act of violence and hate crime among peoples.

To address this problem, this research proposed hate speech detection using machine learning and text-mining feature extraction techniques to build a detection model. A hate speech data written in Tigrigna language was collected from the Facebook public page and manually labeled into hate and hate-free classes to build binary class datasets. The research employed an experimental approach to determine the best combination of the machine learning algorithm and features extraction for modeling. Support Vector Machine (SVM), Naïve Bayes (NB) and Random Forest (RF) classification algorithms are employed to construct hate speech detection model using the whole dataset with the extracted features based on word unigram, bigram, trigram, as well as combined n-grams and TF*IDF.

An experimental result shows that the Naïve Bayes classification algorithm with TF*DF feature extraction were achieved slightly better performance than the SVM and RF models for hate speech detection with 79% accuracy. In this study we achieved a promising result for designing hate speech detection for Tigrigna language. Since there is no data set available for experimentation, we used limited data for constructing an optimal hate speech detection model using machine learning classification algorithm. Hence, we recommend the need to prepare standard corpus for hate speech detection in local languages, including Tigrigna language.

Key Words: *Tigrigna Hate Speech Detection, Facebook Posts and Comments, Machine Learning Classifier*

CHAPTER ONE

INTRODUCTION

1.1 Background

In recent years, the number of social media users has increased dramatically, and social media communication has grown exponentially [1] [2]. The proliferation of social media has created an environment in which people around the world can easily communicate, exchange information, and do business. While social media has many benefits, it also has many problems [1] [2]. Because, communication on social media, does not have international law and regulation like other mainstream media and requires very little cost and time to transmit information throughout the world. As a result, unethical and unprofessional journalists, individuals with their anti-people agendas, hate speech and misinformation are distributed on social media.

Hate speech is defined by the Cambridge Dictionary [3] as, "public speech that expresses hate or encourages violence towards a person or group based on something such as race, religion, sex, or sexual orientation". Moreover, hate speech is "usually thought to be included communications of disparagement of an individual or a group on account of a group characteristic such as race, color, national origin, sex, disability, religion, or sexual orientation" [4]. According to the definition, hate speech violates the right of individuals and groups to live in freedom and equality as well as hate speech can also lead to disruption and instability and even moral decay.

Hate speech distributed using different social medias has triggered real conflicts, violence and reduce the democratic exercise [1][5]. Generally, many political activists use social media as a tool to spread and provoke their personal interest through hate speech against other people and groups. Such problem of hate speech spread cannot address using traditional norms and rules because those who spread hate speech is from other location or continent or they may be hide their location and identities. A study conducted on young people in Europe [5] showed that, they have developed a tendency to use hate speech in their speech after they join social media. In addition, the study revealed that, many youths who used the social media increased the habits of use hate speeches in their daily speech particularly against their rebellious and emotional nature have been increased [5]. Similarly hate speech on social media have been created a number of problems in Ethiopia, to call for violence, to aggravate conflicts and to defame one ethnic to other. Therefore, the research

attempts to investigate this problem using machine learning algorithms, so as to automatically detect hate speeches on social media mainly on Facebook.

Hate speech detection on social media is difficult because of the dynamic behaviors of human language, production of huge amount of data every day, the inclusion of paralinguistic signals and symbols on social media posts. In addition, posts and comments contain plenty of poorly written text and most of the time hate speech was presented content nature and post was ungrammatical text [6][1].

A study conducted by Zewdu and Jenq [7] was proposed the application of Apache spark in hate speech detection to reduce the challenges. Authors developed an Apache spark-based model to classify Facebook posts and comments into hate and not hate [7]. In the last few years social media platforms increase dramatically, as well as hate speeches and conflict trigger post and comments distrusted using different actors on social media have been also dramatically increased. The online hate speech causes different offline consequence, like triggers and aggravates real violence, rebellion and caused erosion to democracy, justice, peace building and public trusts. To handle such large volumes of hate speech data spread through online using social media, machine learning model trained with prepared labeled dataset as hate and hate free to detect post and comments automatically as hate and hate free posts in social media is essential [8]. However, detection of hate speech from posts and comments using machine learning is a challenging task because of most of posts and comments in social media has been written with poor grammatical and poor written style as well as lack of consensus on what constitutes as hate speech. Therefore, different studies are conducted to address challenges faced during hate speech detection. To mitigate hate speech detection challenges, this study considers different techniques and attempt to detect hate speech by building labeled dataset in Tigrigna language.

1.2 Statement of the problem

Nowadays, there are many studies conducted on hate speech detection in local languages and foreign languages. However, since Natural Language Processing (NLP) applications are language dependent, as per our knowledge no study has been conducted to Detect hate speeches that have spread on social media in Tigrigna language. Currently, hate speech posts and comments in Tigrigna language are dramatically increased because of the ethnic and

political tensions in Ethiopia, especially in Tigrigna speakers. As a result, hate speeches distributed through social media have big role in exacerbating the existing problems.

The increment of social media users leads to increase the hate speech distributed on social media [1]; because social media actors have the opportunity of posting anonymously malicious messages online and easily distribute information across the world in short period of time. Therefore, social media platforms have the same opportunity to spread hate speech in Tigrigna language. Hate-speech causes for develop different behavioral problems, to introduce new hate-speech words and trigger violence. The Ethiopian government has repeatedly block Internet partially and/or fully, fearing that hate-speech spread on social media could aggravate crisis when some part of the country created security crisis. So, there is no doubt that hate speeches distributed through social media especially Facebook is a big problem in our country. From the massive amount of data distributed on social media, it is very difficult or impossible to identify hate speech in a traditional way.

Social media platforms are often misused to spread contents that can be degrading, abusive, or harmful to people. Some legal and academic literature generally defines hate speech as speech that expresses hate against a person or a group of people because of a characteristic they share, or a group to which they belong hate speech as speech which either promotes acts of violence or creates an environment of prejudice that may finally cause the real violent acts against a group of people [7]. Hate speech distributed online have bad impact not only triggering real violence but also it plays a major role in creating unethical and abusive future generation [5].

Over the past three years, Tigrinya-speaker social media users have increased, in addition to newcomers, Facebook pages which created with other language have also moved to Tigrinya language. In addition, a user of the legal mainstream media that was shut down due to political controversy has been completely transferred to social media. For this and other reasons, social media is using as the main means of communication, with the exception of the federal government media.

Hate speech spread on social media is a challenging issue for Ethiopia. Because Ethiopia is a country that prone to political differences, ethnic conflicts and other problems as well as the speeches of unscrupulous political actors have been widely circulated on social media. Lack of awareness society that can justify hate speech spread on social media and believing all information

distributed on social media platforms as truth by social media users. Few researches were conducted focusing on Amharic language to identify hate speech. The study conducted by Surafel [1] attempted to design a model for automated Amharic hate speech posts and comments detection using recurrent neural network. Also, Zewdie and Jenq [7] come up with social network hate speech detection for Amharic language. As to the researcher knowledge there is no study conducted for hate speech detection from social media posts and comments in Tigrigna language.

Tigrigna language has been spoken in East Africa in the northern part of Ethiopia, specifically in Tigray region as well as in Eritrea has been used as national language. Tribal conflicts, political differences and the resulting political instability, as well as the conflict between Eritrea and Tigray, have fueled the spread of hate speech in the Tigrigna language. Unlike other machine learning application, Natural language applications are language dependent and the appropriate algorithms may differ from language to language. Therefore, the aim of this study is to apply machine learning Algorithms for hate speech detection from social media posts and comments written in Tigrigna language.

1.3 Research questions

To explore the problem and fill the gap, this study aimed to answer the following research questions.

1. What are the Corpus Annotation Guidelines that should be used to annotate (labeled) the collected data, as hate or hate-free from Facebook posts and comments in Tigrigna language?
2. What are the appropriate machine learning classification algorithms used for detecting hate speech in Tigrigna language?
3. To what extent the develop classification model works to detect hate speech in Tigrigna language?

1.4 Objective of the study

1.4.1 General objective

The main objective of this study is to develop a machine learning classification model that can detect hate speech from posts and comments communicated in Tigrigna language via Facebook social media.

1.4.2 Specific objectives

- To build the datasets using Tigrigna language posts and comments published via Facebook.
- To develop guidelines for labeling posts and comments in Tigrigna language
- To identify effective features and machine learning algorithms for hate speech detection.
- To construct hate speech detection model using machine learning algorithms
- To evaluate the performance of the constructed model using test datasets

1.5 Scope and limitation of the study

The study focuses only on detecting hate speech from Tigrigna language text using machine learning. Dataset was prepared by collecting Tigrigna text posts and comment from Facebook popular public pages from December 2019 to April2021 and annotating posts and comments into two different classes, which are hate and hate-free speeches. The study implements machine learning classifiers for the detection model and evaluates the result of each classifier based on the classification accuracy.

This study comes across different limitations during the research process. Since there is lack of other studies in hate speech detection for the Tigrigna language, we are unable to compare the progress made in the current research. Also, there is lack of standard dataset for developing hate speech detection model. As a result, this study creates a new but limited dataset. The other constraint of this study is listed as follows.

- Due to the limitation of resources required and lack of guidelines to consult, the annotation, process of the dataset was challenging and takes much time.
- Due to the lack of standard Tigrigna language stemmer and stop-words lists, the study did not apply these two preprocessing methods during data set preparation.
- This study used a supervised learning algorithm with a text mining feature extraction method to build hate speech detection model using binary class classification. However, hate speech

expresses with categories; ethnic, race color, national origin, sex, disability, religion, or sexual orientation which are beyond the scope of current study.

1.6 Significance of the study

The significance of this study is many folds. It can be used as a controlling mechanism for social media platforms to detect hate speech in their day-to-day activities. Social media platform owners benefit by taking this work results as input for developing better hate speech detection or for built hate speech monitoring models for the Tigrigna language on their platform, because they are struggling for developing hate speech detection or hate content moderation system that can support most language used on the social media platform. On the other hand, it helps social media user to be protected from hate speech during the time they spent on social media platform and from its' consequences. In addition, it also serves as a baseline and motivate researchers to work in Tigrigna language so as to improve solutions with better dataset and well-prepared annotation criteria.

1.7 Methodology of the study

The purpose of the research is to build automatic hate speech detection using machine learning in Tigrigna language. Tigrigna language has been spoken in two countries in northern Ethiopian, Tigray region and in Eritrean, used as national language in last few years; hate speeches distributed in social media using Tigrigna language have been increased because there are some political instabilities and ethnic clashes in these regions. Nowadays, social media is used as big tool in order to distribute political propaganda for all language particularly in Tigrigna language. Every day, hate speeches and malicious messages are posted in the social media platforms. Hence, to alleviate this problem, this study was focused on detecting hate speeches spread in social media in Tigrigna languages. To this end, the following methods and procedures were utilized.

1.7.1 Research design

In this study experimental research methodology is followed. Experimental research is a scientific approach to research, where one or more independent variables are manipulated and applied to one or more dependent variables to measure their effect [41]. The effect of the independent variables on the dependent variables is usually observed and recorded over some time, to aid researchers in drawing a reasonable conclusion regarding the relationship between these two variable types. In

the study three major tasks performed; hate speech data preparation, selecting implementation tools and finally evaluating the proposed hate speech detection model to measure its performance.

1.7.2 Data Preparation

Preparing the dataset for experimentation is the primary task for constructing hate speech detection model using supervised machine learning algorithm. Since there are no previously prepared datasets for Tigrigna language, an attempt is made in this study to collect, preprocess and prepare the required data set. To this end, first, the researcher selected the main Tigrigna language Facebook pages which have higher follower. Mainly, Facebook pages owned by Tigrigna speakers were used as data source to build datasets.

Preprocessing of a data includes Normalization of Geez script characters, removing punctuations and removing foreign characters and other symbols were done. The process of avoiding any unnecessary symbols and characters, like foreign language text and characters (other than Geez characters) and Tigrigna language punctuation were done, to increase the dataset quality. Normalizing of Ge'ez character is the process of transforming in to one selected character for those Ge'ez characters have same sound but different symbol, to reduce the ambiguity meaning of words in the dataset. To perform the preprocessing task the researcher uses python regular expression (regex) module to extract and remove unnecessary Tigrigna language punctuation, any foreign language characters, texts, symbols and emoji.

1.7.3 Implementation tools

The study uses several implementation tools and packages to implement the proposed solution towards, Tigrigna hate speech detection. This study uses python programming language for implementing and experimenting with each proposed solution from the data preprocessing, to the model building phase and to evaluate the implemented and proposed classifier model. In this study, Python was used because the programming language is choice for developers, researchers and data scientists who need to work in machine learning models. Table: 1.2 shows the list of implementation tools and python packages with their version and description that are used in this study.

No	Tool	Description
1	anaconda Navigator (anaconda3) 1.10.0	A distribution of different programming languages for scientific computing that it allows specifying package management and deployment. The distribution includes data-science and machine learning packages suitable for Operating system. As well as it Allows to launch development applications
2	Jupyter Notebooks 6.1.4	An open-source web application that allows us to create and share documents that contain live code, equations, visualizations, and narrative text. It was used for data cleaning and transformation, numerical simulation, statistical modeling, data visualization, and machine learning.
3	Python 3.9.0	A compile environment for python code easy to learn, powerful programming language to develop a machine learning application.
4	Sublime text 3	Easy cross-platform code editor well-known for its speed, comfort of use. It's an incredible editor right out of the box, but the real power comes from the ability to enhance its functionality using Package Control and creating custom settings.
5	Facepager 4.3	Social media content retrieval tools. Used for data collection tasks to build the dataset. This tool used to fetch posts and comment on Facebook a public page and store the data in SQLite database. Function to export the database to CSV file which is easy for manage datasets.
6	Microsoft Excel 2013	Used data preparation tasks in cleaning, filtering, sorting the collected data, and remove duplicated data Also, used to manage the annotation task.
7	Scikit-learn 0.24.2	A set of python modules for machine learning and data mining. This study uses it for feature extraction and training and testing model. The name of the package is called sklearn.
8	Pandas 1.2.4	High-performance, easy-to-use data structures, and data analysis tools. This study uses it for data reading, manipulation, writing and handling the data frame.
9	NumPy 1.20	Array processing for number, strings, and objects.

		This study uses it for handling converting the text to numeric data for features and training and testing the model.
10	Nltk 3.6.2	Build python programs to work with human language data. This study uses it for tokenization.

Table 1. 1: Description of the Tools and Python Packages Used During the Implementation [8]

1.7.4 Evaluation procedure

Performance evaluation of the model is one of the core tasks to build machine learning. Model evaluation is the process of determining how good model predicts. The evaluation process should be done using test dataset to determine the model performance.

One of the evaluation metrics used for the study is confusion matrix. Confusion matrix performance measurement is one of the appropriate evaluation metrics for machine learning, at the same for hate speech detection machine learning classification problem. Confusion matrix helps us to identify the correct prediction of a model for different individual classes as well as errors. Metrics of confusion matrix includes accuracy, precision, recall and F-score.

Accuracy: - is the ratio of number of correct predictions to the total number. It shows the level of prediction of an element of data.

Precision: -is the number of correct positive results divided by the number of positive results predicted by the classifier. It was used as a measure for the false positives of a model. In certain situations, a False Positive was considered of greater risk than a false negative [32].

Recall: is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive). Also, it was used as a measure of the number of False Negatives. This is opposite of Precision. It is mostly useful in cases when the false negative is a more detrimental error than the false positive [32].

CHAPTER TWO

LITERATURE REVIEW

2.1 Overview

Although, the emergence of social media has become a major means of communication, it has been creating an opportunity to spread hate speeches. Because, hate speeches distributed on social media can't control easily, because of the social media users have the possibility to hide their identity.

In the last few years social media platform increases dramatically and social media user also increased. This leads to increase data volume produce by social media, as well as haters that posts regularly hate speeches and comments. Online distributed hate speeches cause different offline consequences, like trigger and aggravate real violence and rebellion actions as well as caused erosion to democracy, justice, peace building and public trusts. Therefore, automatic hate speech detection and offensive behavior spread on social media especially on Facebook are gaining a lot of attention [1] [7] [9] [10].

Currently, Facebook has over 3.96 billion users in the world and more than 21.14 million users in Ethiopia [1]. The users are increasing from time to time since social media platforms supports any language in the world and, anyone can spread message on social media with any language from anywhere in the world if he/she is connected to the internet. Therefore, social media is used as a means of communication that allows to join and to transfer message from corner to corner of the world with less cost and moderate quality.

Facebook defines hate speeches uses as a rule for detect hate speeches “content that directly attacks the people based on their protected characteristics such as religious affiliation, sexual orientation, caste, sex, race, ethnicity, national origin, gender identity, and serious disease or disability.” Generally, Facebook defines an attack as dehumanizing or violent speech, calls for exclusion statements of inferiority or segregation [1]. However, they didn't strictly work and applied on it and the effect of hate speech is beyond the definition [5]. Twitter does not have any detection mechanism for hate speech post and comments. Most of the time Twitter has been blocking hate

speech posts and /or acts only upon reports or requests; otherwise. It does not have any kind of its' own filtration mechanism and it does not take responsibility for hateful tweets [5].

2.2 Hate speech and its characteristics

Hate speech is a complex concept and can't define with words only. It includes all discriminations and hatred based on intolerance. There is no internationally accepted definition or/and understanding of it. Politicians, public officials, even influential persons and many of us have directly targeted and commented using hate speech while we posting out concepts, articles or videos on Facebook.

To curb the problem of hate speech, Ethiopian has hated speech and misinformation prevention and suppression Proclamation under the Proclamation No. 1185/2020, which stated "Hate speech is the speech that intentionally promotes discrimination, hatred, or attack against a discernable group of identity or person, based on race, ethnicity, gender, religion or disability".

Definition of hate speech is also outlined by the Council of Europe's Committee of Ministers' Recommendation 97(20) as follows: "Hate speech as cover all forms of expression which spread inside promote or justify racial hatred xenophobia anti-Semitism are other forms of hatred based on intolerance including entire and expressed by aggressive nationalism and ethnocentrism discrimination and hostility against minorities migrants and people of immigrant origin but there are many other forms of discrimination and prejudices". However, many social media sites, such as Facebook, YouTube and Google are providing their own understanding of hate speech.

In general, hate speech defined as the use of language to insult, defame or arouse hatred towards a person or a group of people. It is a tool of spreading biases and discrimination based on such features as: race, ethnicity, nationality, sex, psychosexual orientation, world view, etc. It is directed against groups of a special type belonging to which, essentially, is not chosen. Mostly groups participation is determined biologically (ethnicity, sex, and skin-color, sexual preferences, etc.) or socially (language, citizenship, religion, etc.) [11][5]. Hate speech often seems to be targeted at individuals, but it is also based on the group that represents individuals. Therefore, it may lead to the so-called hate crime which consists in physical violence resulting from anti-social prejudices against a discriminated social group. Hate speech expressed using hateful messages and languages which is not neutral, highly emotion and loading with meanings.

Hate speech can be categorized in different levels [11]. Hate speech that attack subjects with the use of natural language swearwords and insults, Negative labeling, addressing individuals offensively with the use of words that define belonging to a more or less socially discriminated group, dangerous speech, more or less direct incitement to violence and crime and Hate narrations, which are stories about the world divided into the in-group and the out-group.

Because of hate speech is a negative stereotype leading to think of other groups or individuals as inferior, superior different and less worthy of respect. Negative stereotype appears discriminatory institutions, structures and norms which are embedded in the fabric of society and justify and sustain unequal social relations and others. If hate speeches are not controlled and unmanaged properly, human rights abuses, further negative stereotypes are expanded throughout society, groups and create marginalized and isolated discriminations among peoples. Moreover, conflicts and division can grow and abuse or threats also increases. Finally, in the worst case of simple words begin to translate into physical abuse hate speech can lead to hate crime against human rights.

2.3 Steps in hate speech detection

Machine learning can be used to construct a model for the purpose of hate speech detection. To detect hate speech using the machine learning first it needs to collect data from appropriate source. In this research data was collected from Facebook pages, then preprocess can be done to remove punctuation, normalize Repeated Geez script characters and label the dataset according to the category needed for classification. In this research hate and hate-free speech were labeled into binary class.

Next step, feature extraction and train the machine learning classification algorithm was done using a labeled dataset to build hate speech detection model to predict as hate and hate free speech. Machine learning model was trained with the labeled dataset, to increase the accuracy and detection capacity of the machine learning model.

After the training step, the model was evaluated using different evaluation matrixes to determine the level of prediction on hate and hate free speech. A trained machine learning model was evaluated using a test dataset then when the model encounters new data, it can detect as it is hate

or hate free speech, using what it trained using previously labeled data. Figure 2.1 show the overall steps of hate speech detection process using supposed machine learning.

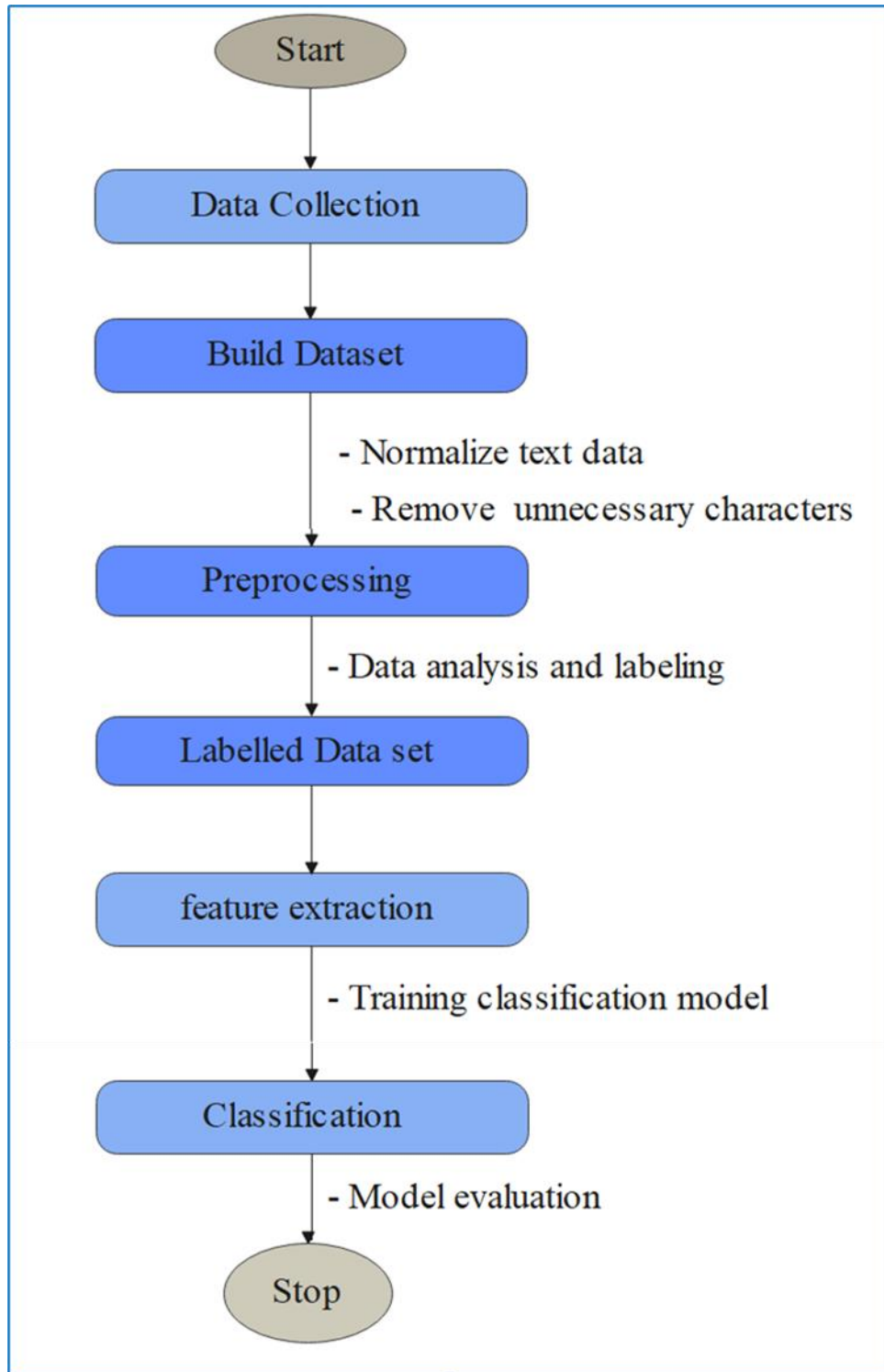


Figure 2. 1 steps of hate speech detection using machine learning [39]

2.4 Machine learning algorithms

Machine Learning (ML) is a branch of artificial intelligence domain within computer science. Machine learning uses algorithms and computational statistics to learn from data without being explicitly programmed or human being intervention. Depending on what we are trying to accomplish, there are many different ways to get a computer to learn from data [12].

Machine learning algorithms allow computers to train from data inputs and use statistical analysis to generate an output from a range of possible inputs. For this reason, machine learning helps computers build patterns from impute data, so they can build an automatic decision-making system based on the inputs received [13]. Machine learning algorithms can learn from data and improve their own capacity from previous experience. The algorithms can learn tasks like mapping input to output, identifying hidden structures in unlabeled data that is stored in the memory [4].

Machine learning, tasks are generally classified into two broader categories [13]: supervised learning and unsupervised learning. Supervised learning trains an algorithm based on input and output data labeled by humans, and unsupervised learning uses unlabeled data. It can be used for detecting patterns, discovering valuable insights, and identifying information structure [13]. As shown in figure 2.2 below, there are different machine learning algorithms under supervised and unsupervised learning.

Supervised machine learning, the algorithm is learning to recognize elements in unlabeled data or unseen data, to categorize them according to the labeled training data. As the algorithm starts making predictions, it can be constantly corrected by the programmer until the algorithm is able to achieve a high level of accuracy. The main goal of supervised learning is to make predictions about unseen, unavailable, or future data based on the available sample data [13]. The supervised Machine Learning Algorithms divided into classification and regression. In the classification stage, the system decides on the type of information it receives. It can be able to classify these data into different classes or categories, based on some predefined criteria, like "hate" or "hate-free". The regression, process the system to identifies a pattern in the information and predict continuous outputs [14].

Unsupervised machine learning involves unlabeled data and since unlabeled data are more prevalent than labeled data. The method is very robust. The goal of using unsupervised learning is to discover hidden patterns within datasets. However, it may also have the goal of setting up a learning pattern to allow the machine to discover the representations needed to classify raw data on its own. Unsupervised learning algorithms can build meaningful relationships between data that without any corrective feedback. Common algorithms of unsupervised learning are Clustering and Dimensionality Reduction algorithms.

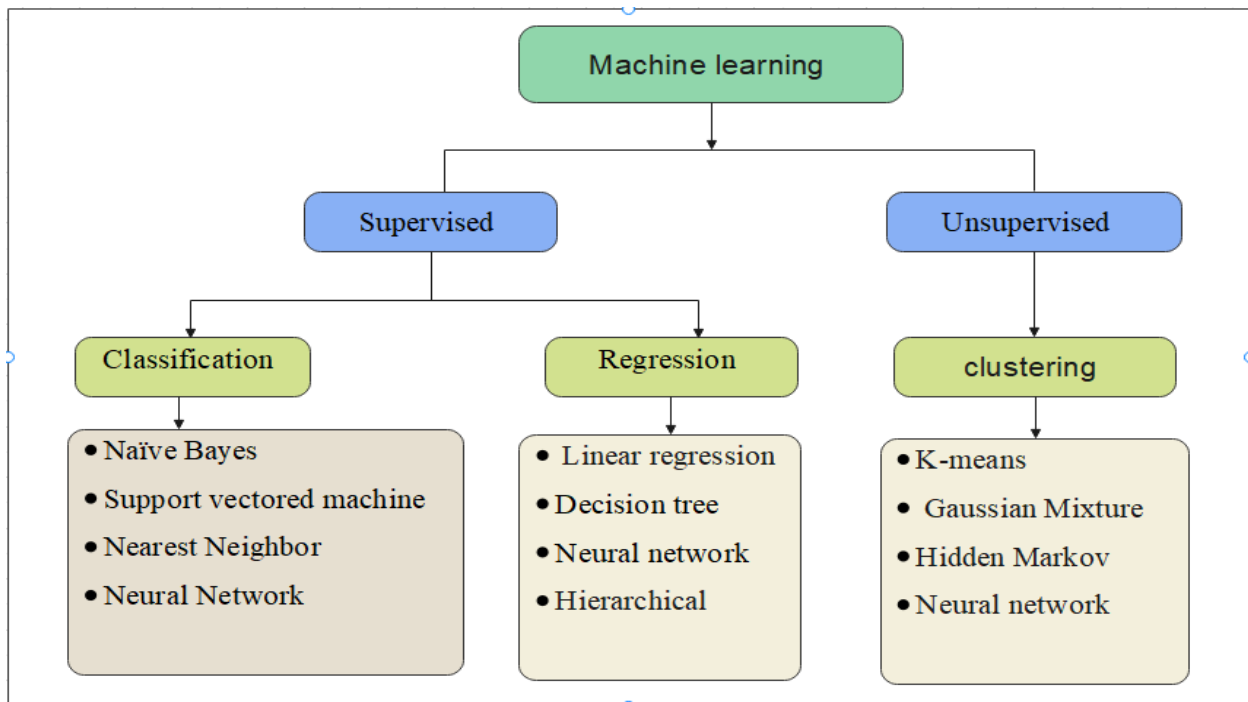


Figure 2. 2: Most common machine learning algorithms [36]

This study applies supervised learning algorithms for hate speech detection from posts and comments collected from online sources, such as Facebook.

2.5 Supervised learning algorithms

As scholars noted, there are different supervised learning algorithms available for different tasks; the common supervised algorithms are [13] [15] presented as follows.

Support vector machine (SVM): - Is a supervised machine learning model that uses as classification algorithms for two-group of classification problems. The Support vector machine (SVM) model has sets of labeled training data for each category (class) in order to categories

(predict) the new input data. The objective of the SVM algorithm is to find the hyper-planes in N-dimensional space that distinctly classify the data points. In this study, hate speech detection have two classes which are hate and hate-free data points. Therefore, two-dimensional space hyper-plane can be appropriate for this study. Hyper-planes are decision boundaries that help classify the data points.

Support Vectors are simply the co-ordinates of individual data points and it is a frontier which best segregates the hate from the Females. Therefore, SVM classification Algorithm is chosen for this research hate speech detection in Tigrigna because of it best and easy for classification binary class data points

Decision tree: Decision Tree is a supervised learning technique that can be used for both classification and Regression problems. But, mostly of the time, it is chosen for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

Decision tree has root node, branch node and leaf node. Root Node is the initial node which represents the entire sample and may get split further into further nodes. Branch Nodes represent the features of a data set and the branches represent the decision rules. Leaf Nodes represent the outcome [33].

To implement classification using decision tree, it begins with the original set ‘S’ as the root node, calculates Entropy (H) and Information gain (IG) for each attribute of set ‘S’.

Mathematically, Entropy for one attribute can be calculated using the following formula [33]:

$$E(S) = \sum_{i=1}^k -p_i \log_2 p_i \quad (2.1)$$

Where, S is the current state, and P_i is the probability of an event i of state S

Then, attribute was selected which has the smallest Entropy or Largest Information gain. The set S is then split by the selected attribute to produce a subset of the data. The algorithm continues to recur on each subset, considering only attributes never selected before.

However, researcher do not use the decision tree algorithm to identify hate speech because of decision tree algorithm it may have an over-fitting problem, which can be resolved using the Random Forest algorithm.

Naïve Bayes: The Naïve Bayes classifier is based on Bayes' theorem and classifies every value as independent of any other value. It allows us to predict a class/category, based on a given set of features, using probability. It uses for both classification and regression analysis [15].

Naïve Bayes classifier Algorithm is one of the most popular applications for text classification application it is simple and easy to implement, it doesn't require as much training data, it is highly scalable with the number of predictors and data points, and it is fast and can be used to make real-time predictions it is not sensitive to irrelevant features [37]. Therefore, Naïve Bayes algorithm the best classification for hate speech detection problem.

K-Nearest Neighbors (KNN): The KNN helps to group similar data points together according to their proximity to each other. In other words, it estimates how likely it is for a data point to be a member of a group. KNN Algorithm used for both classification and regression. [13].

K-Nearest Neighbors (KNN) predict the correct class for the test data by calculating the distance between the test data and all the training points. Then, it selects the 'K' number of points which is close to the test data. The KNN algorithm computes the probability of the test data belonging to the classes of 'K' trained data and class holds the highest probability was selected.

The KNN algorithm works by choose k value to select neighbors trained data points to new input unseen data points. There are no pre-defined statistical methods to find the most favorable value of K. But initially randomly selected values of K were used and changed according our problem. Calculate the distance between the new point and each trained point. There are many methods for calculating distance; of which the most commonly known methods are Euclidean and Manhattan distances.

$$\text{Euclidean} = \sqrt{\sum_{i=0}^k (x_i - y_i)^2} \quad (2.2)$$

$$\text{Manhattan} = \sum_{i=0}^k |x_i - y_i| \quad (2.3)$$

However, KNN algorithm accuracy depends on the quality of data, sensitive for scale of data and irrelevant features and other effects, it is not chosen for the hate and hate free speech classification problem, because of there are challenges of labeling hate and hate-free speech due to dynamic behavior human language.

2.6 Challenges in hate speech detection

There are many layers to the difficulty of automatically detecting hateful particularly in social Medias. One challenges of hate speech detection is, the expression of hate speeches on social Medias [6]. Some expressions that are not inherently hate speech [6] have hidden meaning to express hater. Hater expressions are also depending on period, culture and community awareness level [6]. Some expression was hater in the period, may not hate on current time as well as some expression hater in some cultures are not as such hater in other one. Depending on the context of the expression same words have different use (meaning).

Other challenges in detecting hate speeches are dynamic change human language, so many hater speeches are emerged every time; for example, in Ethiopian many hater speeches have been emerged during last years', like timikihitī hayilitati [ትምክሕቲ ሓይልታት] ፣ nefit'egna (ነፍጠኛ) and junta [ጃንታ] etc. The recommendation to mitigate bias in hate speech is explicitly preparing annotators for it [6]. But annotating social media comments and posts are also, other challenging issues.

There are many definitions of hate speech although they depend on different factors like: level of democracy, public awareness, and no international accepted definition for hate speech [6] [5] [1]. However, many researches revealed that, there is no an agreement that satisfy the criteria of being a universally accepted productive definition for hate speech on several countries. One reason to this may be, lack of legally binding documents between the countries [11] [5] There are many expressions that are not inherently hater, however they can be so in the right context [11]: - but when it is used to describe hate speech, it can be very difficult to annotate as hate speech.

2.7 Tigrigna language

Tigrigna is a Semitic language spoken in Eritrea and in Northern Ethiopian, Tigray Region. It is a national language in Eritrea and Northern Ethiopian Tigray region. Tigrigna has its own alphabet adopted from Ge'ez script. Like English, Tigrinya is written from left to right. The first forms of the Geez script included only consonants, while the subsequent variants of the characters represent phoneme pairs of consonant-vowel. The normal alphabet in Tigrigna is considered as to be consonant. Most of the consonant are written in seven slightly different forms of corresponding to the traditional vowels. The vowels of the language are beginning to be written by the way of small additional or modification to the consonant [38]. Ge'ez, Tigrigna writing uses characters formed by a consonant-vowel combination. Each of the seven distinct forms that reflect the seven vowel sounds are ሀ ሁ ሂ ሃ ሄ ህ ሆ. There are 33 basic alphabets (syllables) with seven vowels for each. Sample alphabets with their seven orders are presented in Table 2.1.

*	e/ä	u	i	a	ē	ə	o
h	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
l	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
hh	ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
m	መ	ሙ	ሚ	ማ	ሜ	ሞ	ሞ
r	ረ	ሩ	ሪ	ራ	ራ	ር	ሮ

Table 2. 1: Sample alphabets of Ge'ez script for Tigrigna language [8]

Tigrigna language has punctuation symbols but a few of them used in a computer system. The basic punctuations included in computer systems are shown in the table 2.2 below.

Tigrigna punctuation	Name	English
፥	Full stop / period	.
፣	Question mark	?
፣ or:	Comma	,
፥	Colon	:
፥	Semi-colon	;
፥-	preface colon	:-

Table 2. 2Common Tigrigna language punctuations [34]

Tigrigna writing scheme has some issues that are difficult to process Tigrigna text. One of those challenges is the redundancy of characters used in Tigrigna. There are some characters that have the same sound but with different symbols. Those characters are [ጸ፣ ፀ] ፣ [ሰ፣ ሠ] and [ሀ፣ ጎ], such characters create ambiguity for Natural language processing of Tigrigna language.

2.7.1 Tigrigna Word classes

Understanding of human language for the machine is very difficult. Natural Language Processing researches are looking closely at this problem and try to build systems that can understand natural languages. Word class (Part-of-speech) tagging is one attempt in the effort of understanding human languages. It is the task of a category to a word which indicates the role of the word. Therefore, for word class have big role Natural language researches to which a word belongs guides its use in a sentence and defines the correct word order and punctuation. Knowing the role that each word has in a sentence structure clearly helps to understand sentences and also to construct them properly.

Including Tigrigna language Semitic languages, in general, are characterized by rich inflectional and derivational morphology, which generates numerous variations of word forms. The inflection of verbs it is possible to extract linguistic information such as tense-aspect mood, number, gender, person, object suffix, negation and case [38].

Many Tigrigna linguists classify Tigrigna words classes into eight major categories [37] [38]. These are verbs, nouns (nominal), pronouns, adjectives, adverbs, conjunctions, prepositions and interjections. There are many word class sub categories of the main categories like Perfective, Imperfective, auxiliary and others.

2.7.2 Challenges of Tigrigna language for hate speech detection

Hate speech detection is challenging for any language but it is more difficult for a language that doesn't have more researches and low grade or absence of electronic data support to make NLP research. For Tigrigna language there is no prior research on hate speech detection and previously prepared dataset for training model. Other difficulties are differing definitions on what constitutes hate speech, limitations of data availability for training and testing of these systems.

Another challenge is, like other Semitic Ethiopian language, Tigrigna language has rich inflectional and derivational morphology, which generates numerous variations of word forms. Therefore, it is very difficult to make Tigrigna morphology analysis and change to root word (STEM).

Collecting and annotating data for the training of automatic classifiers to detect hate speech is challenging. Specifically, identifying and agreeing whether specific text is hate speech is difficult, as per previously mentioned, there is no universal definition of hate speech. Some hate speeches may express with hate free words but have indirectly transfer hater message, that difficult to annotate as hate speech and leads for false prediction.

2.8 Related works

This section shows a comprehensive review of basic related works to the area of automatic hate speech detection on social media. Furthermore, this topic to revealed, clearly understand on the general techniques, methods, and results of existing studies and to identify the knowledge gap. Many foreign and local researches conducted on hate speech detection were assessed.

Challenge of hate speech detection in social media was conducted by György, Pedro, and Rajkumar [6]. Despite detection of hate speech posts on social media using machine learning is very challenging work, this study aimed to resolve basic automatic hate speech detections challenges using different techniques. For hate speech posts challenges have context – dependent nature by building large amount of labeled dataset and make annotation for ambiguous hate speech words as well have different senses. Moreover, for poor text and or ungrammatical texts using deep neural network (DNN) which can solve the grammatical errors. Therefore, the main purpose of this research was to create a compression different model and techniques to obtain compatible and suitable good hate speech and hate-free identifier model. It can be used different available corpus as data source rather than make analysis and preprocessing the data found from corpus, new data was not collected by the researcher. The corpus used as data source was HASOC dataset and OLID dataset. Text processing was the first task for automatic hate speech detection to removing same unnecessary text, removing same extra space from document, remove same special character like [@, #, URL] and others unnecessary words characters. The researchers mainly focus to solve challenges of hate speech detection by using different models and data sets. Experiments

was done using machine learning RoBERTa for feature extraction and Fast Text Algorithm for text classification with HASOC 2019 dataset only and with additional HateBase dataset. An experiment using deep learning was done using combination Convolutional neural network and Long Short-Term Memory (LSTM) layers (CNN-LSTM) with single dataset with more data sets. The experiment shows that the higher datasets amount and deep learning have better solution to resolve challenges of hate speech detection on social media.

Zewdie and Jenq [7], as well as Surafel and Kula [1] are the recent related works on Amharic language and many research works have been done on foreign language. However still there is no research paper work on Tigrigna language to detect hate speech.

Zewdie and Jenq [7] performed a study on hate speech detection for Amharic language. The researcher aimed to develop machine learning application for automatically detecting hate speech spread on social media posts and comments. They created a dataset with 6,120 comments and posts from Facebook, using facepacer API and manually to collect data from different social media pages. After preprocess and normalized the collected data was labeled and categorized as hate and hate-free. For feature selection TF-IDF and word2vec algorithms was and Naïve Bayes and random forest algorithms for text classification was used. Experiments were done with combination of the above selected feature selection and text classification algorithms got the best output Naive Bayes classification algorithm and with word2vec feature model 79.83%, 83.05% and 85.34% accuracy, ROC score and area under Precision and Recall respectively. The nature of hate speech makes challenging to annotate hate speech, one challenging with nature of hate speech posted on Facebook was, hate speech can expression without using hater words and some hate speech comments and posts have been written in the ungrammatical way and poor writing styles. Such thing makes difficult to make annotation of hate and hate-free speech.

However, the above researchers used three manually annotators but they didn't show how they annotate the dataset. They used hate words as indicator of hate speeches but some hate speeches can express without hater words but used indirectly transfer hater message. Therefore, annotating using hate words only didn't included all content of hate speech expressions. In this study we prepared annotation guideline to included even the hate speeches expressed without hater words by considering the overall meaning of the message of post and comments. NLP application

language dependent, hate model build for Amharic language didn't work for Tigrigna language so, there is research gape on determining appropriate hate detection model for Tigrigna language.

Surafel and Kula [1] also further conducted a study on automatic hate speech detection from social media posts and comments using recurrent neural network. Spreading hate speech on social media is a challenging issue now many malicious speeches distributed on social media is trigger violence and has many consequences. The purpose of the study is, to build a hate speech detection model using the recurrent neural network model. Researchers collected data from Facebook pages, which have many followers and collected 30,000 comments and posts data they make preprocess and normalize for extract unnecessary characters, symbols, emoji and punctuations and make normalize the data to label the repetitive Amharic characters. Those researchers make annotation (labeling) manually hate and hate-free. For detecting hate speech on social media out of 30,000 data of dataset 15,949 labeled as hater and the reset as hate-free. They used n-gram and word2vec algorithms for feature extraction process and Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) algorithms for text classification. Based on the data set 80% data used for training, 10% validate set and 10% for test set. The experiment performed using this dataset with different parameters on GRU and LSTM based RNN model by feature representation of word2vec resulted in better test accuracy of 97.9% by RNN-LSTM. They were relatively used a better amount of data set and acquired good accuracy using recurrent neural network technique.

2.8.1 Research gap

A different study was conducted on the problem of hate speech detection, and they used different algorithms to detect hate speech propagated on social media and other online web platforms. most of researches used hate words to labeled pots and comments as hate and hate-free speech, indeed most of hate speeches expressed with hate words but hate speech can also express without using hater words to transfer indirectly hate speech like Mocking and labeling other without using hate speech words. Such types of hate speeches can't handle with labeling hate speeches using hate world only. Therefore, it needs to prepared hate speech annotation guideline (shown Appendix A) to label hate and hate-free speeches, not only hate speeches expressed with hate word but also for these indirectly transfer hater messages.

NLP Applications are language dependent, hate speech model build for one language may not effective for other language, it must be approved by study. Hate speech spread on social media in

Tigrigna language have been increasing in recent times because of political tension and conflicts in Ethiopia. There is also research gape on determine the appropriate feature extraction Algorithms and classification algorithms to build a better hate speech detection model for Tigrigna language. Therefore, the research focus on preparing hate speech annotation guideline and identify good algorithms used to build detection model for Tigrigna language.

CHAPTER THREE

METHODS AND APPROACHES

3.1 Overview

In this chapter, we discuss a research methodology to collect data and build a dataset. Performa some activities preparing couples, preprocessing and other techniques in order to achieve research objectives and answer the research question. The following section in this chapter explains and justifies the methodology used in conducting the study on Tigrigna language hate speech detection. Collected textual data needs preprocessing to extract unnecessary information, feature extraction and classification using different algorithms to build model and predict hate speech and hate-free speech using supervised machine learning technique.

3.2 The proposed architecture

Here under figure 3.1 shows the proposed architecture of hate speech detection on social media which is designed using EdrawMax tool. The Architecture of hate speech detection shows, first Researcher was selected randomly sample Facebook pages from both Ethiopian and Eritrea have higher number of followers as shown in table 3.1. I used FacePager Facebook API for collecting data and prepared a Facebook post and comments couples. The dataset has some unnecessary characters and symbols and others and to clean the dataset data I write sample python code to clean unnecessary data from dataset shown Appendix B-1 and Normalized dataset data to remove duplicated Ge'ez characters, i used python code to normalize shown Appendix B- 2. The nest stage researcher used a sklearn that a built-in python machine learning library for feature extraction and to build hate speech detection model.

The Facebook post and comments was collected mainly using FacePager a Forebook API application that can download posts and comments from each page in CSV file format, I also collect some data manually by select copy and paste in to CSV file.

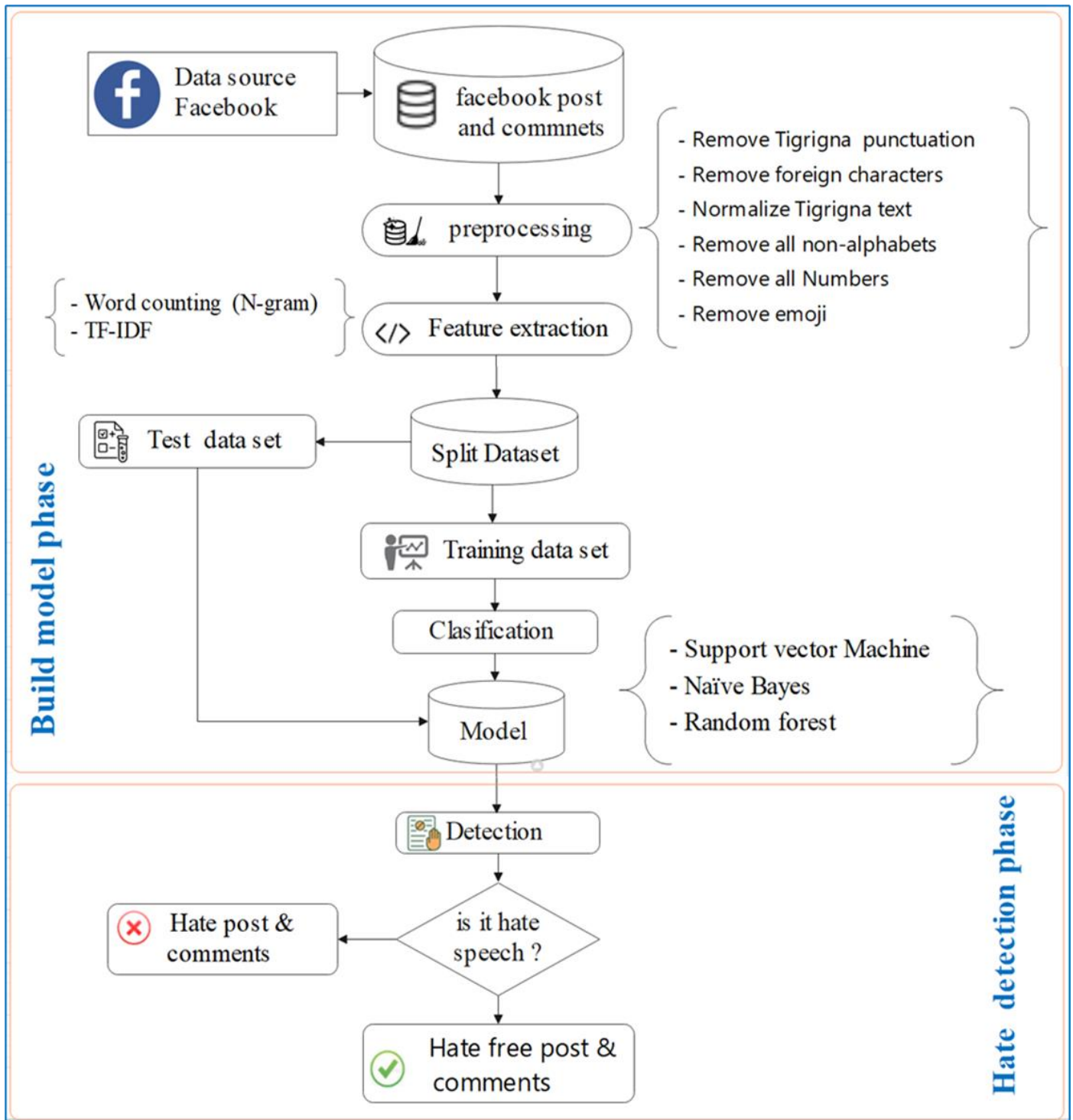


Figure 3. 1 : Proposed architecture of hate speech detection for Tigrigna language [39]

Dataset was split in to training dataset and test dataset, feature extraction is the process of converting textual data in real vector [16] using different feature extraction algorithm, the text

classification used to organize, structure, and categorize feature extracted dataset. The classification algorithms used to build a model. Test data set was used to show how the trained model performs on all of the defined test data. Test dataset was unseen data for the model (i.e., which is not part of the training dataset), It is used to check to what extent the model predicts correctly hate and hate-free posts and comments. Evaluation of the model using different evaluation matrix is an important task to see the level the performance of the model.

3.3 Data collection and preparation

Appropriate training data is necessary for machine learning to predict futures. Machine learning application without Data is just like a car without fuel. There is not prior collected data for most Ethiopian language as well for Tigrigna language. Therefore, researcher collected primary data from the Facebook pages that have many followers, using facepager and manually. Data Collected from Facebook have unnecessary chunk of words, characters and symbols, such elements were removed from dataset. If the text data was not clean from those chunks it reduces data quality and it results weak classification model. Preprocessing is the process of cleaning unnecessary chunks form dataset. The overall data collection and preprocessing architecture is shown in figure 3.2 below

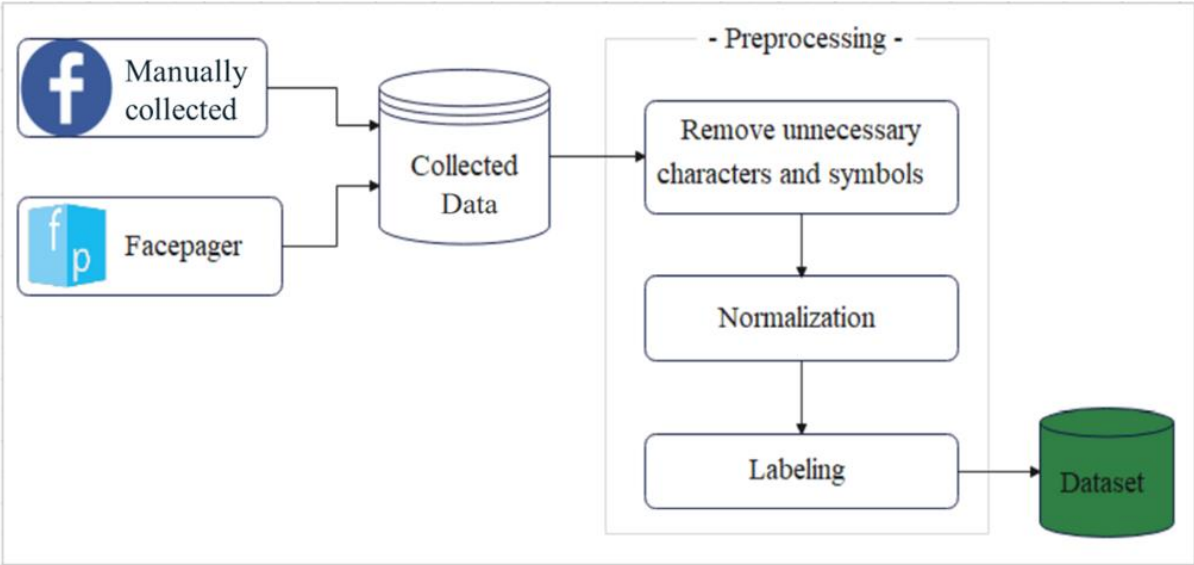


Figure 3. 2 Procedures for building hate speech dataset

3.3.1 Data source

The data source for building a Tigrigna dataset is Tigrigna language Facebook pages, which has Higher followers. However, the Tigrigna language is spoken in Ethiopia and Eritrea, as a result, the data source Facebook pages have been selected from both countries and international media as much as possible.

the data source Facebook pages are randomly selected from those higher number of followers, because of a Facebook page with higher number of followers, believed have different users with different point views as well as from well-known Tigrigna language international media. The criteria for select sample data source is randomly select from those have higher number followers from both counties. List of data source pages are listed below on table 3.1.

No.	Data source name	Type of social media	Number of followers	Owner of Facebook pages
1	voa Tigrigna	Facebook	343,213	International Media
2	BBC News Tigrinya	Facebook	250,705	International Media
3	Digital woyane	Facebook	168,319	Ethiopian
4	Kassa HaileMariam	Facebook	32,850	Ethiopian
7	Tigray media house	Facebook	28,145	Ethiopian
8	Tigray Tigray	Facebook	5,000	Ethiopian
9	Alula Solomon (Al-Solomon)	Facebook	212,844	Ethiopian
10	ዓለም ለኸ መቃልሕ ታሪኽ ተጋሩ	Facebook	111,000	Ethiopian
11	Pres. Debretsion Gebremichael	Facebook	308,032	Ethiopian
12	Sawa-ሳዋMedia	Facebook	16,085	Eritrean
14	Center for research and documentation ማእከል ምርምርን ስነ-ምግባር ማረጋገጫ ሃገራዊያን	Facebook	29,625	Eritrea
16	ERITREA	Facebook	102,559	Eritrea
17	EastAfro	Facebook	218,327	Eritrea
18	MassawaTube	Facebook	28,992	Eritrea
19	Denden Media	Facebook	85,030	Eritrea
20	MOI Eritrea	Facebook	17,542	Eritrea

Table 3. 1 data source on social media platform

3.4 Data preprocessing

Preprocessing dataset is the process of removing unnecessary chunk text, characters and symbols from dataset. Data preprocessing is important to increase the quality of the data and to improve the performance of the classification model. Therefore, the posts and comments text data collected from Facebook page was preprocessed using the following steps.

3.4.1 Removing punctuation marks and other unnecessary characters

Removing foreign words and characters (other than Geez characters), Tigrigna punctuations [፣፡።፣፥፦፧፨፩፪፫፬፭፮፯፰፱፲፳፴፵፶፷፸፹፺፻፼፽፿] and other symbols from text data is one part of preprocessing step. Researchers write regular expression and some additional python code to code extract unnecessary characters. Pseudo code used to remove unnecessary characters and texts are show below in the table 3.2

Input: text dataset with same chunks
Output: clear text
Begin
Read the text in the dataset:
While (! end of the text in a dataset):
If the text contains special_char [,'! @#\$\$%^&*....] then
Remove special_char
If the text contains symbol [⟨⟩«»=:~/_/....] then
Remove symbol
If a text contains geez_number [፩ ፪ ፫ ፬ ፭ ፮ ፯ ፰ ፱ ፲ ፳ ፴ ፵ ፶ ፷ ፸ ፹ ፺ ፻ ፼ ፽ ፿] then
Remove geez_number
If a text contains Tigrigna_Punc='፣፡።፣፥፦፧፨፩፪፫፬፭፮፯፰፱፲፳፴፵፶፷፸፹፺፻፼፽፿' then
Remove Tigrigna_Punc
If text contain Alphanumeric = [a-z A-Z] [0-9] then
Remove Alphanumeric
If a text contains emoji = [🤔👍👎😄....] then
Remove emoji
If a text contains extra white space, then
Join extra space
Return clean_text;
Main ()
Print (data.csv)
End

Table 3. 2 : Pseudocode unnecessary characters, emojis, symbols and others

3.4.2 Normalization

Normalization is the other preprocessing task for normalizing Tigrigna language characters. Geez scripts have character similar sound and meaning but different symbol, such characters create ambiguity when train machine learning model. Therefore, characters have similar sound but different symbol was represented in to one of them. The researcher wrote a simple python code to represent the characters to one appropriate character according the table 3.3 shown below

Nº	Characters	Representative character	Description
1	ሰ, ሠ	ሰ /se/	ሠ replaced by ሰ
2	ጸ, ፀ	ፀ /tse/	ጸ replaced by ፀ
3	ሀ, ኅ	ሀ /he/	ኅ replaced by ሀ

Table 3. 3: Normalization of Tigrigna (Geez) characters

A pseudocode for Normalization of Tigrigna language characters is shown in figure 3.4 below.

```

Input: un-normalized text data
Output: Normalized text
Began:
    Def function (input):
        For data in input:
            If input text contains characters [ኅ][ኆ][ኇ][ኈ][኉][ኊ][ኋ][ኌ][ኍ][኎] :
                Replace characters with [ሀ][ሁ][ሂ][ሃ][ሄ][ህ][ሆ]
            elif input text contain characters [ሠ][ሡ][ሢ][ሣ][ሤ][ሥ][ሦ][ሷ]:
                Replace characters with [ሰ][ሱ][ሲ][ሳ][ሴ][ስ][ሶ][ሷ]
            elif input text contain characters [ጸ][ጹ][ጺ][ጻ][ጼ][ጽ][ጾ]:
                Replace characters with [ፀ][ፁ][ፂ][ፃ][ፄ][ፅ][ፆ]
        Return normalized data
    For data in Normalized data:
        Save (data.csv)
End

```

Table 3. 4: Normalization algorithm for Tigrigna language

3.4.3 Labeling Tigrigna dataset Facebook Text post

Labeling Facebook post and comments as hate and hate free speeches are very challenging task [3]. There is no international criterion for labeling hate and hate-free speech, since the definition of hate speech different from nation to nation [6]. Therefore, every country and social media platform has developed their own definitions of hate speech. For labeling text Dataset Facebook post and comments requires prepared labeling guideline criteria and make careful assessments on the gridlines. Researcher used the definition of Facebook hate speech definition and Ethiopian proclamation “Hate Speech and Disinformation Prevention and Suppression Proclamation No. 1185 /2020” that define as “Hate speech means speech that deliberately promotes hatred, discrimination or attack against a person or a discernable group of identity, based on ethnicity, religion, race, gender or disability” to prepared annotating guideline shown on (Appendix A). According to the guidelines prepared, all posts and comments were annotated (labeled) in to two classes as hate or hate-free speech. According to the definition of hate speech, targets are specific groups and individuals with specific characteristics belonging to the group and individuals. The target of hate speech consider in this research is:

- Ethnicity
- Political point view
- Religious
- Gender
- Disabilities

For labeling the Facebook comments and posts, two annotators participate, those with a good understanding of the language. The number of annotations done by each annotator is described in table 3.5 below.

Annotators	Hate	Hate-free	Total
Annotator1	1,458	1,555	3,013
Annotator 2	2,158	2,622	4,780
Total	3,616	4,177	7,793

Table 3.5 annotation of post and comments for each annotator

As shown in table 4.4 there are 3, 616 instances of hate and 4, 177 instances of hate-free speeches with a total of 7, 793 instances that are used for the experiment. The below figure 3.3 depicts the distribution of data set for the two classes.

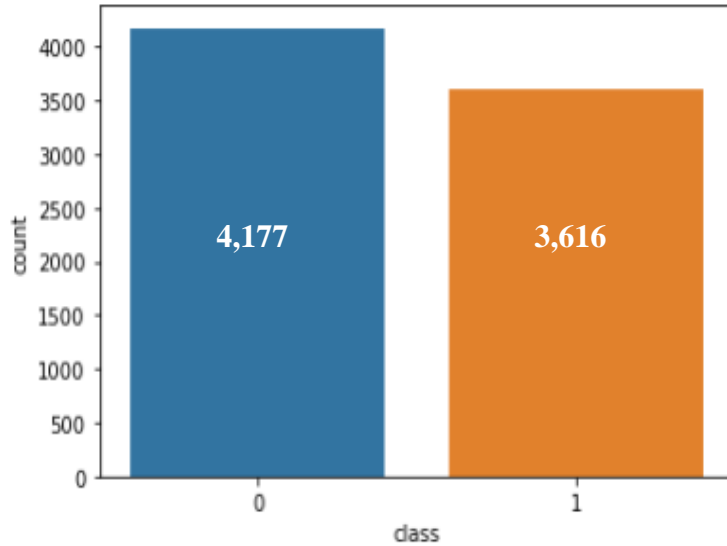


Figure 3. 3 : distribution of class of dataset

3.4.4 Data set for experimentation

Data set used for the experiment was collected from selected Facebook pages. Researcher collected using facepager and manually and used Microsoft excel for remove duplicated posts data comments. For building hate speech detection model for Tigrigna language, we used (7,793) posts and comments. Out of the collected data, 80% (6,234) used for training and the remaining 20% (1559) used as test data set.

The sample training dataset used for the experiment is shown on table 3.6 below.

ID	Post	Source	Labeling	Date of post
1	አብ ውሽጥና ዘሎ ፀገም ዉን ንቅረፍ እቲ ናይ ደገ ፀገም ንክንሓብር ዝገብር እንትትኮን እቲ ናይ ውሽጥና ግና ዝበታትን ስለ ዝኾነ ኣትኩርና ንስረሐሉ	VOA Tigrigna	Hate free	06/2/2021 time: 3:25 AM
2	ሕድሪ ጀጋኑና ኣይነዕብርን	BBC Tigrigna	Hate free	19/02/2021 time: 4:25 AM
3	ፅቡቕ ግን ትግራይ ናይ ባዕላ መብራህቲ ስልኪ ኣየርመንገዲን ባንክን ክህልዎ ክንግብር ኣለና	Dr. D/Tsion G/Michea	Hate free	12/03/2021 time: 9:25 AM
4	አብ ዘሓፀረ እዋን ዓባይ ትግራይ ንርእንኮን ግና ነዛ ኢትዮጵያ ትብሃል በላዕት ህዝቢ ትግራይ ግደፉዎ ሰላም ክንርክብ	BBC Tigrigna	Hate	19/02/2021 time: 5:25 AM
5	ሓቂ ክቡር መራሒና	Dr. D/Tsion G/Micheal	Hate free	19/02/2019 time: 5:15 AM
6	እወ ልክዕ ክቡር ዶክተር ደብሪፅ ኣንታ ጀግና ወዲ ህዝቢ ህዝቢ ትግራይ ድማ እቶም ክረግፁዎ ዝሓሰቡ ቀዲሙ ስለ ዝረገዎ ሓዚ ክቕብርኩም ጥራሕ እዩ ዘለዎ	Kassa H/Mariam	Hate	19/02/2021 time: 9:45 AM
7	ክቡር ደ.ር ነዞም ዓይንን ሰነልቦናን ዘይብሎም ሰባት ኩሉ ህዝቢ ክቃለሶም ከም ዘለዎ ትእዛዝ ይውሃብ ዝብል ሓበሬታ ይህብ ብተወሳኺ ድማ እዚ ናይ ቀይሕ ባሕሪ ዋርድያ ናብ ሓምሓም መሸረፊት መጻኢ ኣሎ እሞ ብውሕልል ኣሰራርሓ ንጠንቆቆ	D/tsion G/michea	Hate	19/02/2019 time: 5:25 AM
8	ናይ ሓባር ራኢ ካብ ዘይብሎም ኣህዛብ ነፃ ንወፀሉ መዓልቲ ቀሪቡ እዩ	VOA Tigrigna	Hate	19/04/2021 time: 5:35 AM
9	ሎሚ እውን ከም ቀደምና ኣለና ኣብ ቅኑዕ መስመርና ተዓወቲ ኢና	VOA Tigrigna	Hate	19/02/2021 time: 5:50 AM

Table 3. 6: Data source on social media plate-form

Collected data set was stored in a CSV format, for convenience of make experimental analysis. The classes are represented as, 1for hate speech and0 for hate-free speech.

sample2.csv	
Insert Page Layout Formulas Data Review View Developer Help	
No	post and comments
1	መልሲ ናይ ኢንጅነር አስቴር ብ ቋንቋታት እዩ ዝውሃብ ነይሩ ብትግርኛ አምሓርኛ ከም ኡ ድማ እንግሊዘኛ ሕውስ ውስ ዝበለ አጠቓቕማ ቋንቋ እሞ ድማ ካብ ምኽትል ሓላፊት ማኒ ትምህርቲ ክልል ትግራይ ብዝኾነ ከምዚ ዓይነት ናይ ቋንቋ ምትሕንፋፅ ካብ ምሁራት ከሳብ መዓስ ኮን ይኸውን
2	እንታይ ዓይነት ሰባት እዮም በጃኻትኩም ቤት ትምህርቲ ዝበሃል ኣብ ዝዓነወሉ ሰዓት ከምዚ ዓይነት ሓሳብ ምሃብ መሕፈሪ ዩ ምኽትል ሓላፊት በጃከን ኣብቲ ህዝቢ ዘሎ ትግር ንምሸፋን አይትሞከራ
3	ሕድርኹም አይነዕብርን
4	አስቴር እንታይ ኢኪ ትብሊ ዘለኪ ዝተጋነነ ኢልኩም እቲ ብብ ማዓልቲ ዝቅተል ዘሎ መንእሰይ አይራአየከን ዘሎ
5	ጎሓዩ ዝተጋነነ ዝኸውን ኣብ መሬት ዘሎ ሓቅከ መዓልታው ብአሻሓት ዝቆፀር ህዝብ ኢንዱስትሪ
6	እዋይ ሰብና ኩሉ እዩ በንዲዱ እንታይ ኢና ንገብሮ
7	ዋእ ኣብ ትግራይሲ ካብዝኩሉ ተምህራይ ጥራይ እዮም ተፈተንቲ ዘለው ከንደይከ ትሕስው
8	ከሓዲት ህዝባን ሃገራን ባንዳ ዝተጋነነ ኢልኩም ማሕጋፅ ኮይንኪዬም እምበር ድሓር ውን ንዓኺ አይከገድፋኺን ዩም እምሓናን ናይ ኤራትራ ፉሽሰታውያንን ተጋሲሶም ምስማር መሊእም ኣብ ዓይደር ከንርአየከን ከንሰምዕን ርሑቕ አይከኸውንን ዩ ሸው ንዓኺ ከአ ተጋንን ኣላ ዝብሉ ተካእትኺ ባንዳ ከሰምዕ ኢና እዚ ዩ ታሪኽን ባህርን
9	ዓአሰኺ ኸመን ኣኺ ጋለይ ቶማም
10	ተጋሩ ቋንቋ ዮብሎምን ን አምሓርኛ እያ ናይ አገው ድማ ይህልዎ እዩ ገለ ድማ ለለ ለለ ዝብል ይበዝሖ ትርጉም ዘዮብሉ ቃላት
11	መንጋግ ባዳ እንታይ አፍልጡኪ ብዝዕባ ትግራይ ማሕጋፅ አርጊት ከፍአቲ ኣብ አርጋንኪ ሓሰት ለሚድኪ
12	ዝተጋነነ ምፍርራሕ ጀላዕ ኣብ ዝባኑ ዓረር እናዘነበሰ ዝተጋነነ ትብሊ ፍርቆም ዝኾኑ ቱይልዎም ይኸውን እምበር ከንደይ መንእሰይ ድዩ ዝቕንጠብ ዘሎ ወይ ጊዜ
13	እዛ ሓላፊት ተመሃሮ ይፈተኑ ምባላ እኺ ድሓን ንቀበሎ ምኽንያት ዋሕዲ ተፈተንቲ ዝተጋነነ ፕሮፖጋንዳ ብዘዕባ ቅትለትን ዓመፅ ደ አንስትዮን እዩ ኢላ እዛ ሰብ ልዕሊሽሕ መንእሰይ ሰብ ተቐይሎ ደ አንስትዮ ብዘሰካሕኩሕ መንገዲ ተዓሚፀን ከንደይ መቐተልትን ዓመፅን ምስተፈፀመ እያ ዘይተጋነነን ምኽንያታውን ስግላት እትብሎ መሕዘኒት ሹመኛ ፒፒ ትምህርትን ሕክምናን ዒላማ ኢሳ

Figure 3. 4 : sample data set in CSV format

3.5 Feature extraction for hate speech detection

Feature extraction was the process of converting from human understandable language (text format) in to computer understandable language (machine language format). Algorithms were needed to numerical features to understand classification rules. The methods involved in selecting a subset of relevant features that were helped in identifying hate, and hate-free from the labeled dataset and can be used in the modeling of the detection problems. Though there are many feature extraction algorithms, in this study N-gram, and term frequency inverse- document frequency (TF* IDF) are used.

3.5.1 N-gram model

N-gram is a statistical language modeling technique which can compute the probability of sequence of words. Especially, it was used for assigning the possibility of a probability score to the next word in the sentence [8]. With the given sequence of N-1 words, an N-gram model predicts the most probable word that might follow this sequence. It's a probabilistic model that is trained on a corpus of text. These three n-grams used because when the number of N increases, the model performance remains the constants [8]. This was estimated using the following formula.

Assume that the statement has words (w_1, w_2, \dots, w_n) , N-gram approximation is computed using following formula [80].

$$\begin{aligned}
 p(w_n/w_1w_2,\dots,w_{n-1}) &\approx p(w_n/w_{n-N+1}, \dots, w_{n-1}) \\
 p(w_n/w_1w_2,\dots,w_{n-1}) &= \prod_{k=1}^n p(w_k/w_{k-1}) \\
 p(w_n/w_1w_2,\dots,w_{n-1}) &= \prod_{k=1}^n p(w_k/w_{k-1}) \tag{3.1}
 \end{aligned}$$

The unigram is the probability of the single word and the probability of each word is independent of any words before it.

$$\text{Uni-gram: } p(w_n) = p(w_k) \text{ where } k = 1,2,3, \dots n \tag{3.2}$$

The bigram model approximates the probability of a word given all the previous words by using only the conditional probability of one preceding word [40].

$$\text{Bi-gram: } p(w_n/w_1w_2,\dots,w_{n-1}) \approx p(w_n/w_{n-1}) \tag{3.3}$$

Similar for tri-gram approximates the probability of a word given all the previous words by using only the conditional probability of two preceding word [40]

$$\text{Tri-gram: } p(w_n/w_1w_2,\dots,w_{n-1}) \approx p(w_n/w_{n-2} w_{n-1}) \tag{3.4}$$

For this study n-gram algorithm use for feature extraction task only, therefore the algorithm uses to count the number of single words with unigram, neighboring continuous sequences of two and three words in bi-gram and tri-gram respectively. The output frequency number of words in unigram (single word), bi-gram (current word one preceding word) and tri-gram (word and two preceding words) become as input for classification algorithm to build hate speech detection model.

3.3.2. TF*IDF vectorizer

TF*IDF is a statistical measure that evaluates how relevant a specific word or phrase is to a given document. This is done by multiplying two metrics: how many times a word appears in a document (TF), and the inverse document frequency of the word across a set of documents (IDF). TF*IDF (term frequency-inverse document frequency) was invented for document search and information retrieval. It works by increasing proportional to the number of times a word appears in a document [8].

Mathematically, TF, IDF and TF*IDF are expressed as follows [8]:

$$\text{Term frequency (TF)} = \frac{\text{numbers of time term appear in a document}}{\text{total Number of terms in the document}}$$

$$\text{Inverse Document Frequency (IDF)} = \log_e \left(\frac{\text{total number of document}}{\text{number of documents with term in it}} \right)$$

$$\text{TF*IDF} = w_{i,j} = \text{TF}_{i,j} \times \log_e \left(\frac{N}{\text{DF}_i} \right) \quad (3.5)$$

Where i terms in document j

$$\text{TF}_{i,j} = \text{number of occurrences } i \text{ in } j$$

$$\text{DF}_j = \text{number of documents containing } i$$

$$N = \text{total number of documents}$$

3.6 Supervised Machine Learning

One of the tasks of supervised machine learning is the process of classifying the text as hate and hate-free speeches. The main purpose the study was to detect hate speech post and comments automatically. Therefore, machine learning model was used to detect weather post and comments are hate or hate free speech after trained a classification algorithm using training dataset. There are many machine learning algorithms used to build machine learning models for binary classification task. It is hard to know which algorithm would be performing the best result. Hence, we selected the most popular machine learning algorithms such as Naive Bayes, Support Vector Machine (SVM) and Random Forest (RF) for the experiment.

3.6.1 Naive Bayes

Naïve Bayes algorithm is a probabilistic machine learning algorithm based on the Bayes Theorem, used in a wide variety of classification tasks. Naive Bayes is a classification

algorithm for binary (two-class) and multi-class classification problems. The technique is easiest to understand when described using binary or categorical input values.

According to Bayes' Theorem, pair of features being classified is independent of each other in text classification dataset. In this study, datasets were divided into two, which are feature matrix and target vectors. The Feature matrix (t) contains all the vectors of the dataset in which each vector consists of the value of dependent features. In this research, all data collected from posts and comments in Tigrigna language were dependent future. The number of features were n i.e., $t = (t_1, t_2, t_3 \dots t_n)$. The target vector (h) contains the value of class variable for each row of feature matrix. In this study hate speech detection targets or class vectors are hate and hate-free, which are used to classify the past and comments in social media. Bayes' Theorem computed the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as follows [18] [17] [19].

$$p(A/B) = \frac{P(B/A) P(A)}{P(B)} \quad (3.6)$$

$$\text{i.e. } P(A/B) = \frac{P(A) \cap P(B)}{P(B)} = \frac{P(A) P(B/A)}{P(B)} \quad \text{How?}$$

The probability of two events A and B happening $P(A \cap B)$ is, probability of A, $P(A)$ multiplied by the probability of B given A has been occurred $P(B/A)$.

$$P(A \cap B) = P(A)P(B/A)$$

the probability of A and B is also equal to the probability of B times the probability of A given B.

$$P(A \cap B) = P(B)P(A/B)$$

Equating the two yields:

$$P(B)P(A/B) = P(A)P(B/A) \quad \text{since } P(A \cap B) = P(A \cap B)$$

and thus

$$P(A/B) = \frac{P(A)P(B/A)}{P(A)}, \text{ where } P(A) \neq 0$$

Where:

A and B are called events.

$P(A | B)$ is the probability of occurrence of event A, given event B has already occurred.

Event B is also termed as evidence.

$P(A)$ is the priori of A (the prior independent probability, i.e., probability of event before evidence is seen.)

$P(B | A)$ is the probability of occurrence of B, given event A has already previously happened

For building a model to classify text from Facebook posts and comments as hate and hate-free, assume that classes (hate and hate-free) are represented by h and feature vector with dimension n by t; i.e.

$t = (t_1, t_2, t_3, \dots, t_n)$, where n is the number of variables/features. $P(h/t)$ is the probability of observing the class h given the sample t with $t = (t_1, t_2, t_3, \dots, t_n)$, where d is the number of variables/features of the sample. Assume that all features in t are mutually independent, conditional on the category h:

$$P(h/t_1, \dots, t_n) = \frac{P(h) \prod_{i=1}^n P(t_i/h)}{P(t_1), P(t_2), \dots, P(t_n)}$$

The denominator becomes constant for the given input

$$P(h/t_1, \dots, t_n) = P(h) \prod_{i=1}^n P(t_i/h)$$

To find the probability of a given sample for all possible values of the class variable h, we just need to find the output with maximum probability:

$$h = \mathit{argMax}_y P(h) \prod_{i=1}^n P(t_i/h) \quad (3.7)$$

3.6.2 Support Vector Machine (SVM)

Support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model set of labeled trained data for each category, they're able to categorize new text. The objective of the SVM algorithm is to find the hyperplanes in N-dimensional space that distinctly classify the data points. In this study hate speech detection problem have two classes which are hate and hate-free data

points. Therefore, two-dimensional space hyperplanes were appropriate for this study. Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes (see figure 3.5).

The support vector machine usually deals with pattern classification that means this algorithm was used mostly for classifying the different types of patterns [20] [21]. Basically, the main idea behind SVM was the construction of an optimal hyper plane, which can be used for classification, for linearly separable patterns. The optimal hyper plane is a hyper plane selected from the set of hyper planes for classifying patterns that maximizes the margin of the hyper plane means the distance from the hyper plane to the nearest point of each pattern. The main objective of SVM is to maximize the margin so that it can correctly classify the given patterns, as the margin size increase classifies the patterns correctly also increased [21].

The linear vector machine begins the concept of linear algebra of vector space as already known vector have direction and magnitudes. The magnitude of n-dimensional vector can be computed as follow

$$||x|| = \sqrt{x_1^2 + x_2^2 \dots x_n^2} \tag{3.8}$$

The unit vector \hat{x} is a vector with length 1 and it computed by

$$\hat{x} = \frac{x}{||x||}$$

We commonly know that the linear equation is of is $y = ax + b$ therefore it can define to the hyperplanes of the classifier

$$y = ax + b \tag{3.9}$$

$$y = ax + b \Rightarrow y - ax - b = 0$$

Which provided two vectors and can computed with, w and x the vector points of our data

$$w \begin{bmatrix} -b \\ -a \\ 1 \end{bmatrix} \text{ and } x \begin{bmatrix} 1 \\ x \\ y \end{bmatrix} \text{ then when make them } \omega^T x = -b * (1) + (-a) * x + 1 * y$$

$$\omega^T x = y - ax - b$$

As shown the two equations where just two different ways of expressing the same thing

For classifying the binary variables using support vector machine algorithm we use the equation

$$\omega^T x + b$$

Where ω , *weight vector*, b is bias and x vector of the data point need to classify

Using support vector algorithm, we need to classify binary data points. In this study, hate and hate-free speeches are used for classifying the data points using two hyperplanes which have distance between them called margin. Assume that x^+ vector hate free data point and x^- is hate-speech vector data point. To get the distance between data points, it needs to subtract between two data vectors.

$$\text{Margin distance} = x^+ - x^- \cdot \omega = (x^+ - x^-) \cdot \frac{\omega}{\|\omega\|} =$$

$$x^+ \cdot \frac{\omega}{\|\omega\|} - x^- \cdot \frac{\omega}{\|\omega\|} \quad \text{is a margin distance between hyperplane}$$

In the SVM algorithm, we are looking to maximize the margin between the data points and the hyperplane. The loss function that helps maximize the margin is hinge loss.

$$L(x) = \sum_{i=0} \max(0, 1 - y_i [\omega^T x + b]) + \lambda \|\omega\|_2^2$$

Now the maximized margin formula is drive from hinge-loss. If a data point is on the margin of the classifier, the hinge-loss is exactly zero. Hence, on the margin, we have

$$\sum_{i=0} \max(0, 1 - y_i [\omega^T x + b]) = 0$$

$$\Rightarrow y_i [\omega^T x + b] = 1 \quad \text{When } y_i \text{ is either } 1 \text{ or } -1$$

Finally compute the maximized margin distance as follow

$$x^+ \cdot \frac{\omega}{\|\omega\|} - x^- \cdot \frac{\omega}{\|\omega\|} = \frac{1-b}{\|\omega\|} - \frac{-b-1}{\|\omega\|} = \frac{2}{\|\omega\|}$$

$$\text{Marginal distance} = \frac{2}{\|\omega\|}$$

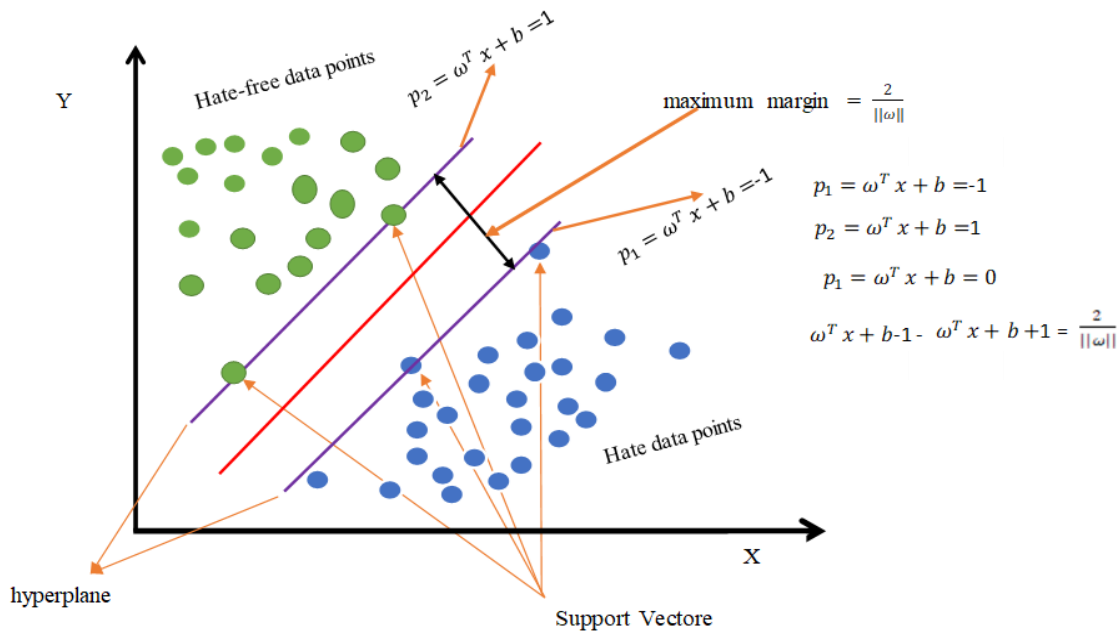


Figure 3. 5 : Support vector machine binary classification with hyperplane [20]

The Goal of the support vector machine algorithm was to find a hyperplane in an N-dimensional space (N the number of features) that distinctly classified the data points. In this study to separate the two classes of data points (hate and hate-free), there are many possible hyperplanes that could be chosen. The objective is to find a plane that has the maximum margin, the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence. Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes.

3.6.3 Random Forest

Random forest is a supervised machine learning algorithm that can be used for solving classification and regression problems [22] [23] [24]. As the name suggests random forest is a combined multiple decision trees to create a “forest” and feed random features to them from the provided dataset. The random forest takes prediction from all the trees and selects the best outcome through the majority voting process. Decision tree is more rule-based system. Given the training dataset with targets and features, the decision tree algorithm was come up with some set of rules. The same set rules can be used to perform the prediction on the test dataset [24]. The target of the Decision Tree is to create a training model which can use to predict class (hate and hate-free in

hate speech detection problem) by learning decision rules inferred from trained dataset. For the study, hate speech detection application, had two class (binary class), hate and hate-free speeches. For implementation of random forest classification algorithm, it needs to build random forest model. To build a model first it needs to create decision trees using randomly selected trained dataset features or using subsets of trained data set data with their labeled class (hate or hate-free speech in this study) and create root nodes using best splitter points. Finally build forest of decision tree for n times to create n numbers of trees, which had a set of rules used to predict unseen data.

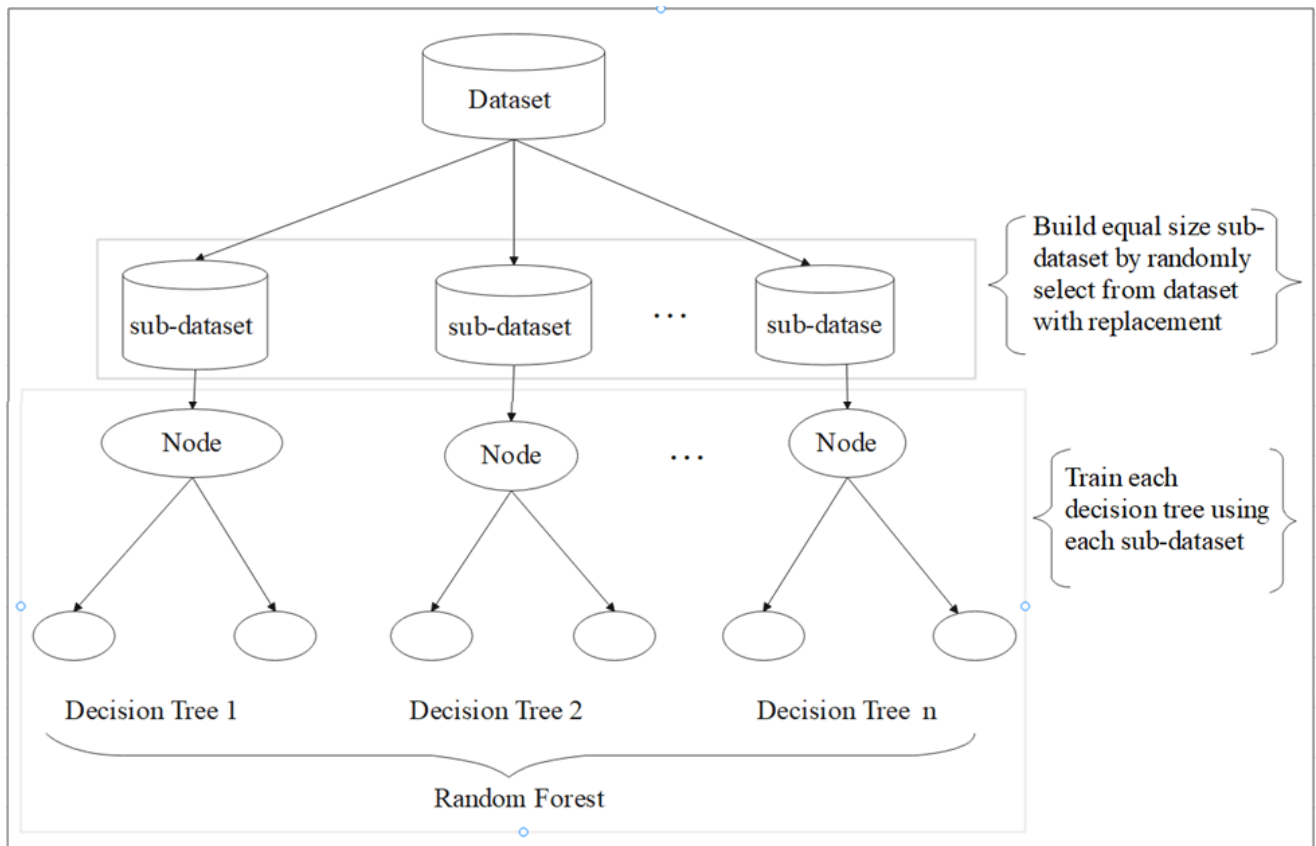


Figure 3. 6: building random forest algorithms [22]

To perform the prediction of hate or hate-free using the trained random forest algorithm, we need to pass the unseen data (test features) through the rules of each randomly created trees and calculate the votes for each predicted target. Then, the high voted predicted targets were selected as the final prediction from the random forest algorithms. The decision tree has two child nodes as shown in figure 3.4 below: -

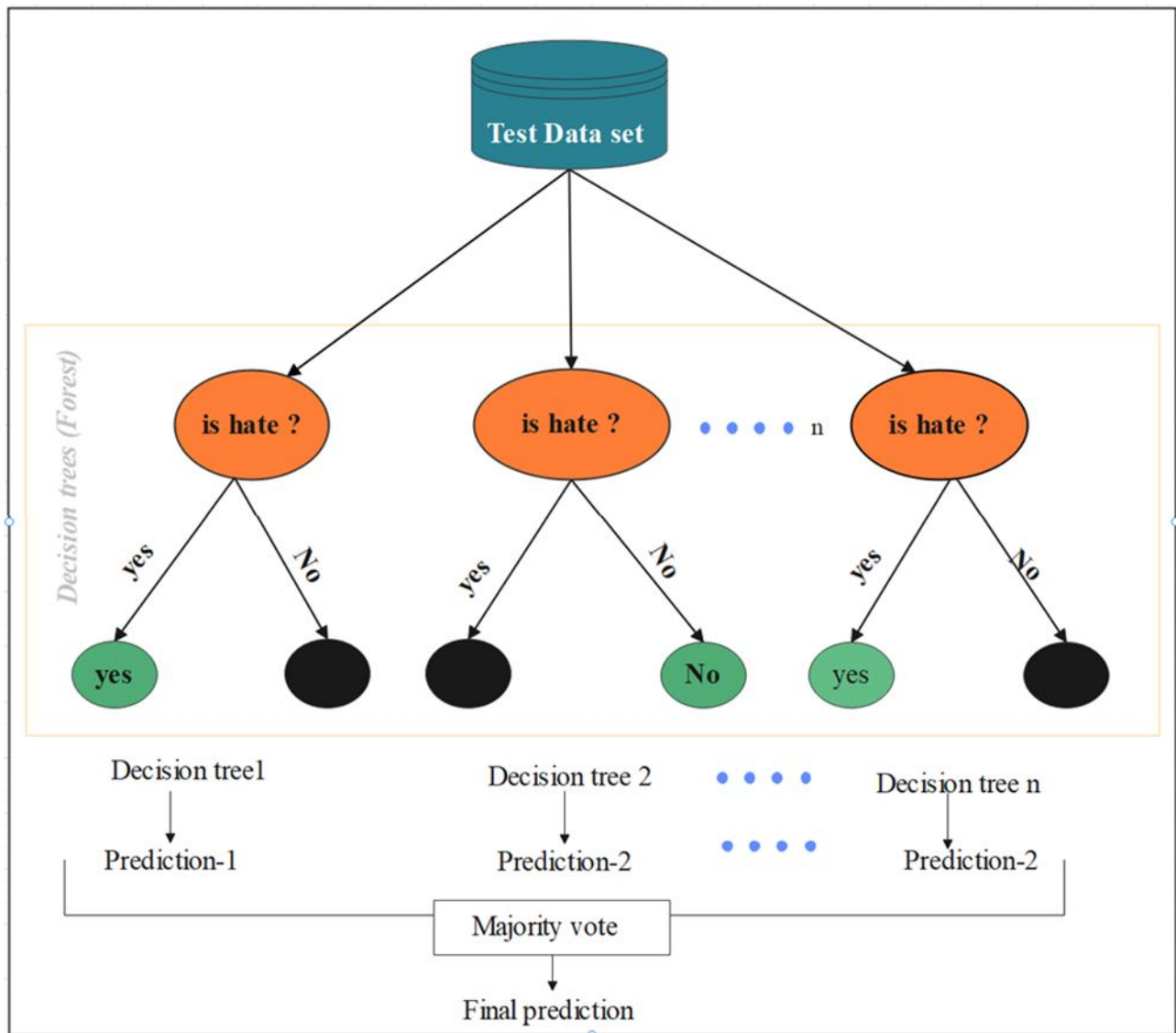


Figure 3.7 : Random Forest algorithm decision tree for prediction

CHAPTER FOUR

EXPERIMENTATION AND DISCUSSION

4.1 Overview

This chapter includes implementation of experiments and discusses the result of experiment. For make experiments, the collected data should be preprocessed, includes removing unnecessary symbols, emoji, punctuation and foreign characters other than Ge'ez character and normalize labeled Tigrigna language character have the same sound in to one. To build hate speech detection model first it needs to implement (make analysis) feature extraction using different algorithms to covert text data in vector, which is an appropriate input for the machine learning algorithms used to build the model uses to build model. After we build the model, evaluation is performed using test dataset to select the best model for hate speech detection in Tigrigna language. Evaluation process is performs using different model evaluation metrics to select the best predictor model of hate speech out of the selected classification algorithms.

4.2 Preprocessing implementation

4.2.1 Removing unnecessary symbols and punctuations

To implement preprocessing for Tigrigna text data used python programming language with some appropriate modules. Researcher write python program with regular expression (Regex) using NLTK modules. The python program read CSV file and preprocessed the data to remove special characters and symbols, emoji, foreign characters (all characters except Ge'ez) and numbers and save as CSV format. The figure 4.1 shows the screenshot of python program for preprocessing texts in Tigrigna language.

```

def Rpunctuation(line):
    for ch in line:
        if ch in "! !: # $ % & ' ( ) * + , - . : ; <0xa0> . - √ : \"' & * / + - [ ]
            i , _ 🚩 🚩 “ : $ ”:
            line = line.replace(ch , ' ')
    return line

def Rforiegn_char(line):
    Rfg = re.findall(r'^[A-Za-z0-9]+',line)
    ss = ' '.join(Rfg)
    return ss

def RNumbers(line):
    RNm = re.findall(r'^[0-9]+',line)
    rss = ' '.join(RNm)
    return rss

def REmojis (line):
    emoji = re.findall(r"^\U0001F1E0-\U0001F1FF \U0001F300-\U0001F5FF\U0001F60
0-\U0001F64F \U0001F680-\U0001F6FF \U0001F700-\U0001F77F \U0001F700-\U
0001F77F \U0001F800-\U0001F8FF \U0001F900-\U0001F9FF \U0001FA00-\U
0001FA6F\U0001FA70-\U0001FAFF\U00002702-\U000027B0]+",line)
    ess = ' '.join(emoji)
    return ess

import nltk
import re
text = 'Trainingdata.csv'
with open(text , 'r', encoding = 'utf-16le' ) as f:
    for data in f.readlines():
        RP = Rpunctuation(data)
        RFC = Rforiegn_char(RP)
        RN = RNumbers(RFC)
        RE = REmojis(RN)
        fb = open ("PreTrainingdata.csv","a", encoding = 'utf-16le')
        fb.write(RE)

print("your Data is preprocessing is Finished ")

```

Figure 4. 1:: python code for preprocessing Tigrigna Texts

Other preprocessing task is normalizing text data set, Normalization of text data set is the process of representing repeated Ge'ez characters in to one selected character. Unlike Amharic language, Tigrigna language has few characters which have similar sound but different symbol characters. These characters are [(ሀ ጎ), (ሰ ሠ), (ጸ ፀ)]; so, the researcher wrote a python program code to normalize Tigrigna language. The complete code is attached in appendix B.

4.2.2 Implementation of post and Comment Tokenization

After the cleaning and normalizing tasks, the Tokenization method follows, which splits the post and comment text into individual words or tokens by using spaces between words. Each post and

comments text data are raw data or it is collection of words but machine can't understand the textual data to build classification model (tarin mode). Therefore, textual data needs to represent in to Vector of numerical Format this process is called feature extraction. To make feature extraction for all words it should be split each word of post and comment statements in to word or token using tokenization process. Therefore, tokenized Facebook post and comments used as input for feature extraction process to represent words in to vector numerical format. Tokenization is important to correct or remove two or more words when there is no space between them. To get token or word of the text in the dataset, researcher used a python module, NLTK. We used `pythonword_tokenize ()` function of NLTK method for tokenizing Tigrigna text into word.

```
def tokenizer(text):
    token =[]
    for sent in nltk.sent_tokenize (text):
        for word in nltk.word_tokenize(sent):
            if len(word) < 2:
                continue
            token.append(word)
    return token
```

Figure 4. 2: Sample code for tokenization sentences

4.3 Feature extraction

Text feature extraction has a vital role in text classification, directly influencing the accuracy of text classification [1]. It is based on vector space model (VSM), in which a text is viewed as a dot in N-dimensional space. Datum of each dimension of the dot represents one feature of the text.

This study used the python Scikit-learn module to implement feature extraction using TF*IDF and N-gram model. Feature extraction of N-gram and TF*IDF algorithms gives different sets of feature vectors with different vector size. These features vectors were used as input to train classification machine learning algorithms, SVM, NB and RF to build models. Table 4.1 shows the extracted features vector size for the dataset.

No	Feature extraction methods	Feature vector size
1	Unigram	33,524
2	Bigram	82,013
3	Trigram	85,838
4	Unigram and bigram	115,804
5	Unigram, bigram and trigram	201,642
6	TF*IDF	33,791

Table 4. 1 : Results of the Extracted Features Vectors Size

4.3.1 Implementation of N-gram

N-gram is one of the mostly used text feature extractor model. To implement N-gram, the study Import CountVectorizer class from feature_extraction.text library of sklearn package. This class converts text data of Facebook posts and comments to a matrix of N-gram features. CountVectorizer are the process of converting raw text to a numerical, vector representation of words and n-grams. This makes it easy to directly use this representation as features in Machine Learning tasks for text classification. The study was performed repeatedly with different ngram_ranges, such as unigram, bigram, trigram and their combinations. Figure 4.3 sample code returns the trigram feature vectors of training dataset below.

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn import metrics
vectorization = CountVectorizer(ngram_range=(3,3))
xvt_train = vectorization.fit_transform(x_train)
xvt_test = vectorization.transform(x_test)
```

Figure 4. 3 : sample code for tri-gram feature extraction

4.3.2 Implementation of TF*IDF feature extraction

TF*IDF is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: - how many times a word appears in a document, and the inverse document frequency of the word across a set of documents. The main purpose is to transform text to numbers knows as text Vectorization. Once text data transformed into numbers, in a way that machine learning algorithms can understand. TF*IDF enables us to

gives us a way to associate each word in a document with a number that represents how relevant each word is in that document.

To implement the TF*IDF features extraction method, this study used a TfidfVectorizer class of scikit learns (sklearn) package. This class converts a post and comment of the text dataset to a matrix of TF*IDF features vectors. Sample code using python in Jupyter Notebook environment figure 4.4 shown below.

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn import metrics
ID_IDFvectorizer = TfidfVectorizer()
xvt_train = ID_IDFvectorizer.fit_transform(x_train)
xvt_test = ID_IDFvectorizer.transform(x_test)
```

Figure 4. 4 : sample code for TF*IDF text Vectorization

4.4 Machine learning Models Implementations

There are many machine learning algorithms used to build a model. It is difficult or impossible to know the appropriate machine learning algorithm for any problem. Any machine learning algorithm must be verified by experiment whether it appropriate for a problem. Therefore, researcher was selected some classification machine learning algorithm to determine the appropriate classification algorithm for hate speech detection through experiment. The researcher selected Naive Bayes, Support vector machine (SVM) and a random forest (RF) classification algorithm to determine the best hate speech classification model.

4.4.1 Naïve Bayes

The naive Bayes Algorithm is one of the mostly used classification machine learning algorithms that supports to classify the data based upon the conditional probability values computation. It implements the Bayes theorem for computation and used class levels represented as feature values or vectors of predictors for classification. Naive Bayes Algorithm is a good algorithm for classification problems. This algorithm is a best fit for real-time prediction, multi-class prediction, recommendation system, text classification, and sentiment analysis use cases [31].

To implement the NB classifier, the study used a sklearn MultinomialNB () classifier. This classifier is suitable for classification data with discrete features vector and multi-class data. Researcher uses x_train and y_train, to train the naive Bayes classifier model. We're using the fit method and passing the parameters as shown below.

```
from sklearn.naive_bayes import MultinomialNB
```

In [19]:

```
NBct = MultinomialNB()  
NBct.fit(xvt_train , y_train)
```

Out[19]:

```
MultinomialNB()
```

In [31]:

```
ynbct_predict = NBct.predict(xvt_test)
```

Figure 4. 5 : Sample python code to train naive Bayes model

4.4.2 Random forest

Random forests are a supervised learning algorithm and most flexible and easy to use. A forest is comprised of trees and creates decision trees on randomly selected data samples, to make prediction. From each tree and selects the best solution by means of voting. To implement the random forest classifier, the *RandomForestClassifier* () method of sklearn package was used to train the classifier. This classifier fits several decision tree classifiers as declare in n_estimators parameters. Sample code for random forest classifier is show below

```
from sklearn.ensemble import RandomForestClassifier
```

```
RFct = RandomForestClassifier (random_state=0)  
RFct.fit(xvt_train, y_train)
```

Out[29]:

```
RandomForestClassifier(random_state=0)
```

Figure 4. 6: sample python code to train random forest algorithm

4.4.3 Support Vectored machine

Support vector machine is a Classification Machine Learning algorithm, its' task is to learn to determine data points whether it belongs to two or more categories in a dataset. In geometrical

terms, associating a set of points to some category involves finding the best possible separation between these. However, for binary classification it used linear function to ensure the best separation between the two categories. Support vector machine (SVM) draws a line or “hyperplane” that divides a space into two subspaces. One subspace contains vectors (tags) that belong to a group, and another subspace contains vectors that do not belong to that group.

This study used the SVM class from sklearn package to building the SVM model. The SVC class allows us to build a kernel SVM model with linear kernel. Here is the statement to import it finally train the model using the x_train and y_train data as shown below in figure 4.7

```
from sklearn.svm import SVC
```

```
svct = SVC(kernel='linear')  
svct.fit(xvt_train, y_train)
```

Out[38]:

```
SVC(kernel='linear')
```

Figure 4. 7 : sample code to train support vector machine model

4.5 Model evaluation

Model evaluation is the process of determine how the model is good. The evaluation processes can be done by using test datasets to determine the model performance. Model evaluation aims to estimate the generalize accuracy of model on future using unseen data or out of trained data.

4.5.1 Model evaluation matrix

Evaluation metrics are used to measure the quality of machine learning model. There are many different types of evaluation metrics available to test a model. A confusion matrix is one of the best-known evaluation metrics, which gives a matrix as output that describes the complete performance of the model. Evaluation matrix involves using a combination of these individual evaluation metrics to test a model or algorithm. The use of evaluation metrics is critical in ensuring whether the model operates correctly and optimally or not. Evaluation matrix generally indicates how correctly the model predicts. Higher the score shows the better model with good prediction power. Confusion matrix is very important to use multiple evaluation metrics to evaluate a model prediction performance. Table 3.1 shows confusion matrix that compares prediction vs. actual values.

True positive: - An instance for which Hate speech Facebook post and comments are predicted as hate speech sentences

True Negative: - An instance for which Hate speech Facebook post and comments are predicted as hate-free speech sentences

False positive: - An instance for which Hate-free speech Facebook post and comments are predicted as hate speech sentences

False negative: - An instance for which Hate-free speech Facebook post and comments are predicted as hate-free speech sentences.

		Predictive value	
		Positive	Negative
Actual values	Positive	TP	FN
	Negative	FP	TN

Table 3. 7: Structure of confusion matrix [8]

When the model is used to classify the given data into hate and hate free, binary classification on confusion matrix can be used.

		Predictive value	
		Hate	Hate-free
Actual value	Hate	Hate speech sentences are predicted as hate speech sentences (TP)	Hate speech statements are predicted as hate-free speech sentences (FN)
	Hate-free	Hate-free speech sentences are predicted as hate speech sentences (FP)	Hate-free speech sentences are predicted as hate-free speech sentences (TN)

Table 3. 8: confusion matrix of hate detection model [8]

Accuracy: - Model accuracy in terms of classification models can be defined as the ratio of correctly classified samples to the total number of samples [25]. Accuracy is a good measure when

the target variable classes in the dataset are nearly balanced [25]. For this research accuracy was used to examine how correctly classify he hate and hate-free speeches. For quantifying the accuracy of model, computer was used to count the correct classified number of datapoints to total number of datapoints.

$$\text{Accuracy} = \frac{\# \text{ correctly classified data}}{\text{Total numbers of data}} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \quad (3. 10)$$

Accuracy is useful when the target class members have well balanced data but is not a good choice with unbalanced classes. When one class member of dataset is larger the other class even if the accuracy higher value it does not predict correct class. For the problem of hate speech and if hate speech class members are larger than hate-free speech class even the accuracy is higher value real hate speeches have higher opportunity to predict as hate speech.

Precision: - precision for a class is the number of true positives [25] (i.e., the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class (i.e., the sum of true positives and false positives, which are items incorrectly labeled as belonging to the class).

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \quad (3. 11)$$

Precision in the hate speech detection was the number of hate-free speech predicted as hate-free (TP) divided by sum of hate-free speech predicted as hate-free and hate-free speech predict as hate. Therefore, high precision means the model had smaller number of false positives.

Recall: - defined as the number of true positives divided by the total number of elements that actually belong to the positive class [25] (i.e., the sum of true positives and false negatives, which are items which were not labeled as belonging to the positive class but should have been).

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \quad (3. 12)$$

In hate speech detection model hate-free speech predict as hate-free divided by sum of positive and correct prediction and false negative i.e., actual hate-free speech predicts as hate.

Mathematically the precision and recall are inversely relationship, higher value of precision indicated that the lower recall. Choosing the appropriate evaluation method for a model selection, it depends on the nature of the problem. For the machine learning model of hate speech detection,

hate speeches predict as hate-free are dangerous one as compares to hate-free speeches predict as hate. But it does not mean no effect: - it may affect some freedom of speech when every hate-free speech predict as hate. Therefore, high accuracy and high precision are good evaluation metrics for our hate speech detection model.

F1-Score: This score was given the harmonic mean of precision and recall. Mathematically, F1 score is the weighted average of the precision and recall. The best value of F1 can be 1 and the worst be 0. We can calculate F1 score with the help of the following formula [25].

$$F1 = \frac{2*(Precision * Recall)}{Precision + Recall} \quad (3.13)$$

F1 score is having equal relative contribution of precision and recall

Training a machine learning model is a key step in the machine learning, it's equally important to measure the performance of the trained model. The aims of Model evaluations are to estimate the generalization accuracy of a model on future (unseen/out-of-sample) data [24][26]. Model Evaluation is the subsidiary part of the model development process and used to decide whether the model performs better [27]. Therefore, it is critical to consider the model outcomes according to every possible evaluation method. Apply different methods can provide different perspectives. In this study, the researcher used accuracy, confusion matrix, F-scale and Area Under the ROC Curve (AUC) and receiver performance measurements (ROC) evaluation metrics, which are most widely used to evaluate binary classification model. For the experiment seven different feature extractor algorithms were applied with three classifier algorithms and evaluation was performed for each model using the selected evaluation metrics.

Feature extraction method	Classification algorithms model accuracy		
	Naïve Bayes	SVM	RF
Uni-gram	78.00	74.00	72.40
Bi-gram	66.13	61.32	61.13
Tri-gram	56.20	55.50	55.30
Combining Uni- & bi-gram	74.10	74.20	71.00
Combining Uni- & tri-gram	75.00	71.00	70.00
Combining Bi- & tri-gram	62.30	56.00	59.00
Combining Uni, - bi- & tri-gram	78.10	72.60	70.30
TF*IDF	79.00	78.00	73.00

Table 4. 2: Models Accuracy for Each Features and each classification model

Accuracy shows the ratio of correctly predicted values from all data instances. That means, the sum of diagonal matrix values (true positive and true negative) in a confusion matrix divided by the sum of all four outcomes (true positive, false positive, true negative and false negative). However, type one and type two errors reduce the accuracy of the mode. The result in table 4.2 above shows the prediction accuracy score of classification models for each feature extraction algorithm.

The accuracy of models with respect to feature extraction models have different outputs. When we compare the accuracy with countvectorizer feature extraction models, Uni-gram and combination of Uni-gram, bi-gram and tri-gram have high accuracy but when we compare with TF*IDF feature extraction, it has higher accuracy. However, when we compare the classification algorithms naïve Bayes classification algorithm has higher accuracy as compared to the other two classification algorithms with all feature extraction.

Therefore, the Naïve Bayes classification algorithm using TF*IDF feature achieved 79 % of accuracy, which is the highest performance than the others. On the other hand, the lowest accuracy is 55.3 % for tri-gram feature extraction and random forest classification algorithm.

Another model evaluation metrics was precision, recall and F1-score /F-measure/. Precision shows, proportion of positive predictions is totally correct [28]. Therefore, as Precision increased

the false positive decreased. Precision shows, the proportion of predicted hate speeches that are truly hate speeches.

Recall also shows proportion of actual positives is classified correctly [28], therefore recall shows the proportion of hate speech class that correctly identified by the model. However, recall and precision have inversely relation, for hate speech detection problem higher value of recall are preferable to maximize the correctly identify (detect) hate speeches.

F1-Score (f-measure) is the harmonic mean of precision and recall values for a classification problem it means measure of a test's accuracy that considers both the precision and the recall of the test to compute the score.

Feature extraction method	Results of Classification algorithms								
	Naïve Bayes			RF			SVC		
	P	R	F1	P	R	F1	P	R	F1
Uni-gram	78	78	78	74	72	71	74	74	74
Bi-gram	67	66	64	64	61	56	65	61	56
Tri-gram	59	56	44	60	55	40	65	55	41
Combining Uni- & bi-gram	74	74	74	74	71	69	75	74	74
Combining Uni- & tri-gram	75	75	75	73	70	68	72	71	70
Combining Bi- & tri-gram	64	62	60	64	59	53	64	56	47
Combining Uni - bi- & tri-gram	78	78	78	74	70	68	73	73	72
TF*IDF	79	79	79	74	73	72	77	77	77

Table 4. 3 : model precision, recall and f1-score for Each Features and each classification model

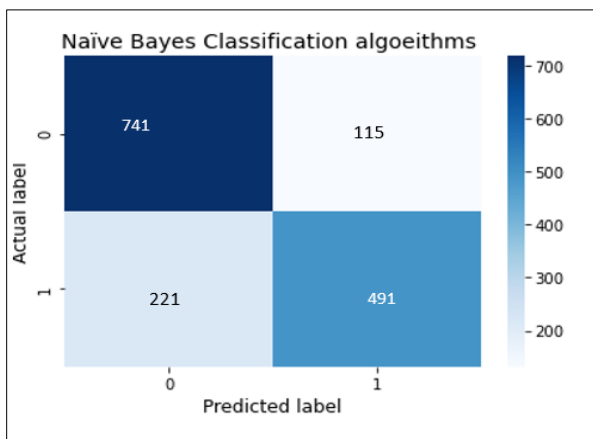
From the above table 4.3 the model metrics precision (p), recall (R) and F1-score (F1) have good prediction with Uni-gram, combination of Uni-gram, Bi-gram and Tri-gram and Naïve Bayes classification Algorithm.

	precision	recall	f1-score	support
Hate free	0.78	0.87	0.82	856
Hate	0.81	0.70	0.75	703
accuracy			0.79	1559
macro avg	0.79	0.78	0.78	1559
weighted avg	0.79	0.79	0.79	1559

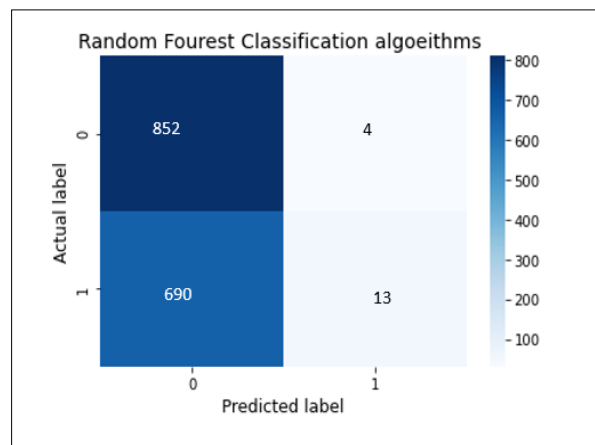
Figure 4. 8 : model metric of with TF*IDF vectorizer and Naïve Bayes classifier Algorithms

According to figure 4.8, Naïve Bayes model predict correctly hate free speech with 78% precision, 87% recall and 82% F1-score. Similarly, Naïve Bayes model predict correctly hate speech with 81% precision, 70% recall and 75% F1-score for hate speeches. Therefore, the weighted average is 79 for precision, recall and F1-scpre.

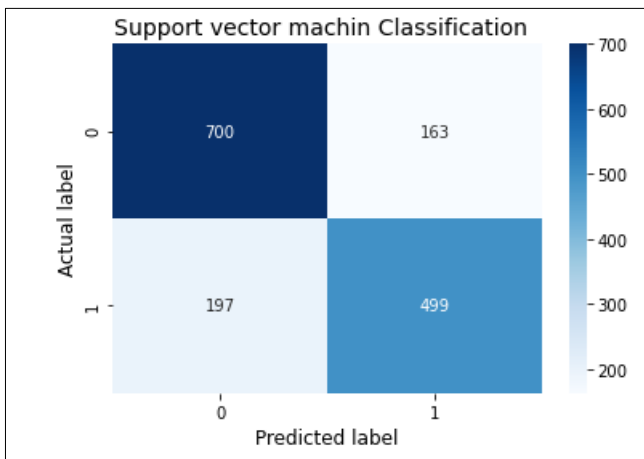
The confusion matrix shows predicted class labels against original true label data [29]. Confusion Matrices could be used to analyze the performance of a classifier and to give us insights into which direction we should work to improve our classifier. Confusion matrix offers a detailed breakdown of correct and incorrect classifications for each class.



(a)



(b)



(c)

Figure 4. 9 : confusion matrix of (a) Naïve Bayes Model; (b) Random Forest Model; (c) SVM Model

In the confusion metrics plot, the diagonal from left top to right bottom show the correctly predicted class. In figure 4.9 (a) the plot shows, Naïve Bayes algorithm predicts 741 hate-free speech instances correctly but 115 true hate-free speech were classified as hate speech incorrectly. As shown in the above figure 4.9 recalls computed as, the proportion of total true class member of hate free (856) to correctly classify hate free class members (741), that is; 87% recall. Similarly, out of all 703 true hate speech class members, 491 class members are correctly classified as hate speech and 221 class members incorrectly classified as hate free speech. Figure 4.9 (b) shows large amount of class members is misclassified. The rest classifications algorithms in between of the two-classification range.

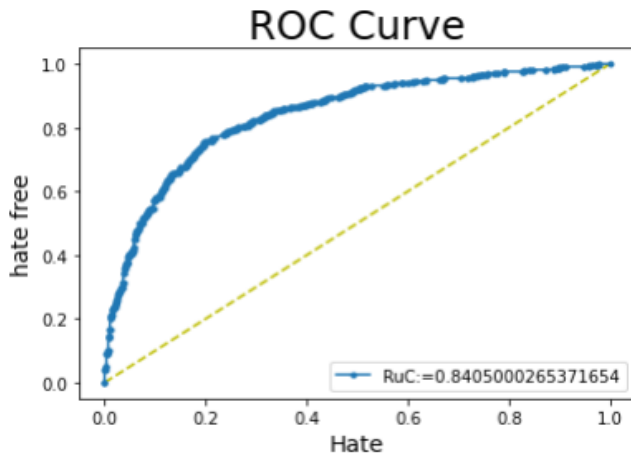
Therefore, researcher concluded that Naïve Bayes classifier algorithm with TF*IDF vectorizer have better prediction for classifying hate speech and hate free from hate speech detection problem for Tigrigna language.

Other evaluation metrics of machine learning model used in the study was receiver operating characteristic (ROC), which is good for evaluating metrics to evaluate performance of a binary classifier models, to differentiate between hate and hate free classes [30].The ROC curve is produced by calculating and plotting the true positive rate against the false positive rate for a single classifier at a variety of threshold, which would be used to indicate predict probability of an observation belonging to the positive class. In this study the probability of predict positive is at > 0.5 value and AUC stand for area under the (ROC) curve. Generally, the higher the AUC score, the better a classifier performs for the given problems.

Feature extraction method	Model evaluation using ROC		
	Naïve Bayes	RF	SVM
Uni-gram	84.91	80.40	80.17
Bi-gram	73.03	69.0	70.35
Tri-gram	58.09	57.39	57.37
Combining Uni- & bi-gram	82.00	81.33	81.33
Combining Uni- & tri-gram	82.00	81.18	80.00
Combining Bi- & tri-gram	71.00	68.00	67.25
Combining Uni- bi- & tri-gram	83.76	81.75	80.81
TF*IDF	85.12	82.15	82.47

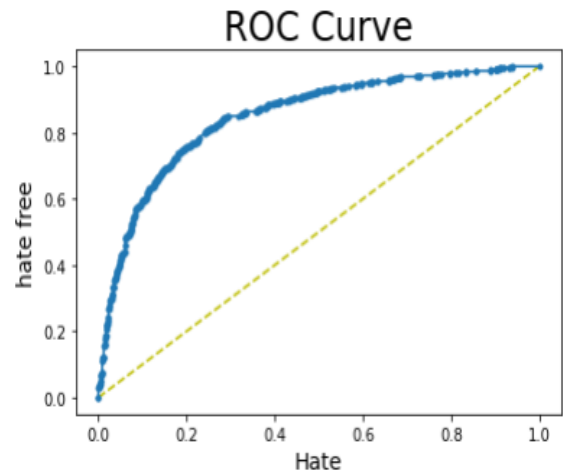
Table 4. 4 : Models ROC for Each Features and each classification model

AUC - Test Set: 84.05%



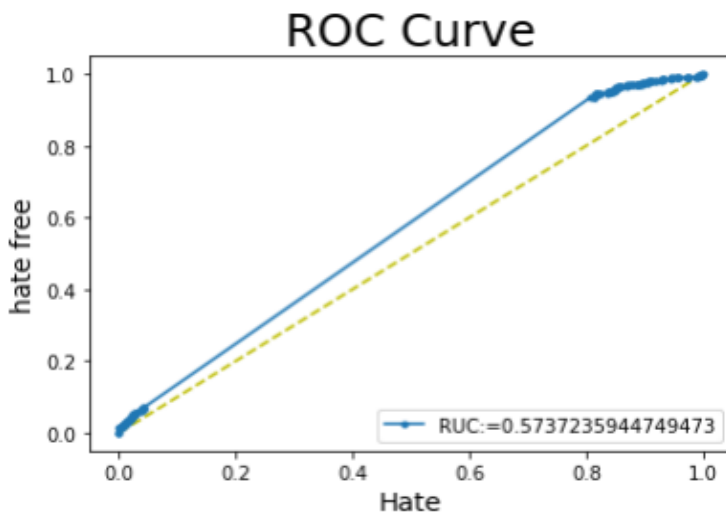
(a)

AUC - Test Set: 84.91%



(b)

AUC - Test Set: 57.37%



(c)

Figure 4. 10 : ROC curve and AUC; (a) of SVM model; (b) Naïve Bayes model; (c) Random Forest model

The AUC value of the ROC curve described in table 4.4 show different prediction probabilities values. In the study, the probability of $AUC = 1$ means it is perfect classifier and when probability $AUC = 0.5$ the model does not have predictive power [30]. Therefore, according the threshold value set for this study, the predictive probability was in between 0.5 and 1. Therefore, model with Naïve Bayes classifier and TF*IDF feature extractor has relatively higher prediction probability

with value 85.12 % (0.8512) probability and the lowest predictive probability of support vector machine classification algorithm with 3-gram with value 53.7 (0.537) predictive probability value. The ROC curve shown in figure 4.10 (a, b, c) as the curve approach to dotted diagonal line it decreases the predictive power of the model. Therefore figure 4.10 (b) has better predictive performance.

4.6 Model selection and evaluation

Based on the above experimental result, table 4.5 summarizes best results of each model as shown below.

Algorithms	Accuracy	Recall	Precision	F1-score	ROC curve	Description
Naïve Bayes	79.00	79.00	79.00	79.00	85.12	Selected
SVM	78.00	77.00	77.00	77.00	82.47	
Random Forest	73.00	74.00	73.00	72.00	82.15	

Table 4. 5 summary of best result registered by each algorithm

Researchers used accuracy, recall, precision, F1-score and ROC model evaluation techniques for evaluate hate speech classification models. According the summary table 4.5 the three classification algorithms show different performance. Naïve Bayes scored 79% accuracy, support vector machine 78% accuracy and random forest 73% of accuracy. Similarly, with evaluation metrics of F1-score Naïve Bayes score 79%, support vector machine 77% and random forest has 73% as well as with ROC curve evaluation metrics Naïve Bayer score 85.12%, Support vector machine and random forest sectors 82.47% and 82.15 % respect.

According different types of model evaluation metrics used in this study, Naïve Bayes Algorithm scored slightly higher than the other classification algorithm for hate speech detection in Tigrigna language. Finally, researcher selects Naïve Bayes Classification Algorithm as best model for hate speech detection

Therefore, researcher selects Naïve Bayes algorithm with a better performing for the hate speech detection in Tigrigna language as compare other used classification algorithm used in the study.

4.6.1 Evaluating the proposed Hate speech detection Model

The main purpose of building a supervised machine learning model is to identify hate speech and hate-free speech using the built-in model. To check whether the selected model really identify hate or hate free speech, we tested using Facebook comments and posts that are not used during training and validation of a model. As shown in figure 4.11 sample model can correctly identify hate or hate free post and comments.

Researcher used a simple python code to receive input Facebook post from user and predict the input data as hate or hate free using selected model.

```
def outputs (n):
    if n == 0 :
        return "Hate free speech"
    else :
        return "Hate speech "
def manualtest(posts):
    tposts = {"text":[posts]}
    new_post_test = pd.DataFrame(tposts)
    new_post_test["text"] = new_post_test ["text"]
    new_x_test = new_post_test["text"]
    new_xv_test = ID_IDFvectorizer.transform(new_x_test)
    pred_NB = NBCT.predict(new_xv_test)
    print("-----")
    print("Naives Bayes detects as : ",outputs (pred_NB))
```

Figure 4. 11 : sample python code for model evaluation using manual test data

The testing code shown in figure 4.11 was used TF*IDF feature extractor, that is TfidfVectorizer and a model built using Naïve Bayes machine classifier Algorithm which is a selected as best classifier Algorithm for this study.

No	Post and comments	Model detection	Correctness of the detection
1	የቅንየልና መራሐና	Hate free speech	Yes
2	ቆማላት ተጋሩ ሐዘ እንታይ ከምእትበልዑ ከንርኢ ኢና	Hate speech	Yes
3	ትግራይ ብፅፍርና ከንሃንፃ ኢና	Hate free speech	yes
4	ፊደሎ ዓጋመ ትግርኛ ዝመፅእ ለውጢ የለን ወያነ ጁንታ ከም ዕፋን ትጭፍጨፍ ዘለ ጦምሳቃት ጁንታ	Hate speech	yes
5	ሓሳዊት ዝኾነት ሃገር	Hate speech	yes
6	አቱም ረሳሓት መዓልቱኹም ተፀበዩ ጥራይ	Hate speech	ye

7	አብ ሕማቅ ጽብቅ አሎ ዝብሉዎ ሲ ከምዚ ኢዩ	Hate free speech	
8	ሕጂ ግርም እዝአም ከምዝአም እዮም ንኢትዮጵያ አብ ሕንፈሽፈሽ ዘእትው ዘለው ዘየለ ተሰፊታት እንደሆነ አብ ዘየድሊ ዓዘቅቲ ዘእትው ዘለው	Hate free speech	No
9	ደጋጊምና ኢልና ኢና ኤርትራያ ዝደፈረ ነዛ ዓለም አየስተማቅራን ኢዩ ካብ ወያኔ ተማሃሩ	Hate speech	yes
10	ናይ ንፁሃት ደም ክፈርደኩም እዩ ኣቲም ዑሱባት	Hate speech	Yes
11	እተን ንረድኤት ኤለን ኣትዮን ንኪናት መመላለሲ ዝጥቀሙለን ዘለ ዉ መኻይን መሊሶመን ድዮም	Hate free speech	yes
12	ካብ ዓፋር ሰመራ ናብ ኣብዓላ ቀጺሉ ናብ ትግራይ ንምእታው ኣዝዩ ቀሊ ል ዝኮነይኩን ናይ ጸጥታ ጸገም ዮብሉን	Hate free speech	Yes

Table 4. 6 : sample output of model evaluation using user entered data

4.7 Discussion of result

Many researches were done on hate speech detection of social media post and comments in previous works. However, NLP Application are language dependent so, there is no research paper on Tigrigna language to detect hate speech of Facebook post and comments. There is no previously collected data for Tigrigna language; so, researcher was collected Facebook posting and comments manually and using Facepacer API, from selected Facebook pages.

Feature extraction methods are an important process to capture patterns from dataset to model using a machine learning algorithm. It is also the process of converting from human understanding text character to machine understandable text or machine code. In the study n-gram and TF*IDF feature extraction algorithms were used. From n-gram feature extraction Uni-gram and combination of Uni-gram, Bi-gram and Tri-gram have better output as compare each other but when compare n-gram in general with TF*IDF, TF*IDF feature extraction has better output from all classification algorithm. Machine learning algorithms used in the study were Naïve Bayes, Random forests and support vector machine classifiers. In general, from experiments implantation Naïve Bayes have relatively higher performance (higher prediction power) for all feature extraction algorithms and random forest has lowest prediction performance but the prediction performance was different for different types of feature extractors, specifically TF*IDF feature extractor have higher prediction performance as compare with others.

According the evaluation of model, test dataset used to evaluate the models. Performance of prediction power of the classification models is different. Out of the three classification models

Naïve Bayes have higher accuracy with 79% accuracy value and random forest model has a lowest accuracy. So, in this study the model constructed by Naïve Bayes since it has the highest correctly classified class members and highest prediction power.

Therefore, for problem of Facebook posts and comments hate speech detection in Tigrigna language, Naïve Bayes classification algorithm with TF*IDF feature extraction gives the highest accuracy in hate speech detection as compared to other classification algorithms.

Even if the hate detection model can detect hate speech properly, it is not full enough or it does not mean that has not limitations on the study. One Limitations of the study is misclassification of the class members of hate and hate-free on the classification model. The higher classifier model Naïve Bayes Model was incorrectly classified 115 class members of hate-free speeches as hate and 221 hate speech class members as hate free this as shown in figure 5.8(a). The reason for such limitations is lack of consistency on manually labeling collected dataset and not enough amount training dataset and others. There are different challenges for label post and comments as hate free speeches; the nature of the hate speech expression, some hate speech may use hate words to express hater speech but some speeches may not seem to express hate speech but indirect its uses to transfer hater messages so, labeling such post and comment are difficult.

CHAPTER FIVE

CONCLUSION, RECOMMENDATION AND FUTURE WORK

5.1 Overview

This chapter includes the conclusion, recommendation and future works of the study of hate speech detection from Facebook post and comments in Tigrigna language. The conclusion shows basic process of implementation and analysis as well as summarized basic finding of analysis then clearly show the best findings. Result of the research should be recommended as a solution for the real-world problems of hate speech detection and future work is expected to improve Tigrigna hate speech detection.

5.2 Conclusion

This research proposed to develop a solution to the problem of hate speech on Facebook using machine learning techniques. The study attempted to develop, implement and compares machine learning and text feature extraction methods specifically for hate speech detection for the Tigrigna language.

To make the study successfully, it was essential to understand and define hate and hate-free speech on social media, explore existing various techniques used to tackle the problem and understand the Tigrigna language, as discussed in previous chapter. Also, different methods followed to implement and design models that have the capability of detecting hate speech. These methods include collecting post and comment from Facebook pages to build the dataset, develop annotation guidelines, preprocessing, features extraction using n-gram and TF*IDF as well as models training using support vector machine, Naïve Bayes and Random Forest and models testing. Finally performs model is evaluated using accuracy, confusion matrix, ROC curve and area under ROC curve (AUC).

In this study, researcher and one another annotator were manually labeled the posts and comments into two classes of hate and heat-free speeches. The binary class dataset has 6234 training posts and comments and 1559 test dataset totally 7,793 post and comments are used in this research.

Using a prepared Tigrigna language dataset, the models developed using SVM, NB, and RF classification Algorithm with eight types of feature extraction methods. The experiment performed using each feature extraction type for these three classification algorithms using the dataset. The Naïve Bayes classification algorithm with TF*IDF feature extraction could provide a better accuracy 79% and 79% of F1-score.

Generally, the researcher concluded that, the model built using Naïve Bayes classification algorithm with TF*IDF future extraction has to some extent better performance than Random Forest and support vector machine models to detect hate speech of post and comments spread on Facebook in Tigrigna language.

Even if the result of the study is promising for hate detection, it is not perfect at all, it needs a lot of works to improve accuracy and reduce errors. First, increasing the dataset size reduce the risk of misclassification hate and hate-free Facebook posts as well as improve the accuracy the model and other statistical results. According to definition of hate speech with body, hate speeches expressed with categories of political, ethnicity, religion, sexual, socio-economic aspects and others. Therefore, analysis using a different aspect of the hate speech category helps for better effectiveness of the study.

5.3 Recommendation

Machine learning has a capacity of doing and processing huge tasks which are difficult to do by human manually. Detecting hate speeches distributed online on social media for human being are difficult task. This study shows some capability of detecting hate speech for Tigrigna language using the small dataset with some inconvenient of labeling process by annotators of posts and comments.

As a result, researcher recommends that a social media owner companies, to use a machine learning models to detect hate speech distributed online on Facebook social media platform and to moderating the communication made on Facebook by reducing the number of posts and comments that have hateful contents. Governmental administration should be enforced social media companies to uses hate speech detection model for each language to handle heater messages.

Finally, the researcher recommends state governments build National Hate speech detection model distributed online. Now a day most of the countries around the world including our county Ethiopian have hate speech prevention and control proclamation. However, most time there are many implementation problems detecting hate or hate free speech; this leads to seen in fact the problems around restricting freedom of expression and solving the problem properly. However, it needs to build national level Hate speech detection model, with enough dataset and well-prepared annotating criteria for different languages. This helps to not only to detect hate speech but also support law enforcement with respect to hate speech distributed online.

5.4 Future work

This study used a supervised learning algorithm with a text mining feature extraction method to build hate speech detection models using binary class classification. However, hate speech expresses with categories; ethnic, race color, national origin, sex, disability, religion, or sexual orientation. Therefore, there is a need to conduct research in multi category classification.

Concerning labeling hate speech, it needs further investigation, not only to identify words used to express hate speech but also indirect expression used to transfer hater messages without using hate words in Tigrigna language.

Further study is needed to collect and prepare large amount of data with better annotation technique which can be used for constructing hate speech detection model with higher prediction power and improve accuracy.

Reference

- [1] Surafel. G, Kula K.T. (2020, December) “Automated Amharic Hate Speech Posts and Comments Detection Model: using Recurrent Neural Network”, Addis Ababa Science and Technology university, pp. 1-5. Available: https://assets.researchsquare.com/files/rs-114533/v1_covered.pdf?c=1631847511
- [2] Sindhu A., Sarang S., Zafar A. Sajid K., Ghulam M.,” Automatic Hate Speech Detection using Machine Learning: A Comparative Study”, International Journal of Advanced Computer Science and Applications, Vol. 11, pp. 1-2, Oct, 2020
- [3] Cambridge dictionary. Hate speech [online]. Available: <https://dictionary.cambridge.org>
- [4] De Smedt, Tom, Sylvia Jaki, Eduan Kotzé, Leïla Saoud, Maja Gwózdź, Guy De Pauw, and Walter Daelemans. "Multilingual cross-domain perspectives on online hate speech", arXiv preprint arXiv, pp. 3, Sep. 2018
- [5] Laanpere, L. “Online Hate Speech: Hate or Crime?” ELSA International Online Hate Speech Competition [online] June, 2017. Available: https://files.elsa.org/AA/Online_Hate_Speech_Essay_Competition_runner_up.pdf
- [6] Kovács, G., Alonso, P. & Saini, R. (2021, Feb.) “Challenges of Hate Speech Detection in social media”: SN Computer Science [online]. Vol. 2, pp. 2-16 Available: <https://doi.org/10.1007/s42979-021-00457-3>
- [7] Zewdu M., JH. Wang. (2018). “Social Network Hate Speech Detection for Amharic Language” in Computer Science & Information Technology-CSCP [online], pp. 41–55. Available: <https://csitcp.net/paper/8/86csit04.pdf>
- [8] Yonas K. (2019, September). “Hate speech detection for Amharic language on social media using machine learning techniques”. Adama Science and Technology University [online]. PP. 1-3, 31-35. Available: <http://213.55.101.23/>
- [9] Raufi, Bujar, and Ildi Xhaferri. "Application of machine learning techniques for hate speech detection in mobile applications." In 2018 International Conference on Information Technologies (InfoTech), pp. 1-4., 2018.
- [10] Almatarneh.S., Gamallo.P., Pena, F.J.R. and Alexeev, A. (2019). “November. Supervised classifiers to identify hate speech on English and Spanish tweets”. In International Conference on Asian Digital Libraries, pp. 1-5, Dec. 2019

- [11] Obrębska.M., (2020) " Contempt Speech and Hate Speech: Characteristics, Determinants and Consequences." *Annales Universitatis Mariae Curie-Skłodowska* [online] Vol. XXXIII, 3, pp. 10-12 Available: <https://journals.umcs.pl/j/article/viewFile/10862/8042>
- [12] Diego M. (2018, Oct 27). What is Machine Learning? [online]. Available: <https://medium.datadriveninvestor.com/what-is-machine-learning-55028d8bdd53>
- [13] Machine learning algorithms: Introduction to Machine Learning Algorithms: You're One-Stop Guide. Available online: <https://in.springboard.com/blog/machine-learning-algorithms/>
- [14] Listlink. (2019, Sep.19). An Introduction to Machine Learning Algorithms [online]. Available: <https://litslink.com/blog/an-introduction-to-machine-learning-algorithms>
- [15] Katrina W. (2021). A guide to machine learning algorithms and their applications [online]. Available: https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html
- [16] Liang, Hong, Xiao Sun, Yunlei Sun, and Yuan Gao. "Text feature extraction based on deep learning: a review." *EURASIP journal on wireless communications and networking* 2017, no. 1, pp. 5, Dec. 15, 2017
- [17] Sunil R. (2017, Sep. 11). Six Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R [online]. Available: <https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/>
- [18] Serafeim Loukas (2020, Oct. 12). Text Classification Using Naive Bayes: Theory & A Working Example [online]. Available: <https://towardsdatascience.com/text-classification-using-naive-bayes-theory-a-working-example-2ef4b7eb7d5a>
- [19] Kaviani, Pouria, and Sunita Dhotre. "Short survey on naive bayes algorithm." *International Journal of Advance Engineering and Research Development* 4, no. 11, pp. 2-5, Nov. 2017
- [20] Rohith Gandhi. (2018, jun. 7). Support Vector Machine — Introduction to Machine Learning Algorithms [online]. Available <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>

- [21] Ashis P. "SUPPORT VECTOR MACHINE-A Survey", International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com Vol. 2, Issue 8, August 2012
- [22] Saumya A. (2020, Dec.). Random Forests in Machine Learning: A Detailed Explanation [online]. Available: <https://datamahadev.com/random-forests-in-machine-learning-a-detailed-explanation/>
- [23] Tony Y. (2019, Jun. 12). Understanding Random Forest [online]. Available: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>
- [24] Saimadhu P. (2017, May 22). HOW THE RANDOM FOREST ALGORITHM WORKS IN MACHINE LEARNING [online]. Available: <https://dataaspirant.com/random-forest-algorithm-machine-learning/>
- [25] Clare L. (2020, April). More Performance Evaluation Metrics for Classification Problems You Should Know [online]. Available: <https://www.kdnuggets.com/2020/04/performance-evaluation-metrics-classification.html>
- [28] Nagesh S.CH. (2020, May). Model Evaluation Metrics in Machine Learning [online]. Available: <https://www.kdnuggets.com/2020/05/model-evaluation-metrics-machine-learning.html>
- [27] Sergen C. (2021, Mar. 7). Have you ever evaluated your model in this way? [online]. Available: <https://towardsdatascience.com/have-you-ever-evaluated-your-model-in-this-way-a6a599a2f89c>
- [28] Ryan A. Mardani. (2020, Nov. 4). Practical Machine Learning Tutorial: Part.3 (Model Evaluation-1) [online]. Available: <https://towardsdatascience.com/practical-machine-learning-tutorial-part-3-model-evaluation-1-5eefae18ec98>
- [29] Steve M. (2019, Apr. 16). Introduction to Machine Learning Model Evaluation [online]. Available: <https://heartbeat.comet.ml/introduction-to-machine-learning-model-evaluation-fa859e1b2d7f>
- [30] Doug S. (2020, Sep. 13). Understanding the ROC Curve and AUC [online]. Available: <https://towardsdatascience.com/understanding-the-roc-curve-and-auc-dd4f9a192ecb>
- [31] Priya P. (2020). Naive Bayes Algorithm [online]. Available: [https://www.educba.com\(\)/naive-bayes-algorithm/](https://www.educba.com()/naive-bayes-algorithm/)

- [32] Aminah M.R. (2020, Jul. 17). MACHINE LEARNING BUZZWORDS [online]. Available: <https://medium.com/analytics-vidhya/machine-learning-buzzwords-ddf5fd491825>
- [33] Apoorvkhare. (2020, Jul. 9). Decision Tree Algorithm, Explained [online] Available: <https://medium.com/@apoorvkhare500/decision-tree-algorithm-explained-158d02fe53e3>
- [34] Antony Ch. (2021, Feb. 2). K-Nearest Neighbor [online]. Available: <https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4>
- [35] ATLAS (2021). The Tigrinya Punctuation available [online]. Available: https://www.ucl.ac.uk/atlas/tigrinya/language_3.html
- [36] Duc, T.L., Leiva, R.G., Casari, P. and Östberg, Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey. ACM Computing Surveys (CSUR), pp.6-7, Sep. 2019
- [37] Simplilearn. (2021, Dec. 3). Understanding Naive Bayes Classifier: available [online] Available: <https://www.simplilearn.com/tutorials/machine-learning-tutorial/naive-bayes-classifier>
- [38] Mason, J. Tigrinya grammar. The Red Sea Press, Inc. New Jersey, 1996.
- [38] Pereira-Kohatsu, J.C., Quijano-Sánchez, L., Liberatore, F. and Camacho-Collados, M. “Detecting and monitoring hate speech in Twitter”, MDPI Open Access Journals. pp. 10,31, December 26, 2019
- [40] Kushal V. (2020, Dec. 22). N-gram Language Models: available [online]. Available: <https://towardsdatascience.com/n-gram-language-models-af6085435eeb>
- [41] pollfish. (2020). What is Experimental Research & how is it Significant for Your Business [online]. Available: <https://resources.pollfish.com/survey-guides/what-is-experimental-research-how-is-it-significant-for-your-business/>

APPENDIX

Appendix A: Tigrigna Hate Speech Corpus Annotation Guidelines

The goal of this guideline to give a guideline for annotating (labeling) posts and comments to the class of Hate and hate-free speech by their content. The Specific rules presented to point out that we have aimed to a more inclusive and general definition of ‘hate speech’ with some other perspectives found in literature, laws and general recommendations. Also, we want to better describe hate speech and differentiate it from offensives speech.

- Hate Speech

Hate Speech: is language and expression or kind of writing that attacks or diminishes, that incites violence or hate against groups based on specific characteristics such as race, ethnic origin, religious affiliation, political view, physical appearance, gender or other characteristics. The term ‘hate speech’ in this guideline, besides the above definition, is being used to describe post and comment that constitutes a slur or an instance of written abuse against a wide range of targets. To make it even more precise, these three characteristics are considered for the identification of a post and comment is hate speech those are:

1. The target: as the definition of hate speech state that, a target is a specific group with specific Characteristics that belongs to the group. We consider following as target in this study.

- Ethnicity
- Political group and view
- Religious
- Gender

2. An action: a post or comment contains an action that suggests the following action:

- ✓ Spreads promote or justify hatred,
- ✓ Suggesting, inciting, or calling for threatening violence.
- ✓ Discriminating or dehumanizing.
- ✓ Suggests killing, beating, evicting, or intimidating a target group

3. Use “us vs. them” a verbal expression that references to the alleged inferiority or superiority of one target group concerning to other groups. Therefore, we consider that a post or comment that has a joint presence of these characteristics must be marked as hate speech. The following are specific rules for labeling hate post or comment.

1. A post or comment is considered hate speech if there is the reference that explicit. Incitement to discrimination or just implied to hostility or violence actions of any kind or actions list above to a target group,
2. A post or comment is considered hate speech, references to the alleged inferiority or superiority of some target groups concerning others or dissemination of ideas based on target group's superiority or inferiority, by whatever means. (Us vs. Them).
3. A post or comment is considered hate speech if there is a combination of hateful expression, which is an insult, threats, or denigrating toward a target’s groups.
4. A post or comment is considered hate speech if its dehumanization or association the target group with animals or beings considered inferior on grounds in (b) above,
5. If the post or comment has direct target group slur or uses a direct word that the society denounced as hatred, hostile or disrespectful nickname of the target group.
6. A post or comment is considered hate speech Accusing or Condemning people based on their target groups.
7. A post or comment is considered hate speech if it contains stereotype which means overgeneralized belief about a given target.
8. If post or comment contains insulting, dirty, disgusting, or upsetting words but not contain any action listed above.
9. If post or comment contain violent or insulting words but not possible to explicitly identify a target group in the post/comment
10. If post or comment described or considered the target as unkind or unpleasant people.
11. If the post or comment described or associated the target with a negative feature or quality typical human flaws.
12. If the post or comment contains or refers to a given target with mocking intent.
13. If the post or comment contains defamation, which is a false accusation person or attack on a person's character.
14. If a post or comment contains insulting and disgusting word quote for other people posts to condemn the posts or comments.
 - Hate-free speech

Hate-free speeches are speeches does not contain a character that described in both hate and offensives speech section. Finally, each post or comment should be marked with class labels using the definition and Specific rules as the number and the abbreviation can be used interchangeably.

Hate-free speeches = 0

Hate speech = 1

Appendix B: preprocessing of dataset python codes

B-1): Removing unnecessary text from text dataset

Preprocess is the process of avoiding symbol, punctuation, numbers, emoji and other unnecessary foreign characters. The python code used for preprocessing is shown below

```
def Rpunctuation(line):
    for ch in line:
        if ch in "! !:;# i... ?)% @ ( ) ♥ = ... : : <0xa0>.- v:~\"'& */ + -[]
            i , _ 🚗 🚗 “ ” $ “ ”:
            line = line.replace(ch , ' ')
    return line

def Rforeign_char(line):
    Rfg = re.findall(r'^A-Za-z0-9+',line)
    ss = ' '.join(Rfg)
    return ss

def RNumbers(line):
    RNm = re.findall(r'^\d-\d+',line)
    rss = ' '.join(RNm)
    return rss

def REmojis (line):
    emoji = re.findall(r"^\U0001F1E0-\U0001F1FF \U0001F300-\U0001F5FF\U0001F60
0-\U0001F64F \U0001F680-\U0001F6FF \U0001F700-\U0001F77F \U0001F700-\U
0001F77F \U0001F800-\U0001F8FF \U0001F900-\U0001F9FF \U0001FA00-\U
0001FA6F\U0001FA70-\U0001FAFF\U00002702-\U000027B0]+",line)
    ess = ' '.join(emoji)
    return ess

import nltk
import re
text = 'Trainingdata.csv'
with open(text ,'r', encoding = 'utf-16le' ) as f:
    for data in f.readlines():
        RP = Rpunctuation(data)
        RFC = Rforeign_char(RP)
        RN = RNumbers(RFC)
        RE = REmojis(RN)
        fb = open ("PreTrainingdata.csv","a", encoding = 'utf-16le')
        fb.write(RE)

print("your Data is preprocessing is Finished ")
```

Figure A- 1 : sample python code for removing unnecessary text from text dataset

B-2): Normalization of Tigrigna language

Normalization is the part of preprocessing uses to represent characters have similar sound and character symbol in to one character. The python code used for Normalize characters are shown below.

```

def Normalization (line):
    for ch in line :
        # Replaced for character 7
        if ch == "ገ":
            line = line.replace(ch,"U")
        elif ch == "ጉ":
            line = line.replace(ch,"U")
        elif ch == "ጊ":
            line = line.replace(ch,"Y")
        elif ch == "ጋ":
            line = line.replace(ch,"Y")
        elif ch == "ጌ":
            line = line.replace(ch,"Y")
        elif ch == "ግ":
            line = line.replace(ch,"U")
        elif ch == "ገፍ":
            line = line.replace(ch,"U")
        # Replaced for character 8
        elif ch == 'ጸ':
            line = line.replace(ch,"θ")
        elif ch == "ጹ":
            line = line.replace(ch,"θ")
        elif ch == "ጺ":
            line = line.replace(ch,"q")
        elif ch == "ጻ":
            line = line.replace(ch,"q")
        elif ch == "ጼ":
            line = line.replace(ch,"q")
        elif ch == "ጽ":
            line = line.replace(ch,"θ")
        elif ch == "ጾ":
            line = line.replace(ch,"p")
        # Replace for character ሰ
        elif ch == 'ሠ':
            line = line.replace(ch,"ሰ")
        elif ch == "ሡ":
            line = line.replace(ch,"ሰ")
        elif ch == "ሢ":
            line = line.replace(ch,"ሰ")
        elif ch == "ሣ":
            line = line.replace(ch,"ሰ")
        elif ch == "ሤ":
            line = line.replace(ch,"ሰ")
        elif ch == "ሥ":
            line = line.replace(ch,"ሰ")
        elif ch == "ሦ":
            line = line.replace(ch,"ሰ")
        elif ch == "ሧ":
            line = line.replace(ch,"ሰ")
        elif ch == "ረ":
            line = line.replace(ch,"ሰ")
    return line

txt = 'CTrainingdata.csv'
with open(txt , 'r', encoding = 'utf-16le' ) as f:
    for data in f.readlines():
        ssd = Normalization(data)
        #print(ssd)
        fbs = open("NCTrainingdata.csv","a",encoding = 'utf-16le')
        fbs.write(ssd)
        #fbs.write(ssd)
print("Normalization process of the is successfully finished ")

```

Figure A- 2 sample python code for Normalization Tigrigna Language Dataset

Appendix C: sample python code for Build model and evaluation

C-1): Random Forest Algorithm

```
from sklearn.ensemble import RandomForestClassifier
```

```
RFcT = RandomForestClassifier (random_state=0)  
RFcT.fit(xvt_train, y_train)
```

```
RandomForestClassifier(random_state=0)
```

Evaluate Random forest model

```
RFcT.score(xvt_test, y_test)
```

```
yrct_predict = RFcT.predict(xvt_test)  
print(yrct_predict)
```

```
[0 1 0 ... 0 1 0]
```

Figure A- 3 sample python code for Random Forest classification algorithm

C-2): Naïve Bayes Algorithm

```
from sklearn.naive_bayes import MultinomialNB
```

```
NBcT = MultinomialNB()  
NBcT.fit(xvt_train , y_train)
```

```
MultinomialNB()
```

```
ynbct_predict = NBcT.predict(xvt_test)
```

Evaluate Neiv Bayer model

```
scoreNBcT = metrics.accuracy_score(y_test,ynbct_predict)  
print(scoreNBcT)
```

Figure A- 4 : sample python code for Random Forest classification algorithm

C-3): Support vector machine (SVM) Algorithm

```
from sklearn.svm import SVC
```

```
svct = SVC(kernel='linear')  
svct.fit(xvt_train, y_train)
```

```
SVC(kernel='linear')
```

Evaluate Support vector machin model

```
svct.score(xvt_test, y_test)
```

```
ysvct_predict = svct.predict(xvt_test)
```

Figure A- 5 : sample python code for support vector machine classification algorithm

```
import matplotlib.pyplot as plt  
import itertools  
cmap = plt.cm.Blues  
title = "Support vector machin classification metrix "  
classes = 2  
normalize = False  
tick_marks = np.arange (classes)  
plt.imshow(cm, interpolation = 'nearest', cmap=cmap)  
plt.title (title)  
plt.colorbar ()  
tick_mark = np.arange (10)  
fmt = '.2f' if normalize else 'd'  
thresh = cm.max () /5  
for i ,j in itertools.product(range (cm.shape[0]), range (cm.shape[1])):  
    plt.text (j , i ,format (cm[i, j], fmt ) , horizontalalignment= "center" , color = 'white' if cm [i ,j]>thresh else "black")  
plt.tight_layout()  
plt.ylabel('Actual data')  
plt.xlabel ('predicted label')  
plt.show()
```

Figure A- 6 : sample python code plot Support vector machine

Appendix D: sample python code manual testing

Manual testing is the process of testing the trained model using unseen data (New post) to check whether detect hate and hate-free post and comments. Researcher write a simple python code to manually test a mode as shown below.

```
def outputs (n):
    if n == 0 :
        return "Hate free speech"
    else :
        return "Hate speech "
def manualtest(posts):
    tposts = {"text":[posts]}
    new_post_test = pd.DataFrame(tposts)
    new_post_test["text"] = new_post_test ["text"]
    new_x_test = new_post_test["text"]
    new_xv_test = ID_IDFvectorizer.transform(new_x_test)
    pred_NB = NBCT.predict(new_xv_test)
    print("-----")
    print("Naives Bayes detects as : ",outputs (pred_NB))
```

Figure A- 7 : sample code for user test models

No	Post and comments	Model detection	Correctness of the detection
1	የቅንብልና መራራና	Hate free speech	Yes
2	ቆማላትተጋሩሐዘዘእንታይከምእትበልዑከንርኢኢና	Hate speech	Yes
3	ትግራይብፅፍርናከንሃንግኢና	Hate free speech	Yes
4	ሬድዮዓጋመትሕግኛዝመፅእለውጢየለንወያጁንታከምዕፉንትጭፍጨፍ ዘላጦምሳቃትጁንታ	Hate speech	Yes
5	ሓሳዊት ዝኾነት ሃገር	Hate speech	Yes
6	አቱምረሳላትመዓልቱኹምተፀበዩጥራይ	Hate speech	Yes
7	አብ ሕማቅ ጽብቅ አሎ ዝብሉዎ ሲ ከምዚ ኢዩ	Hate free speech	Yes
8	ሕጂ ግርም እዝአም ከምዝአም እዮም ንኢትዮጵያ አብ ሕንፈሽፈሽ ዘእትው ዘለው ዘየለ ተሰፊታት እንደህብ አብ ዘየድሊ ዓዘቅቲ ዘእትው ዘለው	Hate free speech	No
9	ደጋጊምና ኢልና ኢና ኤርትራያ ዝደፈረ ነዛ ዓለም አየስተማቅራን ኢዩ ካብ ወያኔ ተማሃሩ	Hate speech	Yes
10	ናይ ንፁሃት ደም ክፈርደኩም እዩ አቱም ዑሱባት	Hate free speech	No
11	እተን ንረድኤት ኤለን አትየን ንኪናት መመላለሲ ዝጥቀሙለን ዘለዉ መኻይን መሊሶመን ድዮም	Hate free speech	Yes
12	ካብ ዓፋር ሰመራ ናብ አብዓላ ቀጸሉ ናብ ትግራይ ንምእታው አዝዩ ቀሊል ዝኮነይኩን ናይ ጸጥታ ጸገም የብሉን	Hate free speech	Yes
13	መሰሓቅ ዝኮነ ኮመድያ፣ ባዕላቶም ማካይን ነዳድን ህበም ናብ ትግራይ አግዕዝዎ እለሞም ከብቅዑ ሕጅ ታሓለቅት ክመሰሉ ትዋሰኡ ይዋሰኡ አለው ዝገርም ታዋሳእትን ደረሰትን	Hate free speech	Yes
14	አዝዩ ቅኑዕን ሓቀኛን ዜና ብምቅራብኩም አዝዩ የመስግነኩም	Hate free speech	Yes

15	ብመሰረት ትልምና ናይ መጀመሪያ ምዕራፍ ወፍሪ ኣብ ክልተ ሰሙን ኣ ዐዊትና ዝበሉ ቀዳማይ ሚኒስትር ኣብይ ኣሕመድ ንዝተወሰነ ግዜያት ና ብ ቤት ፅሕፈቶም ከምዝተመለሱ ኣብ ማሕበራዊ መራኽቢ ገምገማ ኣብ ዘቐመጥዎ ጽሑፍ ገሊጾም	Hate free speech	Yes
16	ካብ ውግእ ዝረብሕ የለን ወይ ብሰላም ወይ እቲ ውጽዑ ብዓል ዕላማ ዝዕወት ዘልኣለም ሓንሳብ ትድፋእ ሓንሳብ ትድፋእ እምበር መዋእል ይ ነብር ረቢ ዝኸለቐ ፍጡር ሰብ ኣየጥፈኡን ክትዳኸም ትኸእል ንግዝይኡ መጻኢ ወለዶ እንዳ ተተከአ ድጉል ሓዊ ፈኸም ብውግእ ዝዕወት የለን ኩሉ ህውት ሰብ እዩ ዝኸፍል ሰላም ይኹን ምርጫና ጸላእና ይዋጋእ	Hate free speech	Yes
17	ደንዝ ሃሳዊ ጀግና ዘይትፈልጥ ሰራቂ ቢሑቕ ሰነፍ ዓወት ፅባሕ ተፀቢ ዘይ ደልዮ ኣይደልዮን ኣብ ዘድልየና ግዜ ኸንሕዞ ኢና ደንዝ ብኸብዱ ዝነብር	Hate speech	Yes
18	እቲ ክልቢ ይነብሕ ገመል ግንጉዕዝኡ ይቐጽል	Hate speech	Yes
19	ወያነ ዳኣ ኣዚ ጽሑፍ ድኹን ኣስትራተጂ ማዓስ ይጋደፎም ኮይኑ ካብ ት ፍጠር ሒዞ ሓንቲ ከየፍረየት ኣናረኣናያ ትቐህምን ትጠፍእን ኣላ	Hate speech	Yes
20	ኣየ መስክናይ ህዝቢ ትግራይ ናይ ወያነ ስልትስ ንመን ኢዩ ክጠፍእ ዶር ሆስ ተሓለምት ጥረ ምረ ዱ ተባህለ ወደይ	Hate speech	Yes
21	ሕወሓት ከዳሚት ኣሜርካ ሙኻን ገዲፋ ብናይ ዝኣእ ርኣሳ ውጥን ክትከ ይድ እንተጀሚራ ሓቅነት እንተለዎ እዚ ሓበሬታ ንዓና ውን ብስራት እዩ ክኸውን ዝኸእል ምኽንያቱ ዓቢ ዝላን ዓወትን እዩ	Hate free speech	Yes
22	እዚ ሕጂ ኣብ ኣፍሪቃ ብሓፈሻ ብፍላይ ድማ ኣብ ዞባ ቀርኒ ኣፍሪቃ ክለዓዓል ጀሚሩ ዘሎ ናይ ዘይተምበርካኸነትን ዘይምእዙዝነትን ቃልሲ ኤርትራ ንብዙሕ ዘመናት በይና ዝተረባረቡትሉ ቅነዕ ዕላማ እዩ ኣብዚ ሕጂ ሰዓት ናይ ማለሊት ናይ ምፍናው ስነስርዓት ኣብ ዝተቃረበሉ ንዓዕሎም እቶም ብምእዙዝነትን ብኸ ድምናን ቀንጻ መጋበርያ ዝነበሩ ኣሓት ህዝብታት ናብቲ ነኹሉ ፈኸም ክብል ዝጸንሐ ቃልሲ ህዝብን መንግስትን ኤርትራ ሰሚሮም ኣለው ህዝቢ ኤርትራ ም ስ ዋሕዱን መሪርን ብዙሕ ተጸብኡን ሃደሽደሽ ኣቢሉ መራሒ እዚ ክምህ ዘይብ ል ቅነዕን ታሪኻውን ዕላማ ኮይኑ እዩ እሞ ኣብነት ኣፍሪቃ ዝኾነ ህዝቢ ኮይኑ እዩ	Hate free speech	Yes
23	እነ ካብ ዝገርመኒ ብከመይ መለኮዒ እዩ ክሳብ ክንድዚ ዝእከል ሸሮ ሰራ ዊት ጁንታ ሰተት ኢሉ ካብ ትግራይ ነቂሉ እዚ ኩሉ ከተማታት ክኣቱ ክ ኢሉ ብፍላጥ ድዩ ወይ ብዘይፍለጥ ዝተገብረ እዩ? እዚ ጽፍዒት ናይ ሕ ጂ ካብ ንጁንታ ንላዕሊ ንሱዳን ንምስሪ ኣቀንዝይዎም ኣሎ ነቶም ካልእ ት ግን እተቀዶ ቀዶ እንተዘይቀዶ ሕንግጦ እዩ ዝኸውን ዘሎ ክሳብ ካል እ ጁንታ ከዳሚ ዝረከቡ ሰለዚ ከምኡ እዩ ዘሎ እቲ ወረ	Hate speech	Yes
24	ኣይ ኣታ ቆንዳፍ ምህዳምን ምዝላቅን ነቲ ዘይፈልጥ ድፍን ህዝቢ ትግራይ ኣታ ልለሉ እምበር እንዳተጸፋዕኩም እግረይ ኣዉጽኢኒ እንዳበልኩም ዕድለኛ ይማረ ክ ዝበዝሕ ካኣ ይረግፍ ኣሎ በረካታት ሃገረ ሰላም መን ከሊኢካ ኣብታ ዝነበርካ ያ ባዓቲ ትከደላ ማዓልታት ተሪፍካ ኣሎ ኣታ ድሁል ጺን	Hate speech	Yes
25	ወይዘሮ ደብረጸንዋይ ኣብ ኣፍ ሞት ከለክ ንሓሶት ግደፋ ምጻሩስ እና ኸናት በ ርትዑና ምቕጻጻጽንተጸዋርነቱ ከቢዱ ምባል ናይ ለባማት እዩ ዓብበዓቲታትተ ሓብእክንመራሕቲትበሃላክትድፈኣናይግድንእዩግደፍሓሶት	Hate speech	Yes
26	ክሉ እቲ ዘድሊ ተኸፊልዎን እኸፊላን ኣሎ እዞ ሃገር ብሕጂ ውን ኣሎ እ ንተኸይኑ ንሃገሩ ከም ፈረስ ደው ኢሉ ዝሓድር ሓጺን ዝኾነ ህዝቢ እዩ ዘለዎ	Hate free speech	Yes
27	ዓዲን ኣዲኡን ዘይናፍቕ ፍጡር የቀን ኣቲ ሸግር ግን ቅሳነትን ልእልና ሕ ግን ዘይምህላው እዩ ስለዚ ዝበልካ ኣንተ በልካ ኣይትሓዘለይ ዝሰምዓካ ሰብ የለን ምክንያቱ ዝተቀየረ ነገር ስለዘየለ ካብ ብዙሕ ምዝራብ ምርኣ ይ ይቐልል እዩ እሞ ሕግን ቅዋምን ስርዓትን ዘለዎ ሃገር እሞ ኣርእዩና ሸ	Hate free speech	Yes

	ዑ ከይተለመና ክሳብ ትመንዉና ክንመላለስ ኢና ወደሓንኩም ምልኪ ይፍረስ ሕጊ ይገንስ ኩሉ ክድበስ		
28	በየናይ መንገዲ እንታይ መታኣማመኒ ምስተፈጥረ ነዚ ትብልዎ ዘለኹም ንኸፍጻም እኮ መጀመርያ እታ ሃገርና ሃገራዊ ሓታትን ተሓታትን ዝህልዎ ሃገራዊ ቅዋም ናይ ህዝብታት ይኹን ናይሰራዊታት ንመራሕን ተመራሕን ብማዕረ ዝፈርድ ቅዋማት ክህሉ ኢዩ ዘለዎ እዚ ክሳብ ዘየለ ከምቲ ትደልይዎን ትሓስብዎን ክኸውን ይኸእል ኢዩ ክትብሉኒ እንተኾይኑ ኣፍሉጦ ፓሎቲካ ዘየብሉ እንተኾይኑ ክፍጽም ይኸእል ይኸውን ይብል እዚ ግን ናተይ ኣረዳዳኣ ኢዩ ንዓይ ድማ ይምልከት ጌጋታት ተለኒ ድማ ንኸእረም ይሓትት	Hate free speech	Yes
29	ክቢኒ ሚኒስትራት ሎሚ ዓርቢ ኣብ ዘካየዱ ካልኣይ ስሩዕ ኣኼባ ሓደ ካብቲ ዝኣኢ ሰልፊ ብልጽግና ቅድሚ ምርጫ ዝኣተዎ መብጽዓታት ፖለቲካዊ ዘተ ዝመርሕ ኮሚሽን ምቛም ምንባሩ ገሊጹ ንኹሉ ዘሳትፍ ሃገራዊ ልዝብን ምይይጥን ብምክያድ ሃገራዊ ምርድዳእን ኣብ ሃገራዊ ጉዳያት ናይ ሓባር መርገጺ ከምዝህሉ ዘኸእሉ ስራሓት ምስራሕን ሓደ ካብቶም መብጽዓታት ምንባሩ ቤት ጽሕፈት ቀዳማይ ሚኒስተር ኣብዩ ኣሕመድ ዘውጽኦ መግለጺ ይጠቅስ	Hate free speech	Yes
30	እተን ማሕበራት ቀይሕ መስቀልን ቀይሕ ወርሕን፡ ዝህብኦ ሰብኣዊ ረድኤ ኤት ከምዘስፍሓ ብምግላጽ ሽሕ ሰባት መጽለሊ መግቢ ማይ ገንዘብን ካልኣትን ቀረባት ከምዝረኽቡ ምግባረን ኣፍሊጠን ሓላፊ ቀይሕ መስቀል ኣብ ኣፍሪቃ መሓመድ መክታር ብወገኑ ምስ መሻርኽትና ብምቃን ረድኤ ኤት ንምሃብ ዝክኣለና ንገብር ኣለና እንተኾነ እቲ ዝውሃብ ዘሎ ረድኤት ምስ ብዝሒ ተጸባይቲ ብምንጽጻር እኹል ኣይኮነን ኢሉ ብምቅጻል ንኹሉ እቲ ሓገዝ ዘድለዩ ሰብ ረድኤት ንኸረክብ ኣስታተ 27 ሚልዮን ስዊዝ ፍራን ከምዘድሊ ገሊጹ	Hate free speech	Yes

Table A- 1: Tigrigna language post and comment used for user test of model