Tigrinya Hate Speech Detection and Classification from Facebook Posts and Comments Using Deep Learning Approaches

Girmay Gebremeskel Bahru

A Thesis Submitted to

The Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master of Science in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September, 2022

Adama, Ethiopia

Tigrinya Hate Speech Detection and Classification from Facebook Posts and Comments Using Deep Learning Approaches

Girmay Gebremeskel Bahru

Advisor: Dr. Mesfin Abebe

A Thesis Submitted to

The Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master of

Science in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September, 2022

Adama, Ethiopia

DECLARATION

I hereby declare that this master thesis entitled "***Tigrinya Hate Speech Detection and Classification from Facebook Posts and Comments using Deep Learning Approaches***" is my original work. That is, it has not been submitted for the award of any academic degree, diploma or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation

Girmay Gebremeskel Bahru _____ _____

Name Signature Date

RECOMMENDATION

I, the major advisor/supervisor of this thesis, hereby certify that I have read the revised version of the thesis entitled "***Tigrinya Hate Speech Detection and Classification from Facebook Posts and Comments using Deep Learning Approaches***" prepared under my guidance by Girmay Gebremeskel Bahru submitted in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science and Engineering. Therefore, I recommend the submission of revised version of the thesis to the department following the applicable procedures.

Dr. Mesfin Abebe _____ _____

Major Advisor/Supervisor Signature Date

APPROVAL PAGE

I/we, the advisors of the thesis entitled "***Tigrinya Hate Speech Detection and Classification from Facebook Posts and Comments using Deep Learning Approaches***" and developed by Girmay Gebremeskel Bahru hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Dr. Mesfin Abebe	_____	_____
Major Advisor/Supervisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis Girmay Gebremeskel Bahru have read and evaluated the thesis entitled "***Tigrinya Hate Speech Detection and Classification from Facebook Posts and Comments using Deep Learning Approaches***" and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science and Engineering.

_____	_____	_____
Chairperson	Signature	Date
_____	_____	_____
Internal Examiner	Signature	Date
_____	_____	_____
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

_____	_____	_____
Department Head	Signature	Date
_____	_____	_____
School Dean	Signature	Date
_____	_____	_____
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGMENT

First and foremost, I would like to praise the almighty God, for he has dwelled me under his mercy and kept me strong to complete this study. Then, I want to start by sincerely thanking my advisor, Dr Mesfin Abebe, for his unwavering support and continuous assistant for my thesis work, as well as for his patience, inspiration, excitement, and vast knowledge. His advice was helpful to me throughout the entire research and thesis-writing process, therefore I really wanted to say thankyou Doctor!

Additionally, I want to give my deepest gratitude to my sister and friend Miss Hareg Yosief for she has helped me a lot regarding the Tigrinya language. Next, I would like to thank my friends for they have shared with me what they know, and their endless support was priceless. Finally, I would like to thank my family for they have been supporting me in different aspects to accomplish this study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF SAMPLE SOURCE CODES

LIST OF ACRONYMS AND ABBREVIATIONS

API	Application Programming Interface
AUROC	Area under the Receiver Operating Characteristic
AI	Artificial Intelligence
ANN	Artificial Neural Network
BERT	Bidirectional Encoder Representation from Transformers
Bi-LSTM	Bi-directional Long-Short Term Memory
CSV	Comma-Separated Values
CBOW	Continous Bag of Words
CNN	Convolutional Neural Networks
DT	Decision Tree
DW	Dmtsi Woyane
FN	False Negative
FP	False Positive
GRU	Gatted Recurrent Unit
GB	Gradient Boosting, Giga Bytes
GPU	Graphics Processing Units
IDE	Integrated Development Environment
KNN	K-Nearest Neighbor
LR	Logistic Regression
LSTM	Long-Short Term Memory
NB	Naïve Bayes
NLP	Natural Language Processing
OOV	Out-of-Vocabulary
PR	Precision-Recall
RAM	Random Access Memory
RF	Random Forest
ROC	Receiver Operator Characterstic

ReLU	Rectified Linear Unit ReLU
RNN	Recurrent Neural Network
RQ	Research Question
SVM	Support Vector Machine
TB	Terabyte
TF-IDF	Term-Frequency Inverse Document
TMH	Tigray Media House
TTV	Tigray Television
TN	True Negative
TP	True Positive
URL	Uniform Resource Locator
WWII	World War II

ABSTRACT

*Nowadays, using social media to share information with loved ones, friends, and coworkers is one of the easiest things to do. Even though people are free to express their feelings as they choose, spreading hate speech using this platform has become more convenient. Hate speech detection is a way in which a machine with trained model detects the speech and classifies it as hate or hate-free speech accordingly. As a result, to the best of our knowledge there is no single study conducted regarding the Tigrinya hate speech detection. The main objective of this study is to design and develop hate speech detection model for the Tigrinya language from Facebook posts and comments. Hate speech detection models have been developed worldwide, including in our own country of Ethiopia, for a variety of languages by various scholars. However, due to the difference in language, the hate speech detection model designed for Amharic or Afan-Oromo won't be applicable in the case of the Tigrinya language. As a result, we have developed a model that detects the Tigrinya hate speeches for both binary and multi-class classification. In this study to achieve our objective we have collected 5400 posts and comments as a dataset, and the dataset was collected from different Facebook pages using the Facepager tool and prepared it in CSV file format. While the 5400 of the datasets have been used for multi-class classification, only 3608 of it has been applied for binary classification. We have used the stratified k fold cross-validation techniques for the purpose of splitting our dataset to train and test our model. We have designed the model using three different deep learning approaches with three different feature extraction techniques. For the purpose of designing the model the Bi-LSTM, CNN-LSTM, and CNN has been applied with the feature extraction techniques of Word2vec, Fast text and the Keras Embedding layer. We have applied the dropout and L2 regularization techniques to overcome the overfitting problem occurred in each model. As a result, for the multi-class classification the Bi-LSTM model with the Fast text of CBOW has outperformed the CNN-LSTM and the CNN model with the Accuracy of **87.41 %**, Precision of **88.02 %**, Recall of **85.74 %**, and F1-Score of **86.86 %**. Additionally, for the binary classification the Bi-LSTM model with the Fast text of skip-gram has outperformed the CNN-LSTM and the CNN model with the Accuracy of **96.11 %**.*

***Key Words**: Bi-LSTM, LSTM, CNN, Hate Speech, Tigrinya language*

CHAPTER ONE

1. Introduction

1.1. Background of the study

The world is rapidly changing into one village, which is a well-known reality. It is developing into a village in terms of information exchange, not in terms of place or geography. People before the invention of the internet, when they want to communicate and exchange information with their loved ones, such as their families, friends, and coworkers, they used to put a lot of time and effort into it. However, with time and the emergence of the Internet in the late 1960s, this approach has become obsolete[1]. Since then progressively, a lot has changed drastically, making it simpler for us to receive or send information from our home, place of work, or anywhere else thanks to the Internet of Things. The term "social media" refers to a set of Internet-based technologies that were developed on the web's technological basis and that enable the production and sharing of user-generated content[2]. Facebook is one of the platforms used by the Internet system to share information and transmit data. Facebook is a well-known social media website with many users nowadays, allowing individuals from all over the world to quickly exchange information. According to a study by Piotr Sorokwsk et al (al., 2020) 79% of internet users use Facebook, and with more than 2.2 billion members now, Facebook is the most popular social networking platform[3], and therefore it indicates that Facebook is used by the majority of social media users.

When we came to our country Ethiopia and our neighboring country Eritrea, currently the number of internet users has increased by 17.9% and 6.9% respectively[4]. As a result, the number of Facebook user's growth rate is also increasing year by year, for instance in our country in January 2020, 5.2% of the population was using Facebook and this status has increased by 1.4% by October 2021 and became 6.6%[5]. Additionally, there are Eritreans in the world who can write and speak the Tigrinya language using Facebook[6]. Therefore, these statistics shows that there is

[1] https://www.history.com/news/who-invented-the-internet
[2] https://www.unapcict.org/Socia Media Development and Governance Module
[3] https://www.websiteplanet.com/blog/social-media-platform-users/
[4] https://www.internetworldstats.com/stats1.htm
[5] https://napoleoncat.com/stats/facebook-users-in-ethiopia/2021/09/
[6] https://napoleoncat.com/stats/facebook-users-in-eritrea/2021/10/

an expeditious enhancement of internet users more specifically Facebook users than the other social networks such as tweeter and YouTube in Ethiopia[7] (Defar, 2019). Tigrinya is a member of Ethio-Semitic languages, which belongs to the Afro-Asiatic superfamily, and is primarily spoken in Eritrea and the Northern Province of Ethiopia, Tigray region. This language uses the writing system of Geez script, specifically known as Ethiopic script which is originally developed for the Geez language (Birhanu, Automatic Text Summarizer for Tigrinya, 2017).

Facebook supports and localizes various types of languages in the world (Defar, 2019), hence in Ethiopia people who use Facebook can interact and communicate with their friends and followers using their own languages and Tigrinya is among those languages. The main purpose of Facebook is to put a group of people together and create the opportunity to share their ideas, knowledge, and awareness about something they are fascinated by. Even though sharing different sorts of information using the Facebook platforms between friends, individuals, and a group of people is very advantageous and helpful, it has its own countless downsides, amidst those numerous limitations and problems the forefront is that, it is too much convenient for the widespread of hate speech.

Even if there is no formal definition of what hate speech is, there is a general consensus between scholars and service providers to define it as "any expression that targets to attack an individual person or a group on the basis of gender, race, ethnicity, religion and disability" (Al-Khalifa, 2020). The reason why Facebook became suitable for the spreading of hate speech is that someone who uses Facebook can hide behind the screen and no one can identify who he or she is, therefore he or she can disseminate hate speech by targeting some individuals or groups based on a different basis (Zewdie Mossie and Jenq-Haur Wang, 2018). In the case of our country Ethiopia, there has been an increasing conflict and violence unfolding due to the ethnic rivalry and religious tensions since April 2018, and the spread of hate speech is using both online and offline dialogue. (Ayalew, 2020). In Ethiopia Facebook has simply intensified existing tensions on a massive scale regarding outspread of hate speech (Yohannes).

But, since the world is advancing in technologies, there are different methods and approaches specifically traditional machine learning and deep learning approaches being used to tackle the online widespread of hate speech on social media. In Ethiopia's context in order to reduce the

[7] https://gs.statcounter.com/social-media-stats

widespread of hate speech across Facebook and some other social media platforms, some research on Amharic and Afan-Oromo languages has been done using traditional machine learning and deep learning approaches and they have achieved promising results. But for the case of the Tigrinya language, there is no single research done yet to minimize the amount of hate speech being spread on Facebook posts and comments based on the Tigrinya language. Tigrinya is a low-resourced language in which it has is no free or enough textual dataset or corpus prepared, and it uses under-resourced material, it has insufficient linguistic material and tools for further natural language processing tasks. In addition to this Tigrinya language is morphologically rich and complex both semantically and syntactically (Awet Fesseha, Eshete Derb Emiru et al, 2021). For those who can read and understand the Tigrinya language, the volume of hate speech produced and disseminated on the Facebook network using the Tigrinya language is damaging and catastrophic.

The hate speech being spread might be based on a group or person's religion, race, color, or gender. As a result, the positive relationships in our society are deteriorating over time because of the rise of hate speech on Facebook. Therefore, it is important to limit the propagation of hatred expressed through Tigrinya's writing. Therefore, as researchers, it is our goal to use a deep learning approach to identify hate speech from Facebook posts and comments that are written in the Tigrinya language.

1.2. Motivation of the study

Since it is easier to distribute hate speech by focusing on a specific person or group of people based on a variety of characteristics on social media, hate speech is proliferating and broadening frequently. We have been reading academic studies for a variety of languages about the use of traditional and deep learning techniques to identify hate speech on social media. According to the research papers we read, Amharic and Afan-Oromo are among the many languages being studied, and they have provided a promising result.

In the meantime, despite the fact that hate speech in the Tigrinya language is being disseminated on Facebook by people who can read and write Tigrinya here in Ethiopia, Eritrea as well as throughout the rest of the world, no research has been conducted to address this issue yet. The propagation of Tigrinya hate speech on social media has its own direct or indirect impact on the society and, which is why it was a major driving force behind our research in this field. Thus, it

was necessary to identify and forbid the dissemination of hate speech written in the Tigrinya language.

additionally, the thing that motivated us to do the research on this area is, that since Tigrinya is an under-resourced language we want to contribute something by building and developing hate speech detection models, and therefore this model and its dataset could be a contribution in the aspects of natural language processing in which it will be useful for further research in the future.

Important and key Motivations of the researchers

- To contribute our effort in achieving peace and security by detecting hate speeches being spread on Facebook using the Tigrinya language
- Since Tigrinya is morphologically rich and complex language and is highly inflected we want to contribute our effort by providing trained model which detects Tigrinya written hate speeches and therefore this could be a contribution to the low-resorced language in the perspective of natural language processing.

Finally, we want to prepare dataset for the Tigrinya hate speech detection which is used to train our model and it could also used as a benchmark and baseline for other researchers for further studies.

1.3. Statement of the Problem

Facebook is among the most used and well-liked social media platforms, as is widely known, and it has become a potent tool for communication and networking. The Facebook platform makes it simple to share several forms of helpful information with its users, including text, images, and video. Unfortunately, hate speech against an individual or a group of individuals based on their sex, color, ethnicity, religion, or other traits is being transmitted using social media and Facebook is one of the social networks. The rapid adoption of mobile phones and internet access is the reason why information reaches users rapidly. Because of this, hate speech on social media may become more prevalent and incite more violence, which can culminate in actual criminal activity. (MercyCorpus, 2019).

In our nation, Ethiopia, conflicts have broken out over a variety of issues, including land borders, and other resources, as well as political disagreements, racial tensions, and religious extremism.

The rise of hate speech on social media has had a big impact on people attacking one another and disagreeing on various issues. (Peace, 2021).

The amount of data on social media is increasing at an exponential rate, making it extremely difficult and time-consuming to manually identify each hateful content one by one and report it to Facebook Company. In our country, in order to address and stop the proliferation of hate speeches using social media, some studies have been done for Amharic and Afan-Oromo using machine and deep learning approaches. Even though it does not completely solve the whole problem there has been a promising result in detecting hate speech in these two languages. Despite there being more than 86 diversified languages in our country Ethiopia[8], most of the hate speeches being spread are based on Amharic, Afan-Oromo, and Tigrinya. To the best of our knowledge, we didn't find single research done for the Tigrinya language to detect hate speeches from Facebook posts and comments. Therefore, we have built a model using the deep learning approach to recognize Tigrinya hate speech from Facebook posts and comments in order to preserve and safeguard Tigrinya speakers who use Facebook from being exposed to hate speeches against particular groups and individuals.

Therefore, as a statement of the problem, despite the fact that there are models developed for Amharic and Afan- Oromo languages, and even many different models for a variety of languages throughout the world to detect hate speeches from social networks, we need to develop a new model for the Tigrinya language. The reason needed to develop a new model is that the Tigrinya language has its own writing style and structure, which means the difference in language, and therefore we cannot directly use and apply the model which is built for the Amharic or Afan-Oromo language. Furthermore, we wanted to employ different algorithm techniques not previously used for Amharic or Afan-Oromo hate speech detection in order to create an efficient model that could be able to detect hate speeches for the Tigrinya languages.

1.4. Research Questions

In order to build the Tigrinya Hate Speech Detection model this research paper comprises the following research questions and throughout the entire work, we have answered the questions listed below:

[8] https://www.tomedes.com/translator-hub/languages-ethiopia

- RQ1: How to identify the data sources and prepare the dataset for the study?
- RQ2: Which feature extrction techniques give good performance for building the model?
- RQ3: Which deep learning approaches can be used to build a best performing model that can accurately predict and classify the Tigrinya texts as hate, offensive and normal speech?

1.5. Objectives of the Study

1.5.1. General Objective

The General Objectives of this study is to design and develop Tigrinya Hate Speech Detection Model for Facebook post and comments using Deep Learning Approaches.

1.5.2. Specific Objectives

In order to fulfill and achieve the general objective, the following specific objectives have been carried out.

- Reviewing various literatures regarding hate speech detection.
- Collecting data from Facebook posts and comments and pre-process it.
- Extracting the processed data using feature extraction methods.
- Training the model by selecting the appropriate algorithm for hate speech detection
- Evaluating and analyzing the performance of the model using different evaluation metrics.

1.6. Significance of the Study

This study has a lot of significance and there are different beneficiaries from it. The first beneficiary is the Facebook Company as they are trying to build a hate-free Facebook platform, our work can be a great input for their company. The second beneficiary is the Ministry of peace of Ethiopia because the widespread of hate speech has the power to incite violence and be changed to the actual crime, therefore in bringing peace and security our designed model has its own advantages to our country Ethiopia. The third beneficiaries of this study are the Facebook users, they will not be poisoned by hate speech spread out based on a different basis of characteristics towards individuals and groups. The final beneficiaries of this study are the society, they will not be exposed to the hate speech and we all know that there has been

displacement of different societies from one place to another place in Ethiopia, and the hate speeches being spread on social media by targeting a specific society has contributed and played a major role for their displacement. Therefore, the stoppage of the wide spread of hate speech written in the Tigrinya language on Facebook has its own benefits for society.

1.7. Scope and Limitation of the Study

1.7.1. Scope of the Study

The scope of this study is to design and develop a model that detects hate speeches from Facebook posts and comments which is written using Fidel of Tigrinya language. The proposed model detects the text and classifies them as hate, offensive, and normal speeches for the multi-class classification and also classifies as hate and normal for the binary classification. We all know that using the Facebook platform we can share various types of information and the information might be in the format of text, image, or video. Thus, our proposed study has only considered detecting the hate speeches which are written in a text (Fidel). Additionally, emojis and texts being written using Latin words are not considered in our study that means we only considered Tigrinya text written in Fidel script.

1.7.2. Limitation of the Study

As a limitation, there were various reasons why we did not consider detecting the hate speeches being spread using video, image, emoji, and texts written in Latin words. The first reason was, that we had a limitation of time; to accomplish this whole work in just a few months was impossible. In addition to the time constraint to achieve our specific objectives, we had also a limitation of resources and tools such as GPU to address such kinds of problems which means we had no enough tools to comprise the hates being spread using emoji, image or video and solve these whole problems.

1.8. Organization of the Study

This study is divided into many sections, each of which is divided into chapters, and every chapter comprises a detailed description of a distinct topic. Let's take a closer look at each chapter's contents.

In the first chapter [*chapter one*] different topics and subtopics are included, such as the background of the study, motivation of the study, objectives of the study, statements of the

problem, research questions, scope and limitations are the main topics being raised throughout this chapter. In the second chapter [*chapter two*] literature review and related works are addressed, and different topics and subtopics are included in this section. Under this chapter we have addressed the effects of AI in NLP, applications of NLP, machine learning and feature extraction techniques, history of hate speech and its consequences, applications of ML in hate speech detection, the Tigrinya language and its writing system in NLP, and finally related works are the main topics throughout this chapter.

In the third chapter [*chapter three*] details about methodologies and materials used throughout this study are described in detail. The topics are procedures of the research design, data collection, data preprocessing techniques, feature extraction techniques, model selection techniques, evaluation metrics, and simulation and hardware tools are the main topics addressed in this section. In the fourth section [*chapter four*], the details of the methodologies we have selected are clearly described. The sections are the proposed model architecture and the details of model architecture starting from data collection to the evaluation metrics that have been applied to our study are described in this chapter. In the fifth chapter [*chapter five*], the implementation details of our study from data preprocessing to evaluating the model are described clearly. In the sixth chapter [*chapter six*], the results of our study are clearly described, and finally, the conclusion and future works are included in this section.

CHAPTER TWO

2. Literature Review and Related Work

2.1. Chapter Overview

The main objective of this thesis study is to design and build a model that can detect and classify Facebook posts and comments as hate, offensive, and normal for the multi-class classification and hate and normal for the binary classification for the Tigrinya language. In order to make this chapter more comprehensible and easily understandable, we have classified them into different sections. The first section covers the impacts of AI on natural language processing and its applications, on the second phase some descriptions of different types of machine learning and feature extraction techniques will be discussed, and the history of hate speech and its outcome is assessed in the third phase. The fourth phase is about the application of machine learning on hate speech, then an overview of the Tigrinya language and its writing system is covered, and at the end of this chapter, some research that has been done on hate speech detection are assessed with their respective results.

2.2. Overview over Effects of AI on Natural Language Processing

The study of artificial intelligence known as "natural language processing" enables machines to comprehend spoken and written words in various natural languages with ease. As a result, machines can converse with people using human languages. Nearly everything in this century is becoming digitized, and frequently the application of natural language processing by integrating it with computers has a beneficial effect on our day-to-day activities. We all know that peoples communicate with each other using natural language in order to do business, to transfer a message, to make a point on something, or for other else purposes. Thus creating an environment in which computers or machines understand human languages has brought major positive impacts to the world, the reason behind this good impact is that machines cannot be tired of doing something, they can work the whole day and night, and they cannot make mistakes as humans do unless they are given invalid inputs.

Machine translation was the initial motivation behind the development of natural language processing in the 1950s. At the time, even though the machine translation didn't work out, the

plan was to translate the Russian-written letter into English[9]. In addition to machine translation, natural language processing is gradually being used in a variety of fields due to the prevalence of language-based artificial intelligence. These fields include hate speech detection, Chabot, sentiment analysis, text classification, text summarization, speech recognition, speaker recognition, etc. As a result, the impacts of artificial intelligence in processing natural language for various purposes in various fields are enormous.

2.2.1. Applications of Natural Language Processing

There are certainly many applications of natural language processing in use today; let's take a quick look at a few of them[10].

Machine Translation: Amongst various applications of natural language processing machine translation is one of the first applications of NLP in which it translates a specific source language to another specific target language. One popular example of machine translation is Google Translate.

Chatbot: A chatbot is a software program that combines AI technology with pre-established rules in the natural language processing field. It is primarily made to comprehend natural languages, through straightforward question-and-answer procedures, and the machine can provide the customer with sufficient information and support.

Text classification: Text classification is a field of study in which unorganized data will be categorized into specific classes. Massive amounts of texts are currently being produced on different platforms, including, YouTube, Facebook, Twitter, emails, digital libraries, and websites. Therefore, those generated texts can be used for different purposes in various areas and categorized into a specific class. For example, spam-filtering, sentiment analysis, hate speech detection, text summarization, news classification, and so on can be grouped as types of text classification.

Sentiment Analysis: Sentiment analysis is a field of natural language processing in which it manages to know the opinions of people by identifying the polarity of specific messages and classifying it as to whether positive, negative, or neutral. Sentiment analysis is a very useful

[9] https://us.k-international.com/blog/the-history-of-machine-translation/
[10] https://monkeylearn.com/blog/natural-language-processing-applications/

application of natural language processing, for example in marketing or specific aspects of business and product launch, it is very essential to measure and know whether your customers are satisfied or disappointed by your product.

Hate Speech Detection: In today's world as freedom of speech is being practiced very well, and as public speech can be mainstreamed easily using different media outlets, hate speech is being proliferated straight-forwardly without hesitation. The hate speech being disseminated is based on different characteristics such as racism, xenophobia, anti-Semitism, anti-Muslims, anti-Christians, genders, etc. Therefore, by detecting hate speech, it is possible to protect our society and the world from being exposed to hate speech, and it is possible to keep the susceptible specific group or individuals from being victims. Hence hate speech detection is also one of the applications of natural language processing, in which the model tries to detect the piece of information and classify it as hate, offensive, and normal speech. Since our study is on hate speech detection for the Tigrinya language, we will see the details of everything about hate speech detection throughout this document.

Speech Recognition: Speech recognition is a way in which the machine translates the speech being said into text, and presents it in text form. This automatic speech recognition can be very useful in different work areas, such as transcribing a movie and preparing subtitles, for reports in a meeting in which rather than writing what is being said and decided in a meeting manually, writing everything using automatic speech recognition system would ease our workload and it is more accurate.

Information Retrieval: A software program that deals with the organizing, storing, retrieving, and evaluating the information from the repositories of documents, specifically textual information, is known as information retrieval (IR). The system helps users to locate the data they need, but it does not clearly return the answers to their questions instead it provides information about the existence and whereabouts of papers that may contain the necessary data.

Generally, Natural language processing has many different types of applications besides these already mentioned, However, this would be sufficient for an overview.

2.3. Machine Learning and Feature Extraction Techniques

2.3.1. Machine Learning

Machine learning is a branch of artificial intelligence that gives computers the ability to recognize patterns, learn from their experiences with acquired data, and develop their states without being explicitly instructed or programmed. Therefore, the major goal of machine learning is to develop an autonomous system that can mimic the functioning of the human brain without the need for human interaction. There are three types of machine learning: supervised, unsupervised, and reinforcement learning[11]. In the supervised machine learning strategy, the datasets will be trained with their respective target classes or labels, allowing the algorithm to be trained on labeled data. On the other hand, unsupervised learning is the opposite of supervised learning, it focuses on training data without any labeled data, meaning the algorithm alone determines the association between multiple data points. Reinforcement learning is the other type of machine learning in which it attempts to learn from the environment by mimicking human brains. When a machine picks up new information from its environment, it tries to learn the new situation through trials and errors and gets updated with the new information.

Numerous machine learning techniques are being used in many fields, and mainlyin natural language processing. Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), Naive Bayes (NB)[12], and many others are among the most important and well-known machine learning techniques.

2.3.1.1. Deep Learning

Deep learning is a part of machine learning in which it solves complex and sophisticated problems in a way the machine learning cannot do in an easy way. The reason behind solving very sophisticated problems using deep learning better than machine learning is that, they have complex and multilayered neural networks which are modeled definitely by emulating the human brain. When compared to machine learning the deep learning approach requires less human intervention because of their highest quality in solving numerous problems. There are different types of deep learning algorithms being applied on different fields of areas for various purposes. The mainly known deep learning algorithms are Multi Layer Perceptron (MLP),

[11]https://www.javatpoint.com/machine-learning
[12] https://www.simplilearn.com/10-algorithms-machine-learning-engineers-need-to-know-article

Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN), Long-Short Term Memory, Generative Adversarial networks (GANs)[13].

2.3.2. Feature Extraction Techniques

Feature extraction is the process of converting text-level documents into numerical forms using different types of feature extraction techniques. The main reason for changing the text into numbers is, the machine can only understand numbers, therefore in natural language processing after the steps of preprocessing such as cleaning, tokenization, stop word removal, stemming, and lemmatization, then in order to feed it to the machine, the cleaned document needs to be converted to numbers using various techniques of feature extractions. There are a number of feature extraction techniques let's see a few of them as follows

- **Bag of Words**

It is one way used for extracting features of a text in which it set out the occurrence of words in different documents. This feature extraction technique worries about the occurrence of the word in the document, which means it takes no notice and pays no attention to the sequence of words, and syntactic and semantic structure of a text. For instance, let us see how a bag of words works using the following illustrations

Document 1 (D1)	Document 1 (D2)	Stop words
እግዚአብሄር አምላኽ ጽንዓቱ ይሃብካ	ኣታ ኣምላኽ ኣቤት በለናባ እንታይ ኢና ንገብሮ	ኣታ ኢና እንታይ

Table 1: Examples of Bag of Words

Terms	Document 1 (D1)	Document 2 (D2)
እግዚአብሄር	1	0
በለናባ	0	1
ንገብሮ	0	1
ኣምላኽ	1	1
ኣቤት	0	1
ይሃብካ	1	0

[13] https://www.javatpoint.com/deep-learning-algorithms

ጽንዓቱ	1	0

- **N-grams**

The N-gram is a feature extraction technique that is used to extract the concept of the text in a given document by using different ranges of word sequences. The main advantage of this N-gram model over the bag of words is that it can handle the sequence of words and tries to handle the contextual meaning of a word in a text. Let's see how n-gram works with the sentence "እግዚኣብሄር ኣምላኽ መንግስተ ሰማያት የዋርሶ", for this example if we assign a unigram model with a range of (n=1), it will store each and every word as tokens of one word and it is called a unigram model. Therefore using unigram it will be [""እግዚኣብሄር", "ኣምላኽ", "መንግስተ", "ሰማያት","የዋርሶ"]. When we make it a bigram model with a range of (n=2) it will be ["እግዚኣብሄር ኣምላኽ", "ኣምላኽ መንግስተ", "መንግስተ ሰማያት", "ሰማያት የዋርሶ"]. Hence whenever the range of n=gram increases, the model will automatically increase to handle the sequence of words in a document.

- **TF-IDF (Term Frequency-Inverse Document Frequency)**

This TF-IDF is also one of the most widely used feature extraction techniques for different studies and the intuition behind this technique is that it considers the frequency of a word occurring in a document and measures how the word is more important than the other based on their frequency of occurrences. While term frequency analyzes how frequently a word appears in the current document, inverse document frequency analyzes how helpful or infrequent the word is overall in the document.

- **Word Embedding**

Even though word embeddings are mainly classified into two categories namely, frequency-based such as TF-IDF, Bag of Words, n-grams, etc., and prediction based in which it predicts the representation of different words by considering the context of words, there are a number of prediction based embedding techniques such as Word2Vec, Doc2Vec, Fast Text, Glove, and etc. (Tela, May 2020)

The most common feature extraction method that addresses the issues and difficulties that arise during feature extraction in natural language processing is called word embedding. The main problem with the bag of words, N-grams, One-Hot Encoding, and TF-IDF is handling the

syntactic and semantic relationship of words in a sentence. The word embedding techniques have tried to tackle such challenges by creating the same representation of real-valued vectors for words in a sentence that have similar meanings. Our study mainly uses and focuses on prediction-based word embedding such as Embedding Layer, Word2Vec, and Fast Text, their details are desribed in the next chapters.

2.4. History of Hate Speeches and Its Consequences

According to (Abrha, 2019), for a very long time especially before WWII our world have been practicing and participating by taking an interest in the thoughts of hate speeches, in which peoples or individuals hates each other based on differences in race, color, political views, ethnicity, religion, gender identity, and disability. But gradually after WWII different European countries started to draft regulations about hate speeches to protect the victims.

In the world, hate speech has brought its own awful consequences, and in the early world history, hate speech has been broadcasted using different platforms. For example, the "Hitler's Anti-Semitic Policy" is propaganda that played a major role in the genocide of Jewish people. Propaganda was the main reason behind the diffusion of hate speeches against the Jewish people[14]. During that time the propaganda had been propagated using movies and newspapers. Therefore, the most terrible genocide of the Jewish people happened because of the hate speech spread out against them.

The other consequence of hate speech in our recent history is the genocide of 500 thousand of the minority people group of Tutsi which happened because of the hate speech propagated using radio against them (Museum). Furthermore, in Kenya, there has been violence because of the political parties divided themselves based on their ethnic background, as a result, local radio stations and messaging were the main tools to disseminate hate speech which resulted the people to fight each other (Museum). Similar incidents have taken place in Nigeria (Snaddon, 2017), and Egypt (Abrha, 2019), these countries have also been in conflict because of hate speech propagation using media outlets.

Also, in recent years because of the spread of online hate different hate crimes has happened as a consequence. For instance, during Trump's presidential election hate crimes has happened, and

[14] https://encyclopedia.ushmm.org/content/en/article/nazi-propaganda

assaults of terrorism happened in New Zealand. Additionally, an assault has happened in the United Kingdom of London and Manchester. Therefore, the world was practicing hate speech and facing its consequences for centuries, and it is still a big problem for the world (Sindhu Abro, Sarang Shaikh. et.al, 2020). Even though the ultimate consequence of diffusing hate speeches against specific groups or individuals is inciting violence, on the other hand, societal intolerance, distracting and poisoning thoughts of individuals and groups of people are also consequences of hate speech.

2.4.1. Hate Speech and Social Media in Ethiopia

In the very long and modern history of Ethiopia, there have been practices of hate speeches, more specifically slavery and racism were the main results of hate speeches. Yohannes Ayalew (Ayalew, 2020) claims that in Ethiopia there were derogatory acts towards certain people by labeling them as “Barya” which can be interpreted contextually as “slave” or “a person who has very dark skin, flat nose, and kinky hair”. All of these allegations and dehumanization towards a specific group of people were an apparent manifestation of hate speeches.

Recently, according to (Abrha, 2019), in Ethiopia hate speech is escalating highly, the reason behind the escalation of hate speech is the ethnicity-based formation of regional states of the country, and political differences, which create a label of “them” and “us”, blaming the current generation based on wrong did of the previous generation, and instead of condemning an individual for his wrong did, trying to generalize his faults to his ethnicity, political, and religious views which leads to violence, the lack of societal watchfulness in which our society has no awareness about the dangerous outcome of hate speech are the main reason of hate speech dissemination and escalation. In addition to that, the emergence and rapid developments of technology have brought a significant opportunity in which everyone can easily use the social media platform in a cost-effective way. As a result of using these social media platforms some irresponsible elites, political parties, activists and individuals will propagate hate speeches to society. Therefore, social media is playing a major role in escalating hate speech in Ethiopia.

(Ayalew, 2020) Asserts that in Ethiopia there has been quarrel and violence being spread towards the society. The reasons for the dissemination of the hate were the online and offline conversations using social networks such as Facebook, YouTube, and Twitter. However to tackle this problem and in order to protect the victims of hate speeches the Ethiopian government

has ratified a law about hate speech under the "Ethiopian Hate Speech and Disinformation Prevention and Suppression Proclamation", which basically tries to define hate as "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." (Surafel Getachew Tesfaye and Kula Kekeba Tune, 2020).

2.4.2. Definitions of Hate Speeches

Our model detects the text and classifies it as hate, offensive, and normal speech for the multi-class classification and hate and normal for the binary classification for the Tigrinya language, therefore let us see the definitions of hate speeches. Hate speech has been defined by different legal and academic literature, and organizations such as by Facebook Company, but there is no agreement on the definition of hate speech. The reason for no common consent on the definition is because hate speech depends on social norms, context and situation, individual and group interpretations (Zewdie Mossie and Jenq-Haur Wang, 2018). That means the definitions vary from place to place based on such criteria. However, let us put the definition written and stated by Facebook Company, Ethiopian Hate speech law, and other academic literature.

Table 2: Definitions of Hate Speeches

Articulators	***Definitions***
Facebook	"*Objectionable content that directly attacks the people based on their protected characteristics such as religious affiliation, sexual orientation, caste, sex, race, ethnicity, national origin, gender, gender identity, and serious disease or disability*." (Surafel Getachew Tesfaye and Kula Kekeba Tune, 2020).
Ethiopian Hate Speech Law	"*Ethiopian Hate Speech and Disinformation Prevention and Suppression Proclamation Page 12339 under Proclamation No. 1185 /2020 "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*." (Surafel Getachew Tesfaye and Kula Kekeba Tune, 2020).

Twitter	"*Hateful conduct: You may not promote violence against or directly attack or threaten other people based on race, ethnicity, national origin, sexual orientation, gender, gender identity, religious affiliation, age, disability, or disease*." (kanessaa, 2021)
Academic Literatures	"*A speech or any form of expression that expresses hatred against a person or a group of people because of characteristics they share or a group to which they belong*." (Zewdie Mossie and Jenq-Haur Wang, 2018).
Recent Studies	"*Speech which either promote acts of violence or creates an environments of prejudice that may eventually result in an actual violent acts against a group of people*." (Zewdie Mossie and Jenq-Haur Wang, 2018).

As we have seen from the table there are a variety of definitions about what hate speech is and the meaning of the definition is dependent on subjective interpretation. Despite all of the definitions having almost related meanings to each other, for our study we have mainly considered the definitions used by both Facebook and Ethiopian Hate Speech law.

In addition to the hate speeches, there is a speech called offensive speech which has a low degree of hate speeches. Offensive speech might happen to make a point during conversation, debates, or jokes between peoples or friends, and the speech might offend someone, but has no intentions of hate towards individuals based on different characteristics. (Surafel Getachew Tesfaye and Kula Kekeba Tune, 2020).

Finally, there is normal speech, we say a speech is normal whenever it is neither hate speech nor offensive speech rather it is normal. Therefore, our proposed model detects the text of the Tigrinya language based on both binary and multi-class classification using deep learning approaches.

2.4.3. Hate Speech vs Freedom of Speech and Expressions

There are different controversial debates among scholars regarding hate speech detection and freedom of speech and expression (Scheffler, 2015). One of the rights protected by human rights is the freedom of expression and speech. However, using hate speech in the course of exercising this right has the potential to endanger the wellness of individuals, groups of people, and the peace and security of entire countries. This has given rise to arguments for and against expression restrictions due to the unanticipated negative effects of the exercise of freedom of expression. In the perspective of freedom of expressions the major debates were because of restricting freedom of speeches, truths might not be told and will be left without being exposed, and free speech is considered as the most reliable means of discovering the 'truth', and therefore its restriction is judged as a real social loss (Mendel, 2006). Contrarily, a number of arguments are brought forward in favor of restricting free speech, concerning hate speech. Hate speech is believed to serve no purpose in society rather it has the power to incite violence and leads to genocide, and also it undermine a person's dignity (Waldron, 2012). Hate speech prohibitions are believed to be able to protect particularly minority groups and thus also protect the inclusive character of a society (Stevens, June 2012). However, in case of our country Ethiopia to tackle the problem a law prohibiting hate speech has been approved by the Ethiopian government under the "*Ethiopian Hate Speech and Disinformation Prevention and Suppression Proclamation No. 1185/ 2020*".

2.5. Applications of Machine Learning in Hate Speech Detection

It is well-known that there is a high amount of hate speech diffusion across various social networks. On Facebook, to protect against the spread of hate there was a mechanism in which a person could report to the Facebook company if she/he saw hateful content written by somebody on the platform, and the Facebook company will check the content of the text, then passes some decisions about the user whether to restrict the account or give a warning. But this process is very tiresome and it is too difficult to manually report each and every hate written on Facebook pages.

To tackle the problem, the idea of machine learning has created a better situation for detecting hateful content. Machine learning is a branch of artificial intelligence in which the model can train and learn a pattern from experience of the given data and then makes predictions when new

data is given to the model. Using machine learning algorithms, it is possible to build a model which can be able to detect hateful content from non-hate and can create a hate-free environment on Facebook and other social media networks. Machine learning is playing a major role in detecting the hateful contents of texts. Different studies have been conducted on hate speech detection using machine learning algorithms from social media networks. (Sindhu Abro, Sarang Shaikh, et.al, 2020) (kanessaa, 2021) (Zewdie Mossie and Jenq-Haur Wang, 2018).

2.5.1. Deep Learning and Hate Speech Detection

Despite hate speech from social media can be detected using a machine learning algorithm, it is also possible to detect it using deep learning approaches. Deep learning is a type of machine learning algorithm which uses a lot of neurons to learn a pattern from data. The deep learning approach draws a conclusion by analyzing the data using different algorithms in a similar way in which human draws a conclusion. From the perspective of performance, it has better achievement than the traditional machine learning. There is been a lot of interest in this field lately, different researchers have used this deep learning approach for the purpose of hate speech detection from social networks and the results are extremely surprising and full of hope for the future. (Ashwin Geet D'Sa, Irina Illina, et.al) (Jitendra Singh Malik, Guansong Pang, et.al, Feb 2022)

2.6. The Tigrinya Language

Tigrinya (also written as ***Tigrigna***) is a Semitic language. Although there are other Semitic languages in Ethiopia and other parts of the world, some scholars assert that the most widely spoken Semitic languages are Arabic, Amharic, Tigrinya, Hebrew, and Aramaic. Tigrinya is spoken in Eritrea and northern Ethiopia of Tigray regional states and uses the Ethiopic writing system which evolved from the Geez language (Regassa, 2017).

2.6.1. The Tigrinya Writing System

Tigrinya language has its own characters (Fidel), punctuation, and numbering system. Tigrinya has 35 basic symbols in seven orders, where each symbol is represented by a combination of consonants and vowels (Regassa, 2017).

2.6.1.1. Punctuation Marks of Tigrinya Language

Like many other languages, the Tigrinya language has also its own punctuation marks. A punctuation mark is a symbol in which each representation of the symbol has a specific purpose in the language. The mainly well-known and being used punctuation marks are the following:

ኣርባዕተ ነጥቢ (።)/ '***arbaete netbi*** ' **/:** is a punctuation mark used in the Tigrinya language, and it is used for the purpose of ending a sentence, it has the same purpose as a full stop in English.

ነጸላ ሰረዝ (፣)/ '***netsela serez***' **/:** used to separate a list of elements from each other, and its functionality is the same as commas functionality in English

ድርብ ሰረዝ (፤)/ '***drb serez*** ' **/፡** used to show relationships of one idea between phrases and sentences, it has similar functionality as semicolons for English.

ክልተ ነጥቢ ምስ ሰረዝ (፡-)/ '***kilte netbi ms chret*** ' **/:** used for indicating a list of elements in a sentence, it has similar functionalities as a colon in English.

ክልተ ነጥቢ (፡)/ ' ***klte netbi*** '**/:** the functionality of this symbol was to separate one word from the other, but recently instead of using this symbol, white space is being used.

ሕቶ ምልክት(?)/ '***hito mlkt*** ' **/፧** a symbol that we put at the end of a sentence to make our sentence in interrogative form.

ትእምርተ ኣንኩሮ(!)/ '***tmrte ankro***'/: this is a symbol that we put at the end of the sentence to indicate some strong emotions, feelings or emphasis on something.

2.6.2. Morphology of Tigrinya language

Like most Semitic languages are morphologically rich and complex, the Tigrinya language is also morphologically rich and complex. The reason behind its complexity is that it is highly inflectional and derivative language.

2.6.2.1. Morphological Inflection of Tigrinya Language

The inflection of a word in Tigrinya can be created by adding different affixes to the root words, therefore the root of the word can be found in various forms. There are highly inflectional parts of speech in Tigrinya languages these are the verbs, nouns, and adjectives (G/Medhin, 2018).

***Verbs*:** Tigrinya **verbs** can be inflected to different forms such as numbers, persons, gender, tense, and so on by adding affixes to the stem verb. There are different forms of verbs these are gerundive, jussive, imperative, perfective, and imperfective. For example, to create an inflected word from the verb adding affixes to the stem verb will inflect the verb. The result of the inflection can inform to *number* which is used to identify singular or plural in number, *gender* to identify the masculine from feminine, and *person* to identify whether it is first, second or third person, *tense* to indicate the time of the action (G/Medhin, 2018).

Let's see an example illustrated by (Birhanu, Automatic Text Summarizer for Tigrinya Language, 2017), Inflection of verb for gender persons, and numbers for the ደቀሰ/dks/'-sleep

Table 3: Inflections of Verb for Gender Persons, and Numbers

Gender	persons					
	First person		Second person		Third person	
	Singular	Plural	Singular	Plural	Singular	Plural
Masculine	ደቂሰ /'dekise'/ I sleep	ደቂስና /'dekisna'/ We sleep	ደቂስካ /'dekiska'/ You sleep	ደቂስኩም /'dekiskum'/ You sleep	ደቂሱ /'dekisu'/ he sleep	ደቂሶም /'dekisom'/ they sleep
Feminine	ደቂሰ /'dekise'/ I sleep	ደቂስና /'dekisna'/ We sleep	ደቂስኪ /'dekiski'/ You sleep	ደቂስክን /'dekiskn'/ You sleep	ደቂሳ /'dekisa'/ she sleep	ደቂሰን /'dekisen'/ they sleep

Adjectives: are also one part of speech in the Tigrinya language, and an adjective can be inflected into different forms, and the inflected word can indicate a number, person, degree, gender, etc. adjectives in Tigrinya can be inflected by adding morphemes such as -ኦ/'o'/, -ት/'t'/, -ቲ/'ti'/, -ኣቶም/'atom, -ኣዊ/'awi, -ኣት/'at, - ኣዊት/'awit, ክ/k-, ዜ/z- to the suffix of words (G/Medhin, 2018)

Table 4: Tigrigna Inflection of Adjectives

No.	Male	Female	Plural form	Comparative degree	Superlative degree

1	ርሱዕ / 'rsue' /	ርሱዕቲ / 'rseti' /	ርሱዓት / 'rsuat' /	ይርሳዕ / 'yrsae' /	ዝተረሰዐ / 'zterese`e' /

Nouns: Tigrinya nouns is one parts of speech in Tigrinya language in which the noun can be given to places, humans, events, places etc. therefore it can also be inflected to different morphological words, and the inflected word might show numbers, gender, possession, and definiteness, etc.

2.6.2.2. Morphological Derivation of Tigrinya Language

Derivational morphology is a sort of word construction that allows for the creation of new words from old ones that have a different meaning than the original. According to (G/Medhin, 2018) , the Tigrinya language is extremely derivational, and nouns, verbs, and adjectives are the parts of speech that have the most derivation.

Nouns: nouns can be derived using another word class using compound words by adding affixes. When we say compound word, the newly derived noun is constructed from two dissimilar words. For instance, ቤት/'*bet*'/ which means house and ፍርዲ/'*frdi*'/ which means judgement, and ቤት/'*bet*'/ and ብልዒ/'*blei*'/ which means food, those are two separate words, when they get combined together, they give derivate noun. The first example will be ቤት ፍርዲ /'*bet frdi* '/, which means court and the second one ቤት ብልዒ /'*bet blei* '/, which means court. Additionally, by adding different affixes to a word of noun another form of word can be derived from the existing one. Example adding affixes like -ነት /'*net* '/to a word ሰብ /'*seb*'/ which means human being then it will give us ሰብ + -ነት as ሰብነት /'sebnet'/ which means humanity.

Adjectives: similar to nouns, adjectives of the Tigrinya language can also be derived from nouns by adding some different morphemes to the word such as -ኣዊ/'awi, -ዊ/wi, to nouns such as: ኢትዮጵያ/'etyopya'/, ሃገር/'hager'/, and probably it will give as a derived word of adjective ኢትዮጵያዊ/ 'etyopiwi'/ and ሃገራዊ/'hagerawi'/ respectively. In addition to this it is possible to derivate from a root verb and change the form and meaning of the word. Example we can infix the vowel -ኡ/'u'/- , to the root word ውድቕ/'wdk'/, then it will be ውዱቕ/'wduk'/ which means failed or invalid.

Verbs: here verbs of Tigrinya can only be derived from verbal roots and stems. It is not possible to derive a verb from noun or adjectives of a word.

In general, the objectives of the above examples about the morphological structure of the Tigrinya language, either its inflectional or derivational properties of the language is only to show how the language is morphologically rich and complex, therefore it requires more emphasis for further studies.

2.7. The Tigrinya language in NLP

Like many other languages in the world, the Tigrinya language can also be used for natural language processing in this digital era. For this reason, various types of studies about natural language processing for the Tigrinya language have been done over a few years. But since the language is grouped as an under-resourced language, it is very challenging to get different important resources and tools easily. However, except for the hate speech detection for the Tigrinya language few of studies has been conducted by considering various fields of areas. For instance, sentiment analysis (Tela, May 2020) (G/Medhin, 2018), news classifications (Awet Fesseha, Eshete Derb Emiru et al, 2021) (BERHE, 2011), Tigrinya Part-of-Speech Tagging (Yemane Keleta Tedla , Kazuhide Yamamoto, et.al., 2016), word sense disambiguation's, machine translation (Adhanom, Dec 2019) (Zemicheal Berihu, Gebremariam Mesfin, et.al, 2020) and many more are amongst the slightly addressed fields of studies for the Tigrinya language in the fields of natural processing, and more are yet to come.

2.8. Related Works

With the development of technology and the rise of social media users in recent years, it has become easier and more convenient to spread hate speech against society. Therefore, a number of studies on the topic of detecting hate speech from social media networks have been successful in helping to solve this problem.

Although there is no a single study on the topic of detecting hate speech on social media for the Tigrinya language, there are some studies done for Amharic and Afan-Oromo locally in Ethiopia. Furthermore, there are a lot of research on this topic worldwide, so let's take a look at some of the most recent studies on detecting hate speech.

Zewde Mossie et al. (Zewdie Mossie and Jenq-Haur Wang, 2018) this paper tried to detect hate speeches from Facebook posts and comments for Amharic language. They have used Naïve Bayes (NB) and Random Forest (RF) algorithms for classification purpose, and TF-IDF and Word2Vec for the purpose of feature extraction. According to the paper the Naïve Bayes

algorithm has outperformed Random Forest using Word2Vec feature extraction with the evaluation metrics of Accuracy (0.79), ROC Score (0.83) and Area under PR (0.85).

Yonas Kenenisa Defar (Defar, 2019) these researchers have tried to detect hate speech for Amharic language from Facebook posts and comments and they have prepared two separate datasets one for binary classification and the other for ternary classification, which means in the binary classification they classified the text as normal and hate speech, but at the ternary classification they classified the hate speech in to two classes as hate speech and offensive speech which will be classified as normal, hate and offensive speech. In order to classify the data they have used three machine learning classification algorithms namely SVM, NB, RF and also they have tried to use seven feature extraction techniques such as word n-gram(unigram, bigram, trigram), combined n-gram, TF-IDF, TF-IDF combined n-grams, Word2Vec. Finally, for the binary classification the SVM with Word2Vec has performed a better classification result with F1-Score of 73% than that of NB and RF. Also, for the ternary classification the SVM with Word2Vec has outperformed the NB and RF with F1-Score of 53%.

Surafel Getachew & Kula Kekeba. (Surafel Getachew Tesfaye and Kula Kekeba Tune, 2020) In this research paper, the researchers have made an effort to detect hate speeches written in Amharic from Facebook using variants of Recurrent Neural Network. Here they have tried to use Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) for the purpose of classification. They also have used word n-grams and Word2Vec for feature extraction. As a result, the LSTM- with Word2Vec has outperformed the GRU with an accuracy of 97.9%.

Naol Bakala Defersha & Kula Kekeba Tune. (Naol Bakala Defersha and Kula Kekeba Tune, 2021) These researcher's intention was to develop a model which detects hate speech for Afan-Oromo language from Facebook and Twitter, and they have made an attempt to use six machine learning classification algorithms such as Linear Support Vector Machine, Multinomial Naïve Bayes, Random Forest, Logistic Regression, SVM and Decision Tree. They have used bigram with a combination of TF-IDF for the purpose of feature extraction. Generally, the Support vector machine of the Linear Support Classifier has outperformed the above-mentioned classifiers with measurement metrics of Precision 66%, recall 66%, and F-score 64%.

Lata Guta kanessaa. (kanessaa, 2021) These researchers have studied a binary classification of hate speech detection using machine learning for the case of Afan-Oromo. They have used

four different machine learning algorithms namely Linear Support Vector Machine, Logistic Regression, Random Forest, and Naïve Bayes. They have used many feature extraction techniques such as TF-IDF, N-grams, and Word2Vec. As a result, Linear Support Vector Machine with TF-IDF combined with N-grams has outperformed the other three Machine learning techniques with accuracy, F1-Score, precision, and recall of 0.96.

Baharudin Sherif Kemal. (Kemal, 2021) These researchers have studied bilingual hate speech detection and applied binary classification for Afaan Oromo and Amahric languages. They have used six different deep learning approaches of LSTM, GRU, Bi-LSTM, Bi-GRU, BiLSTM+Attention and Bi-GRU+Attention mechanism using word2vec feature extraction techniques. As a result, The BiLSTM model has outperformed the rest of deep learning approaches listed above with an accuracy of of 94.3% and f1-score of 94.3%.

Pradeep Kumar Roy et.al (Pradeep Kumar Roy,Asis Kumar Tripathy et.al, November 20, 2020.) This paper has used both machine learning and deep learning approaches to build a model that can detect and classify tweeted comments as hate speech or non-hate speech for English. The researchers have used different baseline models such as Logistic Regression (LR), Naïve Bayes (NB), Random Forest (RF), Support Vector |Machine (SVM), K-Nearest Neighbor (KNN), Gradient Boosting (GB), and Decision Trees (DT) with TF-IDF and different combinations of n-grams for the purpose of feature extraction. Therefore, from the whole listed above traditional machine learning approaches the SVM has achieved a better performance to detect hate speech than the other models with a Precision of 0.54. On the other hand, the researcher has also used deep learning-based models such as Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and a combination of LSTM and CNN (L-CNN) with Glove embedding techniques for vectorization. Therefore, from the deep learning-based models, CNN has achieved better performance with a recall value of 0.88 for hate and 0.99 for non-hate speeches.

Raghad Alshalan and Hend Al-Khalifa (Raghad Alshalan and Hend Al-Khalifa, 2020) in this paper the researchers have made an effort to detect an Arabic tweets of hate speeches from twitter. They have used deep learning approaches such as Convolutional Neural Network (CNN), Gated Recurrent Unit of RNN (GRU), and combinations of both CNN and GRU by using Word2vec model of Continuous Bag of Words (CBOW). They have done a lot of

comparisons, first they compared the results provided by CNN, GRU, and combined CNN and GRU, and here the CNN has outperformed GRU and CNN-GRU combinations using Word2vec feature extraction methods with F1-Score of 0.79 and Area under Receiver Operation AUROC of 0.89. The researchers have also compared the result they gained with traditional machine learning algorithm called SVM and Logistic Regression (LR). Even though those traditional classifiers resulted a lesser output than that of the above deep learning approach they have showed a promising result and their difference in the result is not that much big. Finally, the researchers have tried to fine-tune their work by using Bidirectional Encoder Representation from Transformers (BERT). But the result didn't even reach with the traditional machine learning algorithms.

Dewa Ayu Nadia Taradhita and **I Ketut Gede Darma Putra.** (Putra, Dewa Ayu Nadia Taradhita and I Ketut Gede Darma, 2021)**,** these researchers have classified the hate speech for the Indonesian language on tweeter. They have used Convolutional Neural Network (CNN) as a method for classification of the hate speeches with TF-IDF for the feature extraction purpose. They performed a binary classification by dividing the dataset into hate speech and non-hate speech. During the training and validation stage, they achieved an accuracy of 90.85% and 88.34% respectively, and finally for the testing stages, they have achieved 82.5% of accuracy.

2.8.1. Summary of Related Works

The first three papers written by (Zewdie Mossie and Jenq-Haur Wang, 2018)**,** (Defar, 2019)and (Surafel Getachew Tesfaye and Kula Kekeba Tune, 2020) have detected Amharic hate speech from Facebook, they all have their own qualities. The first paper written by **Zewdie** and **Wang** has used only two machine learning algorithm called NB and RF using TF-IDF and they have achieved a better results of 0.79 accuracy than that of **Kenenisa Defar** (Defar, 2019) the better achievement and quality of **Yonas Kenenisa** from **Zewdie and Wang** is that it has tried to use three machine learning algorithms SVM, NB, and RF by comparing with seven feature extraction methods. Additionally, they also have prepared two datasets, one for binary classification and the other for ternary classification. They achieved 73% and 53% of F1-score for both binary and ternary classification respectively. Even though they have achieved lower performance on the ternary classification their best quality is their usage of more than two machine learning algorithms and more than six feature extraction methods. But the third one which is written by **Surafel Getachew** and **Kula Kekeba** (Surafel Getachew Tesfaye and Kula

Kekeba Tune, 2020) have used LSTM with Word2vec and achieved a better result of 97.9 % accuracy. Generally, from the three papers which are used to detect Amharic hate speeches from Facebook the one which used deep learning approach has succeeded and achieved an effective result than the rest two papers.

These two papers of **Naol Bakala Defersha, Kula Kekeba Tune** (Naol Bakala Defersha and Kula Kekeba Tune, 2021) and **Lata Guta kanessaa** (kanessaa, 2021)**,** Have also studied on detection of Afan-Oromo hate speeches from Facebook posts and comments. The first paper done by (Naol Bakala Defersha and Kula Kekeba Tune, 2021), they have used more than five machine learning algorithms using bigram combined with TF-IDF feature extraction methods. Even though they have tried to use these all machine learning algorithms which is a good part of the research but they didn't achieve an efficient result. They scored 66% of precision, 66% of recall and 64%. of F-score. The second paper of Afan-Oromo hate speech detection from Facebook is done by (kanessaa, 2021). The researchers have compared four different machine learning algorithms with more than three different feature extraction techniques. He has achieved good performance with an accuracy of 0.96 for binary classification. In addition to this **Baharudin Sherif Kemal.** (Kemal, 2021) Has also studied bilingual hate speech detection for both Afaan Oromo and Amharic languages. The best quality of this paper is that, the researchers have used six different deep learning approaches including the attention mechanisms, and also, they have used word2vec for the purpose of feature extraction. As a result, the Bi-LSTM model has achieved a better result than the rest five deep learning algorithms with an accuracy of 94.3%.

The paper written by (Pradeep Kumar Roy,Asis Kumar Tripathy et.al, November 20, 2020.), for detecting hate and non-hate speeches of tweeted comments for English language. The prime quality of this paper is that it comprises seven baseline models with TF-IDF and combinations of n-grams for feature representations. Comparing more models with each other will enable us to choose the best performing algorithm form the rest. From the baselines the SVM has performed well than the other traditional machine learning algorithm. Additionally, this study has also applied three different deep learning models with Glove word embedding techniques, and CNN has performed better than LSTM and L-CNN.

The downside of this study is that during the preparation of the dataset they have used an imbalanced data whenever they categorize the data as hate and non-hate speeches that means from the total of 31962 tweets the 29720 which means 92.98% are non-hate and the rest of 2242 tweets (7.02%) are hate speeches. Which means this study has the issue of data misbalancing therefore model bias could occur easily

Raghad Alshalan and **Hend Al-Khalifa** (Al-Khalifa, 2020) and **Dewa Ayu Nadia Taradhita** and **I Ketut Gede Darma Putra** (Putra, Dewa Ayu Nadia Taradhita and I Ketut Gede Darma, 2021) have tried to detect hate speeches from tweeter for Arabic and Indonesian language respectively. For the Arabic hate speech detection, they have used various types of deep learning approaches such as CNN, GRU and CNN and GRU combined together with CBOW of Word2vec of word embedding techniques. They have achieved F1-score of 0.79 and AUROC of 0.89. The best quality of this research is, they have compared the result achieved using different deep learning approaches, traditional machine learning algorithm, and according to the result those machine learning algorithms have achieved a lesser result. The other best thing of this research of Arabic hate speech detection is their attempt to fine-tune their work using BERT, even though they have achieved a lesser result than the traditional machine learning algorithm their effort was inspiring. On the other hand **Dewa Ayu Nadia Taradhita** and **I Ketut Gede Darma Putra** (Putra, Dewa Ayu Nadia Taradhita and I Ketut Gede Darma, 2021) have made an effort to detect hate speech from tweeter for Indonesian language. They have used CNN of deep learning approach to classify the text as binary classification of hate and non-hate using TF-IDF of feature extraction. They achieved 82.5% of accuracy. The best quality of this research paper is, they only used a very few datasets and attained a better accuracy. Additionally, they only used one feature extraction methods which is frequency based. If they had tried to use some other methods such as sequence-based word embedding techniques for further comparison they might have achieved a great result.

Table 5 : Summary of Related Works

No	***Title***	***Author***	***Objectives***	***Algorithms & Accuracy***	***Gap***	***year***
1	Social network hate speech detection for Amharic language	(Zewdie Mossie and Jenq-Haur Wang, 2018)	Detecting and classifying whether a text is hate or no-hate	Naïve Bayes using Word2Vec. Accuracy 0.79, ROC Score 0.83 and Area under PR.85.	They have applied only binary classification; additionally, deep learning approach might bring a better result.	2018
2	Hate Speech Detection for Amharic Language on Social Media Using Machine Learning Techniques	(Defar, 2019)	Classifying the text as hate and non-hate speech as binary class and hate, non-hate and offensive speech as ternary classification	SVM with Word2Vec. 73% and 53% of F1-socre for binary and ternary classification respectively	They have used traditional machine learning, they would have brought a better performance using deep learning approach, and the result achieved is not good enough	2019
3	Automated Amharic Hate speech Posts and Comments Detection	(Surafel Getachew Tesfaye and	Classifying the text as hate and free	LSTM with Word2vec. Accuracy of 97.9%.	They got an accuracy of 97.9 which is almost perfect, but it is only for binary classification	2020

	Model using Recurrent Neural Network	Kula Kekeba Tune, 2020)				
4	Detection of Hate Speech Text in Afan Oromo Social Media using Machine Learning Approach	(Naol Bakala Defersha and Kula Kekeba Tune, 2021)	Classifying Afan-Oromo texts as hate and normal	LSVM with bigram combined with TF-IDF Precision 66%, recall66%, and F-score 64%.	Using the traditional machine learning they didn't achieve good performances.	2021
5	Hate Speech Detection Framework from Social Media Content: The Case of Afaan Oromoo Language	(kanessaa, 2021)	Classifying Afan-Oromo texts as hate and neutral	LSVM with TF-IDF combined with N-grams has achieved an accuracy of 96 %.	Using the traditional machine learning they only trained the model for binary classification	2021
6	Bilingual Social Media Text Hate Speech Detection For Afaan Oromo and Amharic Languages Using Deep Learning	(Kemal, 2021)	Detecting and classifying whether a text is hate or free for both Afaan Oromo and Amaharic	Bi-LSTM with Word2vec achieved an accuracy of 94.3%, and F1-Score of 94.2%.	The only gap we did see is, they only have applied for binary classification. but their effort is great	2021

7	A Deep Learning Approach for Automatic Hate Speech Detection in the Saudi Twittersphere	(Al-Khalifa, 2020)	Classifying Arabic tweets as hate and non-hate speeches	CNN using Word2vec. F1-score of 0.79 and AUROC of 0.89	They only tried for binary classification they didn't differentiate hate from offensive speech.	2020
8	Hate Speech Classification in Indonesian Language Tweets by Using Convolutional Neural Network	(Putra, Dewa Ayu Nadia Taradhita and I Ketut Gede Darma, 2021)	Classifying Indonesian tweets as hate and non-hate speeches	CNN with TF-IDF. Accuracy of 82.5%	It only classifies as binary classification, and it only uses one feature extraction method.	2021

Generally different studies have been conducted wordwide about developing hate speech detections using machine and deep learning algorithms, but we mainly considered on the studies that has been conducted recently in our country Ethiopia.

CHAPTER THREE

3. Research Methodology and Tools

3.1. Chapter Overview

The main objective of this study is to create and develop a model that can detect Tigrinya written texts and categorize them as hate, normal, or offensive speech for multi-class classification and normal and hate for binary classification accordingly. We have so employed a variety of methodologies, tools, and techniques in the development of this model. Therefore, in this chapter we have covered the specifics of each tool and methodology utilized in building the Tigrinya hate speech detection model.

3.2. Procedures of the Research Design

In this study we have followed different procedures to achieve our goal. The following are the steps and procedures we followed:

- **Problem Identification:** first we identified a problem area, because before starting to solve something, first the problem should be identified, hence our first procedure was reading different papers and identifying a problem.
- **Paper Review:** the next procedure is to read and review different papers written and studied about hate speech detection. The reason behind reading and reviewing various papers about hate speech is to find limitation and gaps.
- **Formulation of Research Question:** on this stage we formulated research questions that fits to our objectives which later have been answered throughout the study.
- **Data Collection:** after preparing a research question, we have collected the data from different Facebook pages of posts and comments written in Tigrinya language.
- **Defining and Designing the Proposed Model:** by considering the formulated research question we have defined and designed the proposed model.
- **Implementation:** after designing the proposed model, we applied the implementation based on the proposed model, and at this step the Tigrinya hate speech detection model is developed accordingly.

- **Performance Evaluation:** once after we have designed and developed the model, we have evaluated the performance of the model at this step by using different evaluation metrics and techniques.
- **Results and Discussions:** finally, after evaluating the performance of the model, the results are discussed and explained clearly.

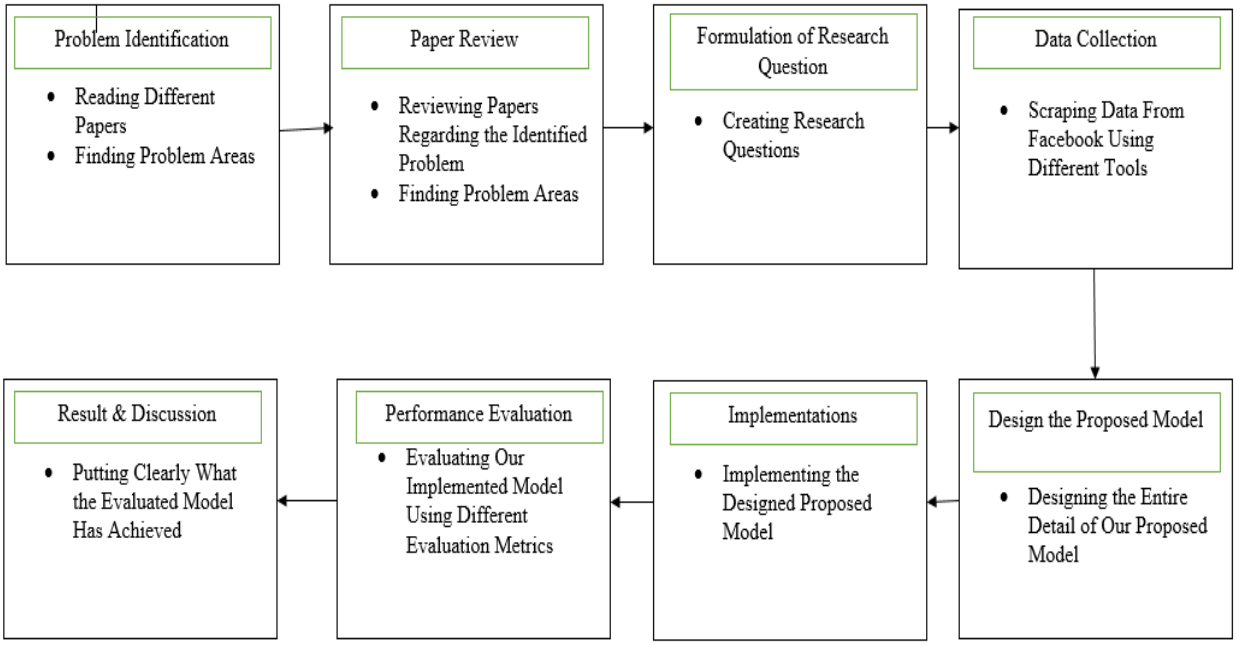


Figure 1: Procedure of the Research Design

3.3. Data collection

In order to build a model for the Tigrinya hate speech detection, first we need a data, without data it is impossible to talk about training a machine about detecting the hate text from the offensive and normal texts, and therefore we have collected the required data from various Facebook pages using different sampling techniques. These sampling techniques includes putting some criteria about the number of followers a specific Facebook page should have, and the languages used to post and write a comment on the Facebook pages are our main sampling techniques. For the purpose of collecting data from the Facebook pages we have used a tool called Facepager API. This tool includes a Facebook API graphs and it can easily collect Facebook posts and comments using different mechanisms. After collecting the data, we have applied dataset labeling using different mechanisms based on the Facebook community standard guide lines and Ethiopian Hate speech law, the detail are described in the next chapter.

3.4. Data Preprocessing Techniques

In machine learning data preprocessing is the main step, there are some preprocessing techniques need to be applied to our collected and labelled dataset. The reason we preprocess our data is to remove the data which has no value for detecting the Tigrinya hate speech, and when we do so, we will increase the efficiency of our model because it won't waste its time on unimportant texts, and it will only consider on the main texts which could have a major effect on identifying the hate, offensive and normal texts of the Tigrinya language. Therefore, the preprocessing stage includes data cleaning, normalizations, tokenization's, and stop word removal. The data cleaning comprises of cleaning and removing unwanted characters, numbers, punctuation marks and etc. on the normalization stages, to reduce the complexity for our model we have normalized different Tigrinya characters which have the same sounds but different representation into one specific characters. On the tokenization stages we have tokenized our sentence into word-level tokenization using different mechanisms, and we have also removed the Tigrinya stop words which have no meaning whenever they stand on their own.

3.5. Feature Extraction Techniques

The technique of turning text into numbers to establish an easy channel of communication with machines is known as feature extraction. Machines can only understand and sense numbers, so we must transform words into numbers in order for them to be understood and used by machines. Because of this, we used three separate word embedding approaches for this process: Word2vec, Fast Text, and Keras Embedding layer. Two different word embedding methods exist. These include prediction-based word embedding techniques as well as frequency-based embedding techniques, and we have chosen the prediction-based word embedding for our proposed model. Word embedding is one way of converting each word-level text into real-valued vectors of numbers, and similar words will be represented with similar vector spaces of real-valued numbers using different dimensions. The details of the word embedding layer we have applied to our model is described in the next chapter.

3.6. Model Selection Techniques

A model in the machine learning and deep learning domains takes the numeric format of the dataset and attempts to learn certain patterns from the data in order to categorize each input data into the desired classes. The fundamental purpose of this study is to develop a model for

classifying Tigrinya written texts as hate, offensive, or normal for multi-class classification and normal and offensive for binary classification. To achieve our goal, we employed the supervised machine learning approach to train our model on the provided dataset.

To obtain our goal we have used deep learning approaches and, in the concept, and notion of deep learning techniques, there are three different layers, these are input layer, hidden layer, and output layer. The input layer is the beginning of the whole work in the network of deep learning and is used to provide the initial data for further operations in the network. The input layer is made up of artificial neural networks with no calculated weighted input and only random weights, hence its major goal is to provide the very first input to the following layer, called the hidden layer. The neurons within the hidden layers have their own weighted inputs; the hidden layer is located between the input and output layers. These layers' main functions are to accept a weighted set of inputs and generate an output using an activation function. The output layer is located at the end of the neural network. As the name implies, it is utilized to generate the neural network's output after iterative calculations in the input and hidden layers[15]. The following diagram depicts how neural network works and our Tigrinya hate speech detection model detects and classifies the sentence "ፀርፊ ገዲፍና ብሓሳብ ንረዳዳእ" as normal and therefore it classifies and predicts each sentence using neural network.

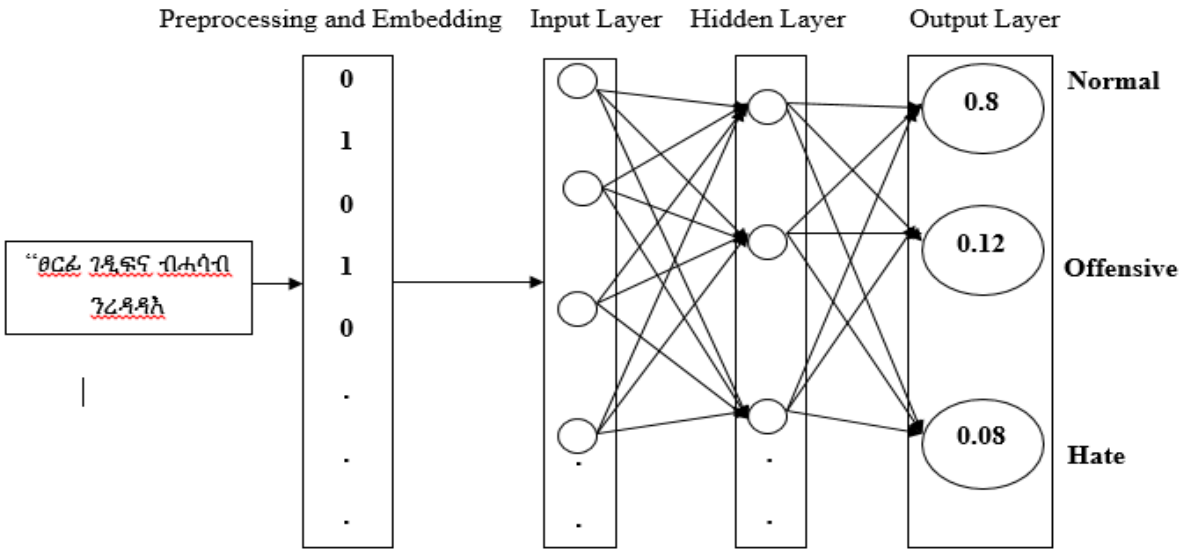


Figure 2: The General Architecture of Classification Using the Neural Networks

[15] https://www.techopedia.com/

Generally, for our proposed model we have used three different deep learning approaches these are Bidirectional Long-Short Term Memory (Bi-LSTM) which is variants of recurrent neural network and also, we have combined the Long-Short Term Memory with CNN. Additionally, for the Tigrinya hate speech detection we have also used the Convolutional Neural Networks as our model to classify the Tigrinya written texts with respect to the three different classes.

3.7. Evaluation Metrics

A trained model on the dataset must be evaluated using a number of evaluation metrics in order to determine whether the trained model is performing as planned. Evaluation metrics are used to measure a model's performance in order to determine whether or not it is working properly. Therefore, for the evaluation of our Tigrinya hate speech detection model we have used different evaluation metrics, these evaluation metrics are confusion matrix, accuracy, precision, recall, and F1-Score. These evaluation metrics are mostly used for measuring classification problems, and their detail are described in the next chapter.

3.8. Simulation Tools

In order to achieve our goal, we have used various types of tools accordingly. different simulation tools have been applied throughout the study, such as to represent the general architecture of the proposed work and to visualize every detail of it diagrammatically, additionally to implement and visualize the model for the proposed work we have also used different implementation tools.

3.8.1. Designing Tools

To visualize different .design of the proposed work we have used a software tool called Wondershare Edrawmax[16], using this software tool we have tried to depict graphically about the procedure of data collecting and labeling it, additionally, we have used this tool to design the general architecture of the proposed model. The architecture for the three different deep learning approaches such as Bi-LSTM, CNN-LSTM, and CNN has been designed using this Edrawshare software tools.

[16] https://www.edrawsoft.com/edraw-max

3.8.2. Implementation Tools

To implement and build a model for the proposed work different implementation tools and frameworks have been used. For the task of implementation purpose, Integrated Development Environment (IDE), programming languages, frameworks and hardware tools are the main tools and materials used.

3.8.2.1. Integrated Development Environment (IDE)

- **Juypter Notebook 6.3.0**[17]

Jupyter notebook is an integrated development environment and it is an open-source web application that comprises different types of python libraries. The reason we have chosen this IDE is that it consists of deep learning libraries that are useful to develop and build a model for the Tigrinya hate speech detection.

- **Anaconda 4.11.0**[18]

Anaconda Navigator is a desktop application that is used for launching different packages without directly using the command line interface. We have used this navigator to launch jupyter notebook and install different packages which are very important to build our Tigrinya hate speech detection model.

3.8.2.2. Programming Languages

- **Python 3.8.8**[19]

We have used python 3.8.8 version which is object oriented, simple to understand and high-level programming languages to build the Tigrinya hate speech detection model, and this language consists of different built-in libraries which could help to build the model.

3.8.2.3. Frameworks

We have used different frameworks to build our model, and these frameworks are very useful for our model.

- **Natural Language Toolkit (nltk) 3.6.1**[20]

[17] https://jupyter.org/
[18] https://www.anaconda.com/
[19] https://www.python.org/downloads/release/python-388/
[20] https://www.nltk.org/

This natural language toolkit is as its name indicates it is a toolkit for natural language processing, and we can use this toolkit by integrating it with the python libraries. We have applied this framework to our proposed model for different purposes in the stages of data preprocessing such as to tokenize the sentence into word-level tokenization and remove stop words from our dataset.

- **Numpy 1.20.1**

Numpy is an open source of a python library that is used to work with multi-dimensional matrices and arrays. We applied this framework while creating an embedding matrix to convert our preprocessed data into numeric form.

- **Tensorflow 2.8.0**[21]

It is an open source that enables developers to build a model and design neural networks using several layers. Therefore, we have used TensorFlow in our proposed model, and it has helped us to use different layers which are helpful in building our model.

- **Sklearn 0.24.1**[22]

The sckit-learn or sklearn is a library in machine learning algorithms that has various functionalities in classification, regression, dimensionality reduction, and clustering by using python. We have applied this framework in our work during splitting our dataset into k-folded groups of n-splits using the library of k-fold cross-validation which resides in the sckit-learn library.

- **Matplotlib 3.3.4**

For the purpose of visualizations and depicting different graphs we have used the matplotlib libraries.

3.8.2.4. Hardware Tools

In order to implement the code, the first requirement is hardware tools, without these tools, it is impossible to think about the software and implementation tools specified above, therefore to

[21] https://www.tensorflow.org/
[22] https://scikit-learn.org/

achieve our objective of designing the Tigrinya hate speech detection model we have used the following hardware tool throughout our study

To achieve our objective of designing the Tigrinya hate speech detection model we have used a personal computer which comprises different properties:

Table 6: Hardware Tools

No	***Tools***	***Specification***	***Description***
1	Hard Disk	1 TB	To store our collected dataset
2	RAM	8 GB	To increase the processing speed of our computer
3	Laptop	Core-i7 hp	To accomplish the overall work we have used hp computer

CHAPTER FOUR

4. Proposed Model of the Tigrinya Hate Speech Detection

4.1. Chapter Overview

In this chapter the detailed architecture of the proposed model and every process performed to build the Tigrinya hate speech detection model is described and explained. Therefore, in this chapter, we have divided the topics into different sections and sub-sections and tried to show how the proposed model has been built starting from collecting the dataset to evaluating and measuring the performance of the model.

4.2. Proposed Model Architectures

The proposed model architecture shows every step from the beginning to the end, it starts with data collection and preparation and then ends with model classification and evaluation. At the stages of data collection and preparation, there are various steps such as scraping the Tigrinya written texts from Facebook posts and comments, and preprocessing it which includes data cleaning, normalization, tokenization, and stop word removal. Then after the preprocessing is performed since raw texts cannot be directly inserted to the model, we have used different feature extraction techniques to change the text format into numerical form, we have applied different mechanisms for the purpose of feature extractions, such as Word2vec, Fast Text, and Keras Embedding Layer. Then since the text data is converted to numbers, we can feed the extracted data to the models directly for the purpose of classifications. In the training and classification steps, we have used different deep learning algorithms, specifically Bidirectional Long-Short Term Memory, Long-Short Term Memory combined with Convolutional Neural Network (CNN-LSTM), and Convolutional Neural Network. Hereafter the classification steps, each of the classification algorithms are evaluated using the evaluation metrics of Accuracy, F1-Score, Precision, and Recall, and additionally, we have evaluated it in terms of the confusion matrix. The following architecture describes the whole work and process of our proposed model.

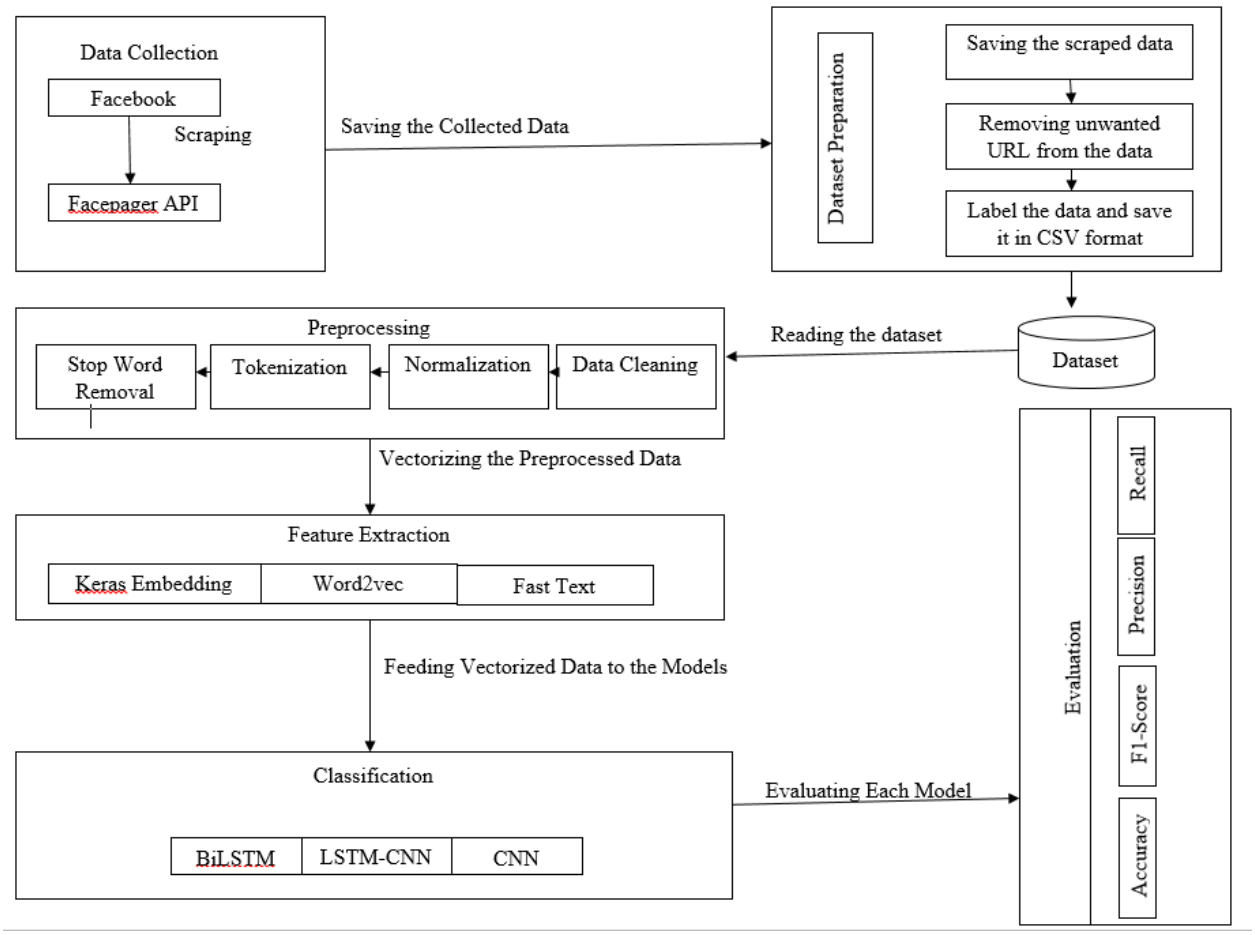


Figure 3: Proposed Models Architecture

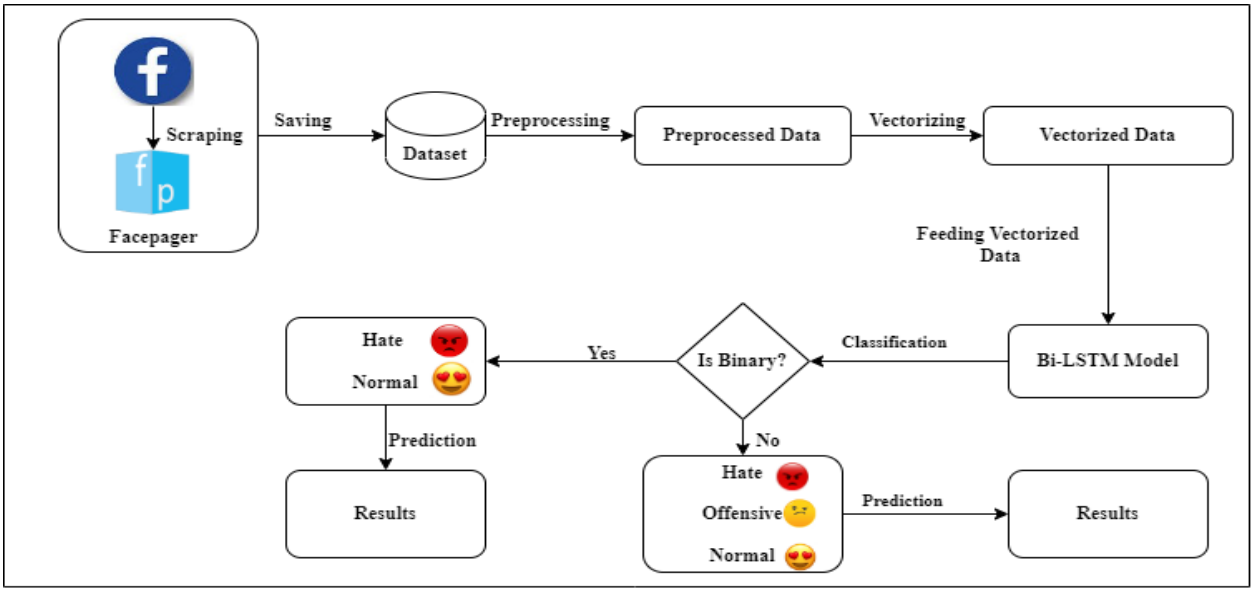


Figure 4: Tigrinya Hate Speech Detection Architecture using Bi-LSTM Model

4.2.1. Data Collection

When we come to the idea of machine learning or deep learning, it is apparent and crystal clear that the first thing that comes to our mind is data, because in the digital world without data it is impossible to achieve something. For this reason, collecting the data for the Tigrinya language hate speech detection is inevitable. As a matter of fact, since there is no research done for the Tigrinya hate speech detection, there is no dataset built for it. Therefore, we collected the dataset from scratch from various Facebook pages of posts and comments which are written using the Tigrinya language. We have used different sampling techniques to collect the data from Facebook pages. These techniques are, first, the Facebook pages need to have at least 15 thousand followers, and the reason behind the number of followers is that, as the number of followers increases, the number of ideas flowing through the page will also increase. Therefore, since the hate speech being diffused and proliferated are mostly based on religious, political views, ethnicity, race, and gender the pages that we used for collecting the data needs to be well-known activist's page, religious pages, political parties' official pages, and different public Media. The second sampling approach we used for data collection was to require that the Facebook pages from which we collect data should use Tigrinya (Fidel script) whenever they post something. For our study the main data sources to build the Tigrinya hate speech datasets were Tigray Media House (TMH), Dimtsi Woyane International (DW), center for research and documentation ማእከል ምርምርን ስነዳን ግዱሳት ሃገራውያን, Tigray Television official Facebook pages (TTV), Tigray Prosperity Party Official Pages, ኤርትራ ወይ ሞት Eritrea wey Mot and Tigray Communication Affairs Official pages were the main sources. The main reason we chose these public pages is that they have many followers and on these different pages various types of ideas such as political, religious, ethnic issues, and so on could be easily raised. In order to collect our data, we have used Facepager, which has Facebook API graphs, this tool could easily scrape any kind of posts and comments from Facebook public pages, and therefore to collect our data we have broadly used this tool.

4.2.2. The Dataset Preparation

After crawling the data from various Facebook page posts and comments, the collected data needs to be prepared and constructed. During the preparation of the dataset the main issue is annotating and labeling the data, therefore to build and prepare a neat dataset we have removed the unwanted data and brought the data together, and merged them into one file, then finally we

have annotated and labeled it as normal, hate and offensive for multi-class classification and hate and normal for binary classification accordingly and saved it in a CSV file format. Mainly the data is labeled and annotated by the researchers. And the rest of the data is annotated by the other three instructors who speak and write the Tigrinya language fluently. The annotation was performed based on the Facebook community standard guidelines and Ethiopian Hate Speech law. For the labeling process, we gave the same number of comments to be annotated by the instructors using the guidelines, and they labeled it as hate, offensive and normal accordingly.

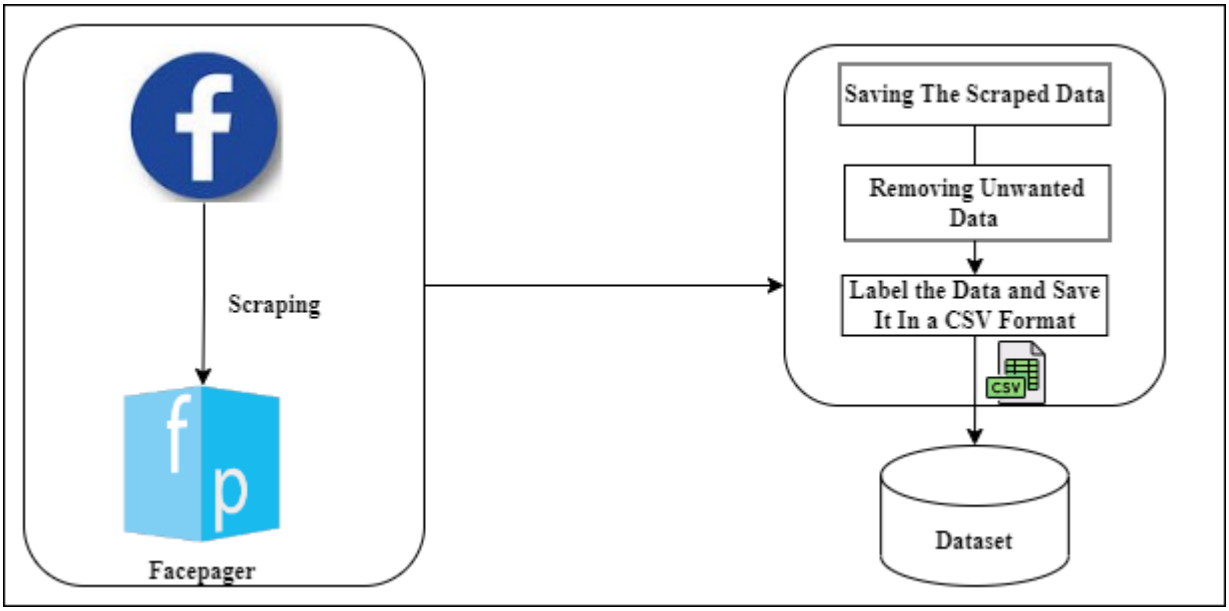


Figure 5: Procedure of Dataset Preparation

4.2.2.1. Dataset Labeling Guidelines

Even though there are no common definitions of hate speeches because of the behavior of the subjectivity of hate towards a specific society of groups or individuals, which means one speech might be hate for one group and maybe not for others. But based on the definition given by different countries and social media companies we can conclude that hate speech is a speech that targets a specific group of people or individuals based on different backgrounds such as ethnicity, political views, disability, Gender, Religion, race, and etc. Therefore, to label a specific comment as “hate” we have used the Ethiopian hate speech law and guideline of Facebook which is called the Facebook Community Standard. This community standard is a guide that describes what is allowed and is not allowed to disseminate different content towards society through the Facebook pages, and the community standard guideline covers a wide range

of policies, from these policies, hate speech is the forefront issue. Facebook defines hate speeches as "a direct attack on people based on what we call protected characteristics[23]." These protected characteristics are race, ethnicity, gender identity, religious affiliations, national origin, and serious disease or disability." Hence according to the Facebook policy of hate speech protection, our dataset is labelled as hate if the content targets an individual or group of people based on the aforementioned characteristics with:

- "Violent speech or support in written or visual form"
- "Dehumanizing speech or imagery in the form of comparisons, generalizations, or unqualified behavioral statements"
- "Mocking the concept, events, or victims of hate crimes even if no real person is depicted in an image"
- "Designated dehumanizing comparisons, generalizations, or behavioral statements"
- "Calls for segregation"
- "Explicit Exclusion which includes but is not limited to "expel" or "not allowed" "
- "Political Exclusion defined as the denial of the right to political participation"
- "Economic Exclusion defined as the denial of access to economic entitlements and limiting participation in the labor market"
- "Social Exclusion defined as including but not limited to denial of opportunity to gain access to spaces (incl. online) and social services"

4.2.2.1.1. Inter Annotator Agreements

We have given 1000 datasets to three annotators, and they have labeled each data as hate, offensive, and normal. The reason we assigned three annotators is that, if the dataset is labeled with only one rater, it would be exposed to biases the classification would be subjective and unfair. Therefore, to prevent the labeling from biases we have selected three different raters to annotate the same dataset. To ensure the reliability of the agreement between the annotators, we have used the Fleiss Kappa agreement. The Fleiss Kappa agreement is used to measure the reliability of the agreements between classifiers for multi-class classifications[24]. For the Fleiss

[23] https://transparency.fb.com/en-gb/policies/community-standards/hate-speech
[24] https://medium.com/@nick.acosta/an-introduction-to-fleiss-kappa

Kappa, there is a level of agreement for each Kappa value. The following table shows the Kappa value and its respective level of agreement.

Table 7: Fleiss Kappa Values with Its Respective Level of Agreement

Kappa Value	Level of Agreement
> 0.8	Almost perfect agreement
> 0.6	Substantial agreement
> 0.4	Moderate agreement
> 0.2	Fair agreement
> 0	Slight agreement
< 0	No Agreement

Therefore our 1000 datasets have been annotated by the three different annotators based on the guidelines of the Facebook community standards and Ethiopian hate speech law, and has resulted with **0.65** of Fleiss Kappa values, which implies the level of the inter annotator agreement is substantial level of agreements. Hence during labeling the dataset, we have reduced the level of subjectivity and biases occurs during annotation by applying the Fleiss Kappa mechanisms.

4.3. Data Preprocessing Techniques

The idea of preprocessing is the process of preparing, cleaning, and organizing the raw data in a way in which it will become more convenient to train and test it in a model using different machine learning techniques. Data preprocessing is a primary issue in building machine learning models in order to bring a better performance. First when we collect and gather our data it was unstructured and therefore it was mandatory to clean the data and removing the unwanted feature. To preprocess the raw data there are different preprocessing techniques that we have applied for hate speech detection of the Tigrinya language. From the main preprocessing techniques, we have mainly used data cleaning, text normalization, tokenization and removal of stop words.

4.3.1. Data Cleaning

When we collect the data, since it is scraped from social media, specifically Facebook pages, there are plenty of unwanted contents being crawled together with our main data. In the Tigrinya language, even though numbers are sometimes essential to describe years and some statistical data in a sentence, in hate speech detection the value of the presence of numbers in a sentence is very less, and therefore different writer uses different numbering system to list something in order while they are writing about something. Some of them uses the Arabic numbering system of [0, 9] and some of the other might use the Ge`ez numbering system [፩, ፪, ፫, ፬, ፭, ፮, ፯, ፰, ፱]. Therefore, since those numbers are less important for our study, we have cleaned numbers from our dataset. Furthermore, the Tigrinya language has punctuation marks used in its writing system, different users use those punctuations in the language such as ኣርባዕተ ነጥቢ(።)/arbaete netbi / (fullstop), ነጸላ ሰረዝ(፣)/netsela serez / (commas), ድርብ ሰረዝ(፤)/ drb serez /(semicolon), ክልተ ነጥቢ ምስ ሰረዝ (፡-)/ kilte netbi ms serez /(Colon), ክልተ ነጥቢ(፡)/ klte netbi /: (Colon) and some other general punctuations such as question mark (?), and exclamation mark (!), thus we have removed those punctuations from the dataset. In addition, these users might use emojis by mixing them with letters to express their ideas for instance they might say [ሰላም (hello) ✋✋] or [መጽለኢ (weird) 😀 😀] in the comment section of Facebook pages, therefore we have removed and cleaned those emoji's from the text to prepare more clearer dataset which is more essential for our model in which it can easily train and learn the pattern. Generally, the unwanted data that are removed from our datasets are different links, usernames, emojis, digits, special characters, extra white spaces, punctuation marks, and non-Tigrinya characters. Hence these features have been removed from the data, and removing and reducing these all feature from our data makes the dataset more accurate and precise to be trained and predicted using models of different machine learning techniques.

Table 8: Data Cleaning Example

Raw data before cleaning	Cleaned data
ሰላማት ኣሕዋት (hello brothers)✋✋✋	ሰላማት ኣሕዋት
ኣቱም ሰባት፣ 3000 ዓመት ግን ከመይ ኢና ንኑብር ዝነበርና? (Hey guys, how were we living for 3000 years?)	ሰባት ዓመት ከመይ ንኑብር ዝነበርና?

In order to perform the data cleaning process, we have used the following data cleaning algorithm.

Algorithm 4. 1: Data Cleaning Algorithm (Hunde, 2021)

INPUT: Uncleaned data

OUTPUT: cleaned dataset

START:

1: Read the dataset

2: WHILE (it is not the end of file):

IF data contains punctuations [**,**', '**.**', '"',’’’, '**:**' , '**)**', '**(**', '**-**', '**!**', '**?**', '**፡**','**፣**', '**፤**'] THEN

Remove punctuations

IF data contains special characters [’#,’**@**’,’/’, ‘\’,’**%**’,’**$**’,’**&**’,’***’] THEN

Remove special characters

IF data contains number [᎒ ᎓ ᎔ ᎕ ᎖ ᎗ ᎘ ᎙ ᎚ ᎛ ᎜ ᎝ ᎞ ᎟ Ꭰ Ꭱ Ꭲ Ꭳ Ꭴ Ꭵ Ꭶ Ꭷ Ꭸ 0 ١ 2 3 4 5 6 7 8 9] THEN

Remove number

IF data contain symbols [✋ 🔌 ❤️ 😀 😊 😊 😀 😄 😂 😂….] THEN

Remove symbols

Return cleaned data

3. **HALT**

4.3.2. Normalization of Texts

Normalization is one of preprocessing techniques that enables us to make our data simpler and clearer. In the Tigrinya language, there are variants of characters which has the same sound but, are represented with different symbols. According to (Zemicheal Berihu, Gebremariam Mesfin. et. al, 2020) in the Tigrinya language, the letters ‘ኀ’(he) and ‘ሀ’(he), ‘ሰ’(se) and ‘ሠ’ (se), ‘ጸ’(tse) and ‘ፀ’(tse) are different letters but they have the same sound. For instance, some writers might write ‘ጸማም’(tsemam) which means ‘deaf’, and others might write it as ‘ፀማም’ (tsemam) which

has the same sounds and meaning as the word ‘ጸማም’. Another example for the word ‘ሰናይ’ (senay) which means ‘good’ and others might also write this same sound and meaning as ‘ሠናይ’.

Therefore, in order to make everything easier and clearer for our model we have normalized the letter ‘ጸ’(tse) to ‘ፀ’(tse), ‘ኀ’(he) to ‘ሀ’(he), and ‘ሠ’ (se) to ‘ሰ’(se). In general, in our study, the main idea behind normalization is that if the usage of different letters didn’t bring a difference in sound and meaning using only one letter instead of many would be much better to reduce the complexity of our model.

Table 9: Tigrinya Normalization of Characters

Tigrinya letters with their respective replacement for our study						
ኀ/ሀ (he)	*ኁ/ሁ (hu)*	*ኂ/ሂ (hi)*	*ኃ/ሃ (ha)*	*ኄ/ሄ (hie)*	*ኅ/ህ (h)*	*ኆ/ሆ (ho)*
ሠ/ሰ (se)	*ሡ/ሱ (su)*	*ሢ/ሲ (si)*	*ሣ/ሳ (sa)*	*ሤ/ሴ (sie)*	*ሥ/ስ (s)*	*ሦ/ሶ (so)*
ጸ/ፀ (xe)	*ጹ/ፁ (xu)*	*ጺ/ፂ (xi)*	*ጻ/ፃ (xa)*	*ጼ/ፄ (xie)*	*ጽ/ፅ (x)*	*ጾ/ፆ (xo)*

In order to perform the normalization process we have used the following data cleaning algorithm

Algorithm 4. 2: Tigrinya Text Normalization Algorithm (Hunde, 2021)

INPUT: Non-normalized data

OUTPUT: Normalized data

START:

1: Read the dataset

2: WHILE (it is not the end of file):

IF text contains [‘ኀ’ ‘ኁ’ ‘ኂ’ ‘ኃ’ ‘ኄ’ ‘ኅ’ ‘ኆ’] THEN

Replace with corresponding [‘ሀ’ ‘ሁ’ ‘ሂ’ ‘ሃ’ ‘ሄ’ ‘ህ’ ‘ሆ’]

IF text contains [‘ሠ’ ‘ሡ’ ‘ሢ’ ‘ሣ’ ‘ሤ’ ‘ሥ’ ‘ሦ’] THEN

Replace with corresponding [‘ሰ’ ‘ሱ’ ‘ሲ’ ‘ሳ’ ‘ሴ’ ‘ስ’]

IF text contains [‘ጸ’ ‘ጹ’ ‘ጺ’ ‘ጻ’ ‘ጼ’ ‘ጽ’ ‘ጾ’] THEN

Replace with corresponding [‘ፀ’ ‘ፁ’ ‘ፂ’ ‘ፃ’ ‘ፄ’ ‘ፅ’ ‘ፆ’]

Return normalized text

3: **END**

4.3.3. Tokenization

Tokenization is the process of splitting down the sequence of strings into sentences, words, phrases, or other meaningful features or items called tokens. Sentences in the Tigrinya languages are recognized and identified by the delimiter (: :), and words are identified by comma (፣), semicolon (፤), and colon (:). But since these punctuations have been removed before the tokenization process, a white space after a sequence of characters can be used as a separator based on word tokenization. As result in our study for the purpose of tokenization, we have used word-level tokenization which separates each sequence of sentences into word-level tokens. For example, a given sentence of the Tigrinya language "እንኳዕ ድሓን መፃእኹም ኣሕዋተይ" which means "welcome brothers" can be tokenized into word level tokenization's as ['እንኳዕ', 'ድሓን', 'መፃእኹም', 'ኣሕዋተይ'].

4.3.4. Stop Word Removal

Stop words are a word in a sentence that has no meaning by themselves, which means in general stop words cannot convey a meaningful message on their own. Therefore, since the stop words have no direct or contextual meaning, we have to remove them from our document to build a more precise and accurate model. Whenever stop words are removed from the dataset, the rest of the data will only have important texts which are very important for the classification of hate speech detection. In the Tigrinya language, there are numerous lists of stop words, for instance [ኣብ/ab/, ከም/kem/, ዝኾን/zkon/, ዶ/do/, እምበር/ember/…. etc.]. as a result, since these stop words have no full meaning by their own, removing these stop words will bring positive impact to the performance of the model. Here since the Tigrinya language is an under-resourced language it has no stop word library; we have tried to build stop word lists from scratch and used some custom stop word lists used by other researchers. Despite the fact that it is a considerably smaller collection of stop word lists, we have tried to use Awet Fessha's et al. (Awet Fesseha, Eshete Derb Emiru et al, 2021) list of stop words by adding our prepared stop words list to it. Generally, we have successfully achieved the stop word removal.

The algorithm we have used for the stop word removal is the following:

Algorithm 4. 3: Algorithm for Removing Stop words (Hunde, 2021)

INPUT: word-level tokenized list of words

OUTPUT: word-level tokenized list of words without the list of stop words

Variable: Stopwords [], Stopword_cleaned []

1: Read the dataset

2: WHILE (list of words in the dataset):

    IF list of words in the dataset not in Stopwords: THEN

    Append the word in Stopword_cleaned

  Return Stopword_cleaned

3: Halt

4.4. Feature Extraction Techniques

After completing the data cleaning and preprocessing steps, before we start directly feeding the cleaned data to the model of machine learning, the data needs to be changed to numerical values. The process of changing the text to numbers is known as feature extraction. It is clear that machines can only understand numbers, therefore everything we prepared as input data, and it has to be first changed to numerical forms. To achieve this process as described in detail in the previous chapter about feature extraction techniques and its types, we have chosen prediction-based word embedding techniques for our proposed model for the purpose of extracting the features and representing each word as vector space. The reason we chose the prediction-based word embedding techniques is, it can handle the semantics of different words and tries to assign similar vector spaces for the words which have similar meanings for sequential data. Hence, since our study focuses on sequential data, it is better to use word embedding techniques.

The prediction-based word embedding comprises a lot of feature extraction techniques in it, such as Word2vec, FastText, Glove, and many more are amongst the type of words embedding techniques. Therefore for the detection of hate speech for the Tigrinya language, we have used the keras-based embedding layer, Word2vec, and FastText of word embedding's, and the reason we preferred these embedding layers is, they have better performance in Tigrinya text classification (Awet Fesseha, Eshete Derb Emiru et al, 2021). For the purpose of feature

extraction, even though there are pre-trained wiki-based word vectors of Fast Text for the Tigrinya language[25] on (Piotr Bojanowski. et.al, 2016), they were only trained on unique words of 1864, whereas our dataset contains over 19500 unique words and as a result, the wiki-based pre-trained model produced the worst results. For this reason, we have trained and prepared a custom Word2vec and Fast Text. However, as more words are vectorized in the future, everyone working on Tigrinya language of NLP will benefit from the pre-trained wiki-based vectors developed by (Piotr Bojanowski. et.al, 2016). Word2vec and FastText can be implemented in two different ways, namely using Continuous Bag of Words (CBOW) and Skip-Gram models. Let us see graphically how the prediction-based word embedding maps different words into real-valued vectors with output dimensions of 4 for the sentence "ኩሉ ሰ ብዛዕባ ሰላም ይሕሰብ", and "ኩሉ ሰብ ሓደ ኣይኮነን", this two-sentence will be converted into its respective real-valued vectors using the embedding layer as follows

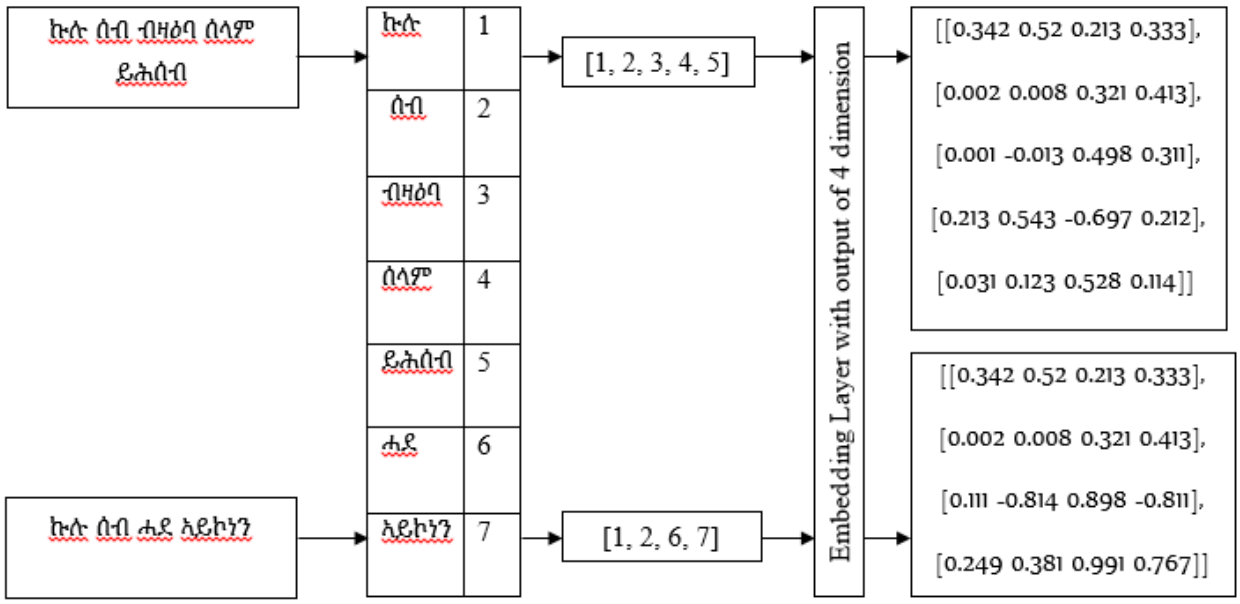


Figure 6: Illustrational Architecture for Word Embedding Techniques

4.4.1. Word2vec

Word2vec is a word embedding technique that attempts to find and identify the similarities of words and dependencies by iterating over a large prepared corpus. It finds the similarities between different words using the metrics of cosine similarities. That means if the angle of the

[25] https://fasttext.cc/docs/en/english-vectors.html

cosine is 90, then there is no similarity in context or the words are different, hence the words will get different vector representations and if the angle of the cosine is 1, then the words are mostly related and similar, therefore the words will get similar or nearly similar vector representations. Word2vec uses two variants of neural network-based models called Continuous Bag of Words (CBOW) and Skip-Gram models.

4.4.1.1. Continuous Bag of Words (CBOW)

A continuous bag of words is one way in which the Word2vec implementation is achieved. CBOW is a simple neural network that has only one input layer, one hidden layer, and one output layer with a number of neurons inside it, in which these neurons are equal to the dimensions of the vector that represents each word. This neural network-based model tries to find similarities between words by taking a number of words as input, then predicts one target or main word as an output based on the contextual words in a sentence. For instance, for the sentence "ኩሉኺም ኣሕዋተይ በብዘለኺምዎ ሰላምታይ ይብጻሕኩም" ፣ for this example let's say the target word is "ኣሕዋተይ" with a window size of 2, then the contextual pair of words to predict the target word of "ኣሕዋተይ" will be

Table 10: Example of CBOW

Sentence	Pair of words or contexts for the target words
"ኩሉኺም **ኣሕዋተይ** በብዘለኺምዎ ሰላምታይ ይብጻሕኩም"	(ኩሉኺም, **ኣሕዋተይ), (ኣሕዋተይ**, በብዘለኺምዎ), (**ኣሕዋተይ**, በብዘለኺምዎ, ሰላምታይ)

Therefore according to the above example for the output or target word of **"ኣሕዋተይ"** the contextual words with window size of 2 could be (ኩሉኺም, **ኣሕዋተይ), (ኣሕዋተይ**, በብዘለኺምዎ), (**ኣሕዋተይ**, በብዘለኺምዎ, ሰላምታይ), the architecture of CBOW is presented as follow.

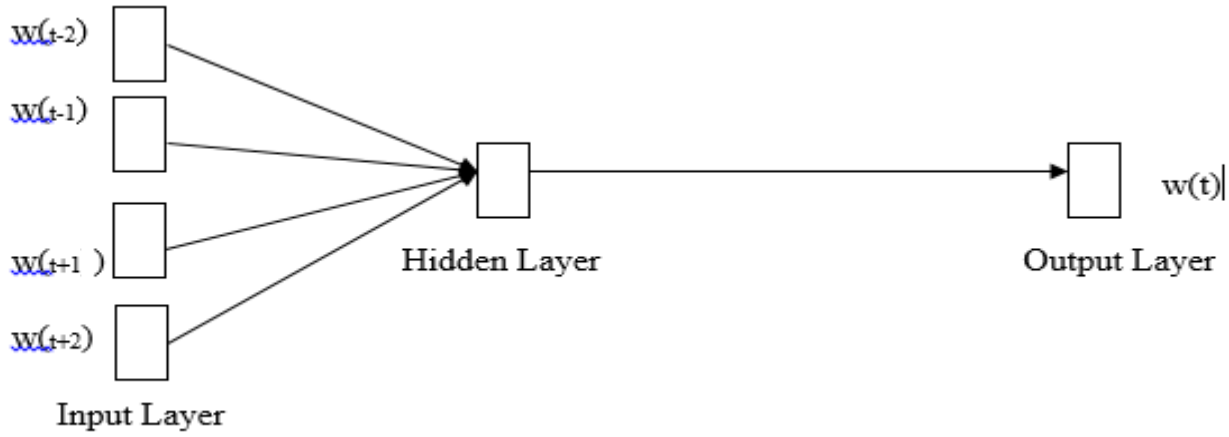


Figure 7: Architecture of CBOW (Eddy Muntana Dharma, Ford Lumban Gaol. et.al, 2022)

4.4.1.2. Skip-Gram models

It is also another way that is used for Word2vec implementation, the same as CBOW. It is also a neural network-based model, the basic difference between Skip-Gram and CBOW is that the Skip-Gram model takes the target word as an input and predicts the context of the neighboring word for the input. That means the skip-gram model performs the representation of words in the opposite way than that of a continuous bag of words, which means it predicts the contextual words based on the main or targets words.

This neural network-based model tries to find similarities between words by taking the target or main word as an input, then it predicts contextual words as an output from the current input word. For instance, for the sentence "ኩሉኹም ኣሕዋተይ በብዘለኹምዎ ሰላምታይ ይብጻሕኩም"፣ for this example let's say the main word is "ኣሕዋተይ" with a window size of 2, then the target output will be the contextual pair of (ኩሉኹም, ኣሕዋተይ), (ኣሕዋተይ, በብዘለኹምዎ), (ኣሕዋተይ, በብዘለኹምዎ, ሰላምታይ) for the current input word of "ኣሕዋተይ",the architecture of Skip-Gram is presented as follow.

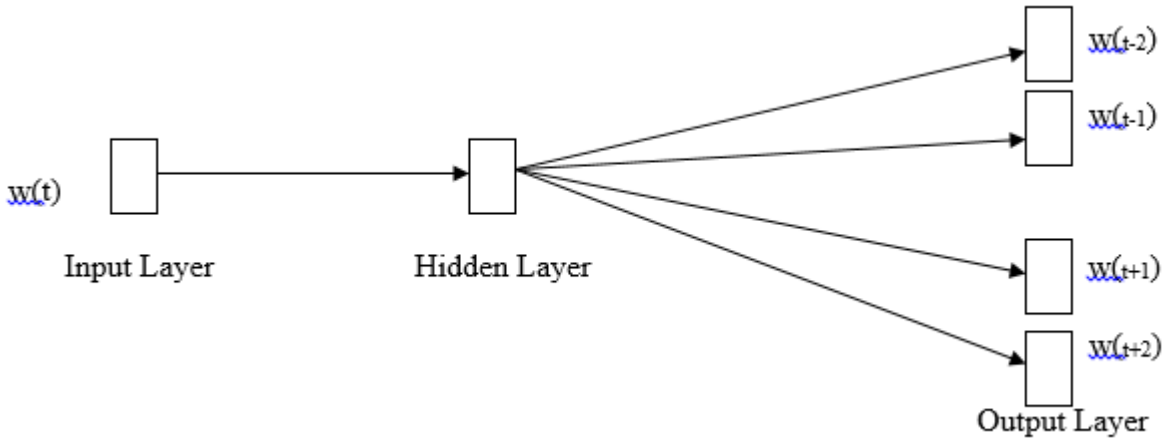


Figure 8: Architecture of Skip-Gram (Eddy Muntana Dharma, Ford Lumban Gaol. et.al, 2022)

4.4.2. Fast Text

This FastText algorithm is an extension of Word2vec and has the same functionality and purpose as Word2vec, which means it tries to detect dependencies between words and identifies their similarity to assign similar vector representations for the words. The basic thing that makes FastText different from Word2vec is that it uses character level of N-gram of word representation while it trains. But this makes it to take more processing time and increases the size of the data. Additionally, the best thing that this FastText algorithm comprises is, it supports words that are out of vocabulary (OOV). For example, if the word "ታሕጓስ(tahguas)" which means "happiness" is not in the vocabulary for the training, Word2vec will response the word "tahguas" does not exist, but in FastText even though the word is not in the vocabulary, it will assign a vector space for it, by relating it to other existing words inside the vocabulary.

Generally, this FastText algorithm is also implemented using neural network based of Continuous Bag of Words and Skip-Gram models, in the same way in which Word2vec is implemented.

4.4.3. Keras Embedding Layer

The Keras embedding layer requires the data to be in an integer-coded format, and therefore each word can be represented with a unique integer. The process of representing each word into unique integers is achieved using the Tokenizer class, which is supported by Keras[26]. Even

[26] https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/

though this Keras embedding is capable and compatible to learn different pre-trained embedding layers such as Word2Vec and FastText, it can also be part of the deep learning without using the pre-trained embedding layers, which means it can learn the relationship between words by itself and assign a real-valued vector for each word and trained with the model. Therefore, in addition to using pre-trained Word2Vec and FastText, for our study we have also used the Keras embedding layer as feature extraction mechanisms.

4.5. Model Selection Techniques

The objectives of the machine learning model are to train on specific data, recognizing and learning some patterns from the data for some specific problems. As described in detail in the previous chapter, there are four main types of machine learning algorithms, these are supervised, unsupervised, and re-enforcement learning[27]. For our study, we have used the supervised machine learning types, and the reason behind using this types of machine learning is that, hate speech is all about classification of a text and labeling it to hate, normal, or offensive, and such kinds of work falls and grouped under supervised learning.

As we have tried to describe earlier in the dataset preparation, our dataset is prepared and stored in a CSV file format. We have labeled Tigrinya's written texts based on the guideline of the community standard of Facebook and Ethiopian hate speech law as hate, offensive, and normal speech accordingly. Therefore, since our model is trained to classify and predict the Tigrinya text into two and three different classes, our criteria for the model selection are based on multi-class classification, and additionally binary classification is also applied to our model which classifies the text as hate and normal. For our proposed work we have selected a deep learning algorithm than that of traditional machine learning. The reason we selected the deep learning algorithm is that most of the results achieved for Amharic and Afan-Oromo languages using the traditional machine learning algorithms are not satisfactory, and additionally, since the Tigrinya language is morphologically rich and complex, deep learning is much better in learning complex patterns better than the conventional machine learning algorithms, and therefore we have preferred deep learning algorithms. Furthermore, the paper of (Son T. Luu, Hung P. Nguyen,

[27] https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms

et.al, Jan, 2020) and many other papers have clearly put that deep learning is a better algorithm for hate speech detection and classification than the traditional machine learning algorithm.

Deep learning is a sub-field of machine learning techniques in which it teaches the models to learn the pattern from the data by imitating it with neurons of human brains. In our proposed work, using deep learning techniques we have enabled the model to learn and classify the text as hate, offensive, and normal for multi-class classification and normal and hate for binary classification after training it using a set of labeled data and neural network architectures that contains many layers. There are different types of deep learning techniques among these Multi Layer Perceptron (MLP), Recurrent Neural Network (RNN), Long-Short term Memory (LSTM) and Convolutional Neural Network (CNN), are the main and forefront in deep learning approaches. Therefore, for our study, we have compared three different deep learning approaches, specifically Bi-LSTM, CNN-LSTM combined, and CNN. We have built the model by comparing the results of CNN, Bidirectional Long-Short Term memory (Bi-LSTM), and Long-Short Term memory (LSTM) combined with CNN. The reason we used these deep learning approaches is that it has been widely used for sequential data and some research indicates that variant of RNN and CNN are very significant in achieving better performance for hate speech detection (Surafel Getachew Tesfaye and Kula Kekeba Tune, 2020) (Auliya Rahman Isnain, Agus Sihabuddin, et.al, 2020) (Al-Khalifa, 2020) (Putra, Dewa Ayu Nadia Taradhita and I Ketut Gede Darma, 2021) .

Therefore, the steps and processes for each of our selected models is, first we preprocessed the text, then we converted it to numerical forms to enable our preprocessed data to be understood by the machine using word embedding techniques, namely Word2vec, FastText, and the Keras embedding layer. After this process we fed the vectorized data to the Bi-LSTM, CNN-LSTM combined, and CNN models. Let's see the architecture of our selected deep learning algorithms with their detailed descriptions as follow.

4.5.1. Bi-directional Long-Short Term Memory (Bi-LSTM)

Bi-directional Long-Short Term Memory is a variant of RNN which uses two layers of LSTM, in which the first layer of the LSTM takes the input in a forward direction, and the second layer of LSTM takes the input from the backward direction in order to have better information and understanding about the sequential data through the neural network. Except for the addition of

one more layer of LSTM to the neural network, the work-flow of Bi-LSTM is the same as LSTM layers, and its general architecture for our proposed work looks like the following.

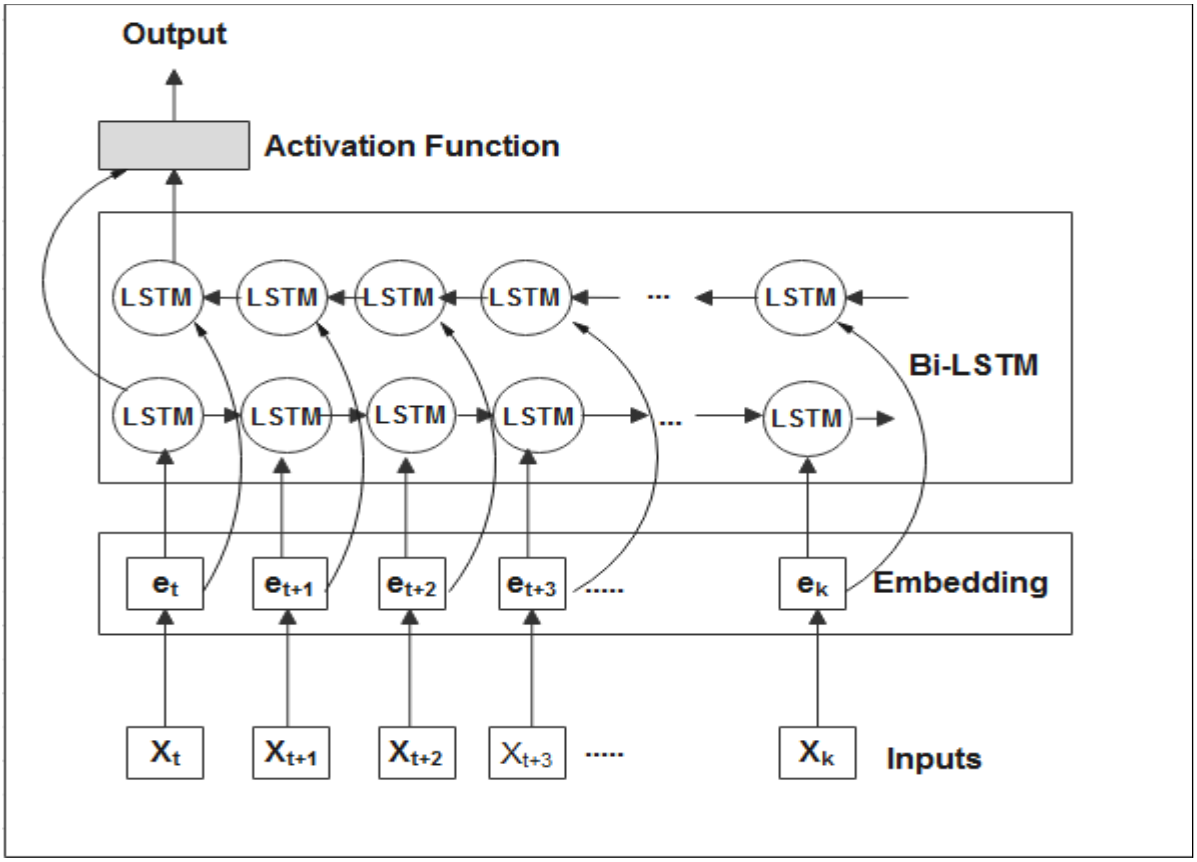


Figure 9: Architecture of Bi-LSTM (Auliya Rahman Isnain, Agus Sihabuddin, et.al, 2020)

Here we used the embedding layer as an input layer which has been used for converting the texts to numeric forms, then at the hidden layer, we have applied the Bi-directional Long Short-Term Memory which is used to represent the sentence and learn a pattern to handle the contextual meanings. Finally using the sigmoid and softmax activation function each sentence can be classified as hate, offensive, and normal for binary and multi-class classification at the output layer.

4.5.2. CNN-LSTM Combined

To achieve our Tigrinya hate speech detection we have also applied the combinations of Convolutional Neural Networks (CNN) and Long-Short Term Memory (LSTM). The reason we combined these two neural networks is, that they have been used in many studies and to take the advantages of both the LSTM and CNN and see if it has a major impact in identifying the Tigrinya hate speeches. The CNN can easily analyze a text and using the pooling layer, the CNN

could extract significant textual properties, and however using the CNN it is challenging to obtain contextual information. Therefore, to handle the problem of handling contexts the LSTM is convenient, as a result for our hate speech detection combining and integrating these two neural networks together has resulted in a good performance. The combination of LSTM and CNN in our work has the following architecture.

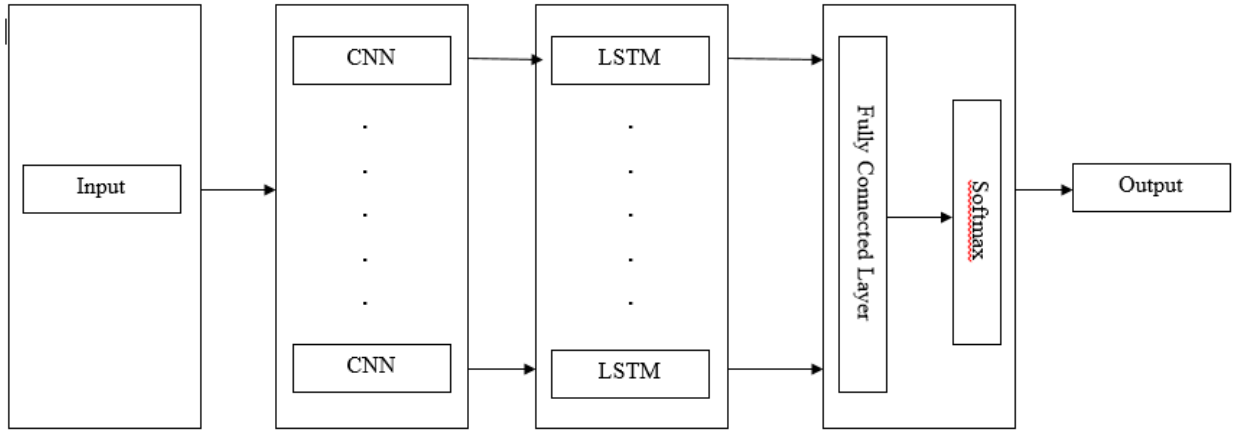


Figure 10: Architecture of CNN-LSTM for Tigrinya Hate Speech Detection

As we can see from the architecture, as an input layer, we used word embedding layers to vectorize the word-level texts in a matrix form then CNN locally extracts significant textual features using different mechanisms, then it feeds those extracted and flatten texts to the LSTM layer, then this layer learns the pattern by considering the semantic relationships between the words, finally with the help of activation function each sentence has been classified accordingly as hate, offensive and normal for multi-class classification and as hate and normal for binary classification.

4.5.3. Convolutional Neural Network (CNN)

It is well known that a convolutional neural network is mainly applicable in computer vision such as for image classification. CNN is a neural network, and it is one amongst different types of deep learning algorithms, it has almost similar layers except it has a unique layer called the convolutional layer, which mainly makes it different from other deep learning techniques. In the case of image classification, the convolutional layer slides to the whole matrix of the image using different mechanisms with a 2D or 3D convolutional layer. But in the case of text classification, since the data is sequential we need to reduce the convolutional layer to one

dimension, here the concept of the convolution layer is the same, except the dimension is changed to one dimension, and the data type is changed from image to sequential data[28]. The architecture of Convolutional Neural Network looks like the following.

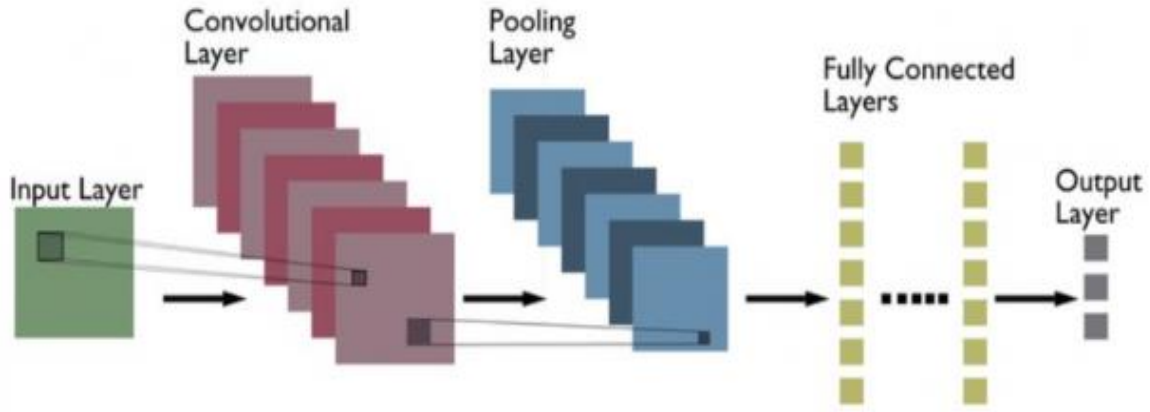


Figure 11: Architecture of Convolutional Neural Network[29]

The architecture of the convolutional neural network shows that there are different layers. The input layer is the first layer that receives data after transforming the text to vectors with embedding layers; hence, the embedding layer is the input layer, followed by a convolutional layer. The convolution layer has filters inside it, which is used to be applied to the input of text data, here the output of the convoluted layer is a feature map, in which this feature map is the representation of the input data after the filter is applied. Then there is a layer called pooling, which is also another type of convolution layer, which is used to minimize the spatial size of the input data. Inside the convolutional layer, the pooling layer is used to minimize the number of parameters, which results to help the model train faster. Basically, pooling layer is used to reduce input size before it is fed into the fully connected layer. Pooling is classified into two types: max pooling and average pooling. Max pooling takes the maximum value from the feature map, while average pooling takes the average value from each of the feature maps. The succeeding layer is the fully connected layer, in which each neuron is fully connected to every other neuron,

[28] https://analyticsindiamag.com/guide-to-text-classification-using-textcnn/
[29] https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/

and eventually, on the output layer, it classifies the text using various activation functions based on the features learned in the previous layer[30].

Therefore for our study, we have used the embedding layer as an input layer which is used to convert the text to number, at the hidden layer we have used the CNN model with its different layers and neurons, and finally using the softmax activation function for multi-class classification, and sigmoid activation function for the binary classification on the output layer.

4.6. Activation Functions and Overfitting Handling Techniques

4.6.1. Activation Functions

In the actual world, the human brain separates helpful information from non-useful information using neurons and preserves the useful one for distinct situations; the same scenario works to the activation function in a deep neural network. The activation function's major aim is to find very useful information and activate the neuron containing it, while also deactivating less useful information throughout the neural network, allowing the neural network to spend less time on less important and irrelevant data. The reason behind the deep neural network requiring activation function is, to add non-linearity to the networks. Whenever sophisticated works are wanted to be trained using neural networks, it is very difficult to identify and classify them using a linear regression model, therefore activation function will add non-linearity, which creates a more flexible environment to classify the data. There are more than ten types of activation functions being applied in the world of machine learning, some of them are Rectified Linear function (ReLU), Leaky ReLU function, Parametric ReLU function, Sigmoid function, Softmax function, tanh function (tanh), and, etc. Activation functions are applied on the hidden and output layer, and therefore for our work, we have used the ReLU activation function on the hidden layer and the sigmoid and Softmax function on the output layer. The reason why we use ReLU at the hidden layer is that, our model has provided better performance than the other activation function, and the Softmax function has been used on the output layer since it is the most common and recommended function to be applied on the multi-class classification problem. For the binary classification we have used the sigmoid function.

[30] https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/

4.6.2. Overfitting Handling Techniques

While training a model for a specific issue, there is a common problem that occurs in either machine learning or deep learning algorithm and leads the model to poor performance, which is called model overfitting and model underfitting. Model overfitting is a problem when the model performs well during training the data, and it generalizes as the model is performing well, but whenever the model is being tested with unseen data, it performs poorly[31]. On model overfitting, the model perceives and learns each and every detail of the noise during the training which later leads it to fail to generalize with the unseen data, and there are a lot of ways to handle this problem. Model overfitting can be handled using different techniques such as by getting more training data, using augmentation of the existing data, early stopping, L1, L2 Regularization techniques, Dropout, and, etc. from these techniques we have used L2 Regularization, and Dropout techniques to handle the model overfitting, and this Dropout is types of regularization techniques which is used for handling such problems, and therefore it performs better than the other overfitting handling techniques..

4.7. Evaluation Metrics

After the completion of building and training the model there is a testing and evaluation session in which the quality of our built model is measured. Therefore, there are a lot of evaluation metrics in machine learning, thus for our study we have used different evaluation metrics such as *accuracy, precision, recall, F1-score* and *confusion matrix*.

In the field of machine learning, the model must be evaluated in order to determine and ascertain whether it is doing well or not after being built using various techniques to address a particular problem. Therefore, we have applied different evaluation metrics to check whether our model is detecting well the hate speeches of the Tigrinya written texts which are trained based on the K-Fold Cross-Validation technique. Therefore, for our study, we have used different evaluation metrics such as accuracy, precision, recall, F1-score and confusion matrix to check and evaluate how good or poor our model is from the perspective of various evaluation metrics

Confusion Matrix: is a method that is used for summarizing the performance of classification algorithms. We can acquire a better understanding of the classification model's successes and

[31] www.ibm.com/cloud/learn/overfitting

failures by calculating a confusion matrix. As only evaluating a model's accuracy cannot provide specific information about the classifications, by applying the confusion matrix, we can quickly determine the correctly and wrongly classified number of texts.

Accuracy: this evaluation metric can be applied in classification problems, and it calculates the ratio of the correct number of predictions to the whole number of total predictions being predicted by the model. It can be expressed as

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{total\ number\ of\ preditions\ made}$$ Or can be expressed as

$$Accuracy = \frac{TN+TP}{TN+TP+FP+FN}$$ Where

TN, stands for true negative, TP, true positive, FP, false positive, and FN, false negative.

Precision: this evaluation metric is very useful whenever the classification of the test dataset into different classes is unbalanced and skewed. This model performance evaluation technique works in a way in which the ratio of the true positives to the all positives being predicted by the model, and can be expressed as

$$precision = \frac{TP}{TP + FP}$$

Recall: It is used to evaluate the models' capacity to recognize positive samples; thus, the more false negatives the model predicts, the lower the recall. It is expressed as the ratio of true positives to all positives in the dataset, and it can be expressed as

$$Recall = \frac{TP}{TP + FN}$$

F1-Score: this evaluation metric combines both the precision and the recall evaluation metrics together. Whenever the value of the F1-Score becomes higher, it shows that the model is performing well, but the F1-Score gets a higher value whenever both recall and precision are high.

$$\mathrm{F1-Score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

CHAPTER FIVE

5. Design Implementation of the Proposed Model

5.1. Chapter Overview

In this chapter, some details about the implementation of the proposed model are described in detail. It tries to show each and every implementation detail used to develop the Tigrinya hate speech detection. It shows the implementation detail starting from data preprocessing to model evaluations by dividing them into different sections.

5.2. Implementation Flowchart

In building a model for the Tigrinya hate speech detection we have followed a lot of implementation steps, which means starting from reading the file and then preprocessing it, which includes different steps, then text vectorization using different feature extraction techniques, and then model selection and training it, finally testing and saving the best performing model. This whole process is achieved using different implementation procedures accordingly.

- **Reading the CSV File**

We have loaded and read our CSV format dataset, therefore to read this file we have used panda's data frame which is provided by the python library in the Jupyter notebook.

- **Data Preprocessing**

After the data is read, it needs to be preprocessed which includes data cleaning, normalization, tokenization, and removing the stop words. We have achieved this implementation process using different libraries provided by python.

- **Vectorization**

In order to be understood by the machine, since the text needs to be vectorized, we have changed the text into numbers using different implementation techniques such as Word2vec, Fast Text, and Keras Embedding layer, which these techniques are provided by the gensim library and Keras library.

- **Model Selection**

For the purpose of hate speech detection for the Tigrinya language, we have used three different deep learning techniques, namely Bi-LSTM, CNN-LSTM Combined, and CNN, and therefore we have implemented each of the above deep learning algorithms accordingly.

- **Train the Model**

To train our model we have used k-fold cross-validation techniques to split our dataset into train and test sets. There are various types of cross-validations, but to train and test the model we have used stratified cross-validation. We have preferred the stratified cross-validation because, it takes into account observing all the classes in the dataset equally, and split them into k groups. The model trains on the k-1 groups and gets tested on the one folded group which is been left from the training set for the purpose of testing the model to check how it has learned from the k-1 folded groups. It trains on each folded group iteratively and this process helps the model to learn easily and to be less biased to predict.

We have trained our model using the vectorized Tigrinya dataset by using different algorithms, therefore during training the model, we have applied different implementation libraries including the deep learning algorithms and cross-validation techniques.

- **Evaluate the Model**

To train our model we have used the k-fold cross validation techniques in which the model learns a pattern from the dataset by splitting them into train and test classes by creating the K-folded groups. Even though the model has trained itself by such a mechanism, it has to be evaluated using various evaluation metrics to determine how well it is performing. For this purpose, we have used different evaluation metrics to check whether our k-fold cross-validation-based trained model is performing well in terms of accuracy, precision, recall, and F1-Score. Additionally, we have used the confusion matrix to clearly put the correctly predicted classes and the wrongly predicted classes in our model.

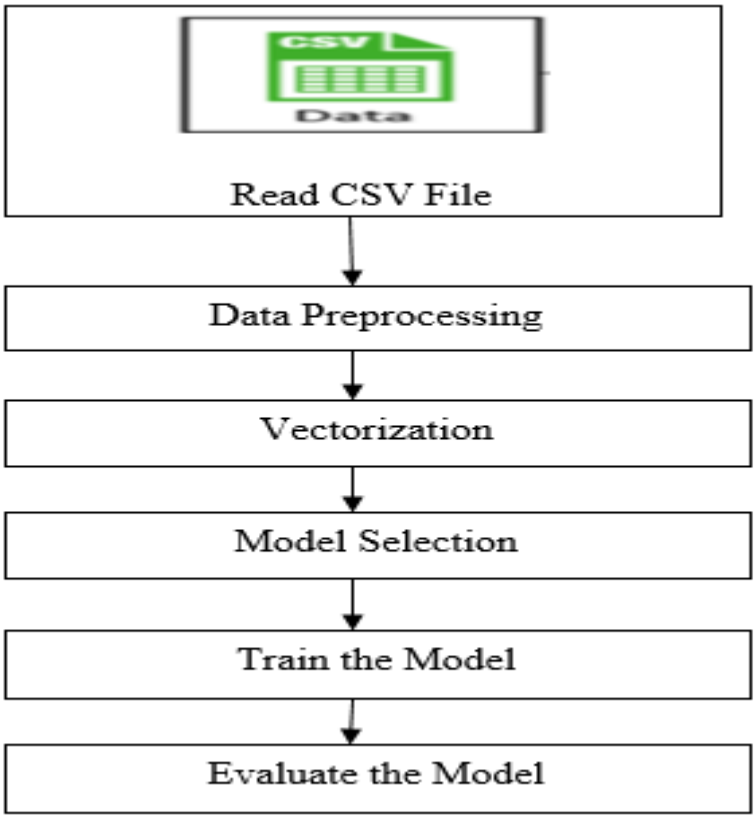


Figure 12: Implementation Architecture Flowchart

5.3. Loading and Reading the dataset

Before preprocessing, or training the model with the data, it has to be loaded to the environment first, and therefore we have loaded and read the CSV file format dataset using Panda data frames as follows:

Code 5. 1: Implementation for Loading the Dataset

```
# Loading and Reading the dataset
import pandas as pd
df=pd. read_csv ("Tigrinya_HateSpeech_Detection.csv", encoding='utf-8')
df
```

5.4. Implementation for Data Preprocessing

As we have discussed, data preprocessing is very essential in machine learning which helps the model to focus only on the main data, different unwanted data need to be preprocessed and removed before feeding it to the model, and therefore at this step, we have implemented different

preprocessing techniques, such as data cleaning, normalization, tokenization, and removing stop words are amongst the main part of preprocessing in our study.

5.4.1. Data Cleaning Implementation

There are various texts, numbers, punctuations and characters in the dataset which have less value in detecting the Tigrinya hate speech, therefore we have removed and cleaned those from our dataset using the following snapshot of code implementations.

Code 5. 2: Implementation for Cleaning the Dataset

```
def clean_text_round1(text):
#  text = re.sub(r'\s+', ' ', text)
 text=re.sub('\[.*?\]', '',text)
 text=re.sub('\n', '',text)
 text = re.sub(r'[^\w\s]','',text)
 text = re.sub(r'\s+', ' ', text).strip() # Cleaning the whitespaces
text=re.sub('\w*\d\w*', '',text)
 text = re.sub('([@A-Za-z0-9_]+)|[^\w\s]|#|http\S+', '', text)
 return text
round1= lambda x:clean_text_round1(x)
data_clean=pd.DataFrame(df.comments.apply(round1))
```

5.4.2. Text Normalization Implementation

Since in the Tigrinya language, there are some repeated letters that have the same sound but different in symbols, and therefore to make it easier for our model to train we have applied text normalization by using the Regular Expression provided by the python library and mapped those letters to only specific letter using the following snapshot of code implementations.

Code 5. 3: Implementation for Normalizing Tigrinya Characters

```
data_clean = data_clean.replace('ኀ','ሀ', regex=True)
data_clean = data_clean.replace('ኁ','ሁ', regex=True)
data_clean = data_clean.replace('ኂ','ሂ', regex=True)
data_clean = data_clean.replace('ኌ','ሄ', regex=True)
```

```
data_clean = data_clean.replace('ጎ','ሄ', regex=True)
data_clean = data_clean.replace('ጎ','ህ', regex=True)
data_clean = data_clean.replace('ጮ','ሆ', regex=True)

data_clean = data_clean.replace('ጸ','ፀ', regex=True)
data_clean = data_clean.replace('ጹ','ፁ', regex=True)
data_clean = data_clean.replace('ጺ','ፂ', regex=True)
data_clean = data_clean.replace('ጸ','ፃ', regex=True)
data_clean = data_clean.replace('ጼ','ፄ', regex=True)
data_clean = data_clean.replace('ጽ','ፅ', regex=True)
data_clean = data_clean.replace('ጾ','ፆ', regex=True)

data_clean = data_clean.replace('ሠ','ሰ', regex=True)
data_clean = data_clean.replace('ሡ','ሱ', regex=True)
data_clean = data_clean.replace('ሢ','ሲ', regex=True)
data_clean = data_clean.replace('ሣ','ሳ', regex=True)
data_clean = data_clean.replace('ሤ','ሴ', regex=True)
data_clean = data_clean.replace('ሥ','ስ', regex=True)
data_clean = data_clean.replace('ሦ','ሶ', regex=True)
```

5.4.3. Tokenization Implementation

We have tokenized our sentence in word-level tokenization which splits each sentence into its respective words, and we have achieved this tokenization process using the natural language toolkit (nltk) library, more specifically the RegexpTokenizer inside the nltk, and the following is a snapshot code for the implementation detail.

Code 5. 4: Implementation for Tokenization

```
# Tokenization
import nltk
from nltk.tokenize import RegexpTokenizer
regexp = RegexpTokenizer('\w+')
```

```
data_clean['text_token']=data_clean['comments'].apply(regexp.tokenize)
tokenized =data_clean['text_token']
tokenized
```

5.4.4. Stop Word Removal

Since stop words have no full meaning on their own, removing them is very essential, however, since there are no built-in stop word lists for the Tigrinya language, we have prepared our own stop word lists of corpora and stored them in the library of nltk folder. After preparing it we have directly applied the stop word removing processes using the following snapshot of implementation details.

Code 5. 5: Implementation for Stop Word removal

```
from nltk.corpus import stopwords
stop_words=set(stopwords.words('TigrinyaStopwords.txt'))
from nltk.tokenize import word_tokenize
def remove_stopwords(sentence):
    word_tokens = word_tokenize(sentence)
    clean_tokens = [w for w in word_tokens if not w in stop_words]
    return clean_tokens
data_clean['S'] = data_clean['comments'].apply(remove_stopwords)
data_clean
```

5.5. Feature Extraction Implementation

To feed our data to the machine first we need to convert it to numbers, and therefore we have used different feature extraction mechanisms such us Word2vec, Fast Text, and Kears Embedding layer. For the Word2vec and Fast Text, we have trained our own custom pre-trained Word2vec and Fast Text and save them to our computer, then finally we have used them accordingly in the neural network. But for the keras embedding layer there is no need to train the dataset separately, rather it only requires the unique integers assigned for each word using the Tokenizer class in the Keras library, then Keras Embedding layer adjusts everything by itself

using some parameters. Let us see the implementation detail for each of feature extraction techniques we have used for our study.

5.5.1. Training and Loading Word2vec Implementation

As we have explained, we have trained our custom Word2vec and saved it, and then finally loaded it to be used in the neural network. The following is the snapshot of the implementation detail.

Code 5. 6: Implementation for Training Custom Word2vec

```
#train word2vec model using gensim
from gensim.models import Word2Vec
from gensim.models import KeyedVectors


Min_count=1
Size=200
Workers=4
Window=7
model=gensim.models.Word2Vec(sentences=stopword_removed,vector_size=Size,sg=0,
workers=Workers,min_count=Min_count, window=Window)
model.build_vocab(stopword_removed, progress_per=1000)
words=list(model.wv.key_to_index)
# vocab_size=len(model.wv.index_to_key)
vocab_size=len(words)
vocab_size
```

Code 5. 7: Implementation for Saving Custom Word2vec

```
filename='w2v_200.txt'
model.wv.save_word2vec_format(filename,binary=False)
```

Code 5. 8: Implementation for Loading and Vectorizing Custom Word2vec

```
# opening the cutom trained word2vec and vectorizing it using the numpy array
```

```
embeddings_index = {}
f = open(os.path.join('w2v_200.txt'),encoding = 'utf-8')
print("file opened",f)
for line in f:
    values = line.split()
    word = values[0]
    coefs = np.asarray(values[1:], dtype='float32')
    embeddings_index[word] = coefs
f.close()

# creating the embedding matrix after opening it
print ('creating embedding matrix...')
embedding_matrix = np.zeros((len(word_index) + 1, 200))
for word, i in word_index.items():
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        # words not found in embedding index will be all-zeros.
        embedding_matrix[i] = embedding_vector
print('Found %s word vectors.' % len(embeddings_index))
```

5.5.2. Training and Loading Fast Text Implementation

We have also used the Fast Text for feature extraction mechanisms; therefore, we have prepared a custom pre-trained Fast Text. Since the saving and loading of the trained model is similar to Word2vec, we only put the snapshot code implementation detail of training with Fast Text as follows

Code 5. 9: Implementation for Training Custom Fast Text

```
from gensim.models.fasttext import FastText
model = FastText(vector_size=200, window=7, min_count=1, workers=7, sg=0, alpha=0.25, epochs=10)
```

```
# build the vocabulary
model.build_vocab(stopword_removed)
words=list(model.wv.key_to_index)
# vocab_size=len(model.wv.index_to_key)
vocab_size=len(words)
vocab_size
```

5.5.3. Training with Keras Embedding Layer Implementation

The Keras embedding layer can convert the text data without using the word2vec or Fast Text feature extraction techniques by using its own mechanisms, and as a parameter, it only requires the total number of unique words with their specific and unique integer value, the dimension of the word to be represented and the maximum length of a sentence in the whole dataset in which it helps to pad and create a matrix. The following is the snapshot code for the Keras Embedding layer.

Code 5. 10: Keras Embedding Layer to Assign Unique Integer for Each Word

```
maxlen = max([len(x.split()) for x in stopword_removed.values])
tokenizer = Tokenizer()
tokenizer.fit_on_texts(stopword_removed.values)
max_features = len(tokenizer.word_index) + 1
print(maxlen)
```

Code 5. 11: Implementation of Training with Keras Embedding layer

```
# Input / Embdedding
model.add(Embedding(max_features, 200, input_length=maxlen))
```

On the feature extraction techniques, we have implemented these three different feature extraction techniques, and we have selected the best feature extraction techniques which have resulted a promising result for our model. We have discussed which feature extraction techniques performed better than the others in the next chapters.

5.6. Implementation for the Dataset Split

As we have been discussing throughout the document, in order to train and test our model we have divided our dataset into train and test sets using the k-fold cross-validation. Hence the k-fold cross-validation technique splits the dataset into k-folded groups based on the ratio of k-1, and the one-leave-out principle where the k-1 folded group is used for training, the one-leave-out from the k-1 folded groups are used for testing, and iteratively all the k-folded group will be addressed. Therefore, we have divided our dataset into 10 number of k splits, and when 9 of the k-folded groups are used for training the rest 1 k-folded group is used for testing. The following is the implementation snapshot code for the train test split of our dataset.

Code 5. 12: Implementation for Train Test Split

```
from sklearn.model_selection import StratifiedKFold
kfold = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)
cvscores = []
for train, test in kfold.split(padded_file, y.argmax(1)):
   Bi_LSTM_Model = Sequential()
   Bi_LSTM_Model.add(emb)
```

5.7. Implementation of the Models

We have used three different deep learning algorithms specifically Bi-LSTM, CNN-LSTM combined, and CNN. In the previous chapter, it is briefly explained how the layers of the neural network work and also described in detail how the architectures of Bi-LSTM, CNN-LSTM, and CNN work with sequential data. Therefore, here we have discussed how these models have been implemented.

5.7.1. Bidirectional LSTM Implementation

We have used the framework of the Keras library to implement the Bidirectional Long-short memory to build and develop the Tigrinya hate speech detection, and the following snapshot code shows how we have built and implemented the Bi-LSTM model.

Code 5. 13: Implementation for Bidirectional LSTM

```
from tensorflow.keras.layers import Embedding, Dense,Flatten, LSTM, GRU, Dropout, Bidirectional
from tensorflow.keras import regularizers
EMBEDDING_DIM = 200
class_num = 3
num_words=len(word_index) + 1
emb=Embedding(input_dim = num_words,
        output_dim = EMBEDDING_DIM,
        weights = [embedding_matrix],
        input_length= max_length,
        trainable = False)
Bi_LSTM_Model = Sequential()
Bi_LSTM_Model.add(emb)
Bi_LSTM_Model.add(Bidirectional(LSTM(64,return_sequences=False)))
Bi_LSTM_Model.add(Dropout(0.5))
Bi_LSTM_Model.add(Dense(class_num, activation =
'softmax',kernel_regularizer=regularizers.L2(0.001)))
Bi_LSTM_Model.summary()
# Compiling the model
Bi_LSTM_Model.compile(loss="categorical_crossentropy",
optimizer=tf.keras.optimizers.Adam(learning_rate=0.01),
        metrics=["accuracy", tf.keras.metrics.Precision(), tf.keras.metrics.Recall()])
# Early stopping and saving the best perfoming model
from keras.callbacks import EarlyStopping, ModelCheckpoint
# es = EarlyStopping(monitor = 'val_loss', mode = 'min', verbose = 1, patience = 5)
mc = ModelCheckpoint('./model.h5', monitor = 'val_accuracy', mode = 'max', verbose = 1,
save_best_only = True)
# Fitting the model
history_embedding= Bi_LSTM_Model.fit(padded_file[train],y[train],
                epochs=10,
```

```
                validation_data=(padded_file[test],y[test]),
                verbose=1, batch_size=64, callbacks= [mc])
```

5.7.2. CNN-LSTM Combined Implementation

In our study we have also tried to combine CNN with LSTM, we have used the keras framework to do so, and therefore we have built a model of combined CNN-LSTM in which it detects the Tigrinya hate speech , the following snapshot of code implementation detail briefs how we have combined both of them

Code 5. 14: Implementation for CNN-LSTM Combined

```
from tensorflow.keras.layers import Embedding, Dense,Flatten, LSTM, GRU, Dropout,SpatialDropout1D, Conv1D,MaxPooling1D
from tensorflow.keras import regularizers
EMBEDDING_DIM = 200
class_num = 3
num_words=len(word_index) + 1
emb=Embedding(input_dim = num_words,
        output_dim = EMBEDDING_DIM,
        weights = [embedding_matrix],
        input_length= max_length,
        trainable = False)
lstm_Cnn_Model = Sequential()
lstm_Cnn_Model.add(emb)
lstm_Cnn_Model.add(Conv1D(filters=32, kernel_size=3, padding='same', activation='relu'))
lstm_Cnn_Model.add(MaxPooling1D(pool_size=2))
lstm_Cnn_Model.add(Dropout(0.7))
lstm_Cnn_Model.add(LSTM(10 , activation='relu'))
lstm_Cnn_Model.add(Dropout(0.5))
lstm_Cnn_Model.add(Dense(class_num, activation =
'softmax',kernel_regularizer=regularizers.L2(0.001)))
lstm_Cnn_Model.summary()
```

```
# Compiling the model
lstm_Cnn_Model.compile(loss="categorical_crossentropy",
optimizer=tf.keras.optimizers.Adam(learning_rate=0.01),
            metrics=["accuracy", tf.keras.metrics.Precision(), tf.keras.metrics.Recall()])

# Early stopping and saving the best perfoming model
from keras.callbacks import EarlyStopping, ModelCheckpoint

# es = EarlyStopping(monitor = 'val_loss', mode = 'min', verbose = 1, patience = 5)
mc = ModelCheckpoint('./model.h5', monitor = 'val_accuracy', mode = 'max', verbose = 1,
save_best_only = True)
# Fitting the model
history_embedding= lstm_Cnn_Model.fit(padded_file[train],y[train],
                        epochs=10,
                        validation_data=(padded_file[test],y[test]),
                        verbose=1, batch_size=64, callbacks= [mc])
```

5.7.3. CNN Implementation

Since Convolutional Neural Networks can be used for text classification, we have also built a model for the Tigrinya hate speech detection using the CNN model, and to build CNN based model we have implemented it using the following codes of snapshots.

Code 5. 15: Implementation for CNN

```
from tensorflow.keras.layers import Embedding, Dense,Flatten, LSTM, GRU,
Dropout,SpatialDropout1D, Conv1D,MaxPooling1D
from tensorflow.keras import regularizers
EMBEDDING_DIM = 200
class_num = 3
num_words=len(word_index) + 1
emb=Embedding(input_dim = hehe,
```

```
            output_dim = EMBEDDING_DIM,
            weights = [embedding_matrix],
            input_length= max_length,
            trainable = False)
Cnn_Model = Sequential()
Cnn_Model.add(emb)
Cnn_Model.add(Conv1D(filters=32, kernel_size=3, padding='same', activation='relu'))
Cnn_Model.add(MaxPooling1D(pool_size=2))
Cnn_Model.add(Flatten())
Cnn_Model.add(Dropout(0.5))
Cnn_Model.add(Dense(class_num, activation =
'softmax',kernel_regularizer=regularizers.L2(0.001)))
Cnn_Model.summary()

# Compiling the model
Cnn_Model.compile(loss="categorical_crossentropy",
optimizer=tf.keras.optimizers.Adam(learning_rate=0.01),
            metrics=["accuracy", tf.keras.metrics.Precision(), tf.keras.metrics.Recall()])

# Early stopping and saving the best perfoming model
from keras.callbacks import EarlyStopping, ModelCheckpoint

# es = EarlyStopping(monitor = 'val_loss', mode = 'min', verbose = 1, patience = 5)
mc = ModelCheckpoint('./model.h5', monitor = 'val_accuracy', mode = 'max', verbose = 1,
save_best_only = True)
# Fitting the model
history_embedding= Cnn_Model.fit(padded_file[train],y[train],
                        epochs=10,
                        validation_data=(padded_file[test],y[test]),
                        verbose=1, batch_size=64, callbacks= [mc])
```

5.7.4. Implementation of Model Evaluation Metrics

In order to evaluate the model, we have used the accuracy evaluation metrics, and different evaluation metrics such as precision, recall, and F1-Score. In addition to this confusion matrix has been applied to see the correctly and wrongly predicted classes.

With the following snapshots of codes we have compiled and evaluated the performance of Bi-LSTM, LSTM-CNN, and CNN models.

Code 5. 16: Implementation of Model Performance Evaluation Metrics

```
Bi-LSTM_Model.compile(loss="categorical_crossentropy",
optimizer=tf.keras.optimizers.Adam(learning_rate=0.01),
        metrics=["accuracy", tf.keras.metrics.Precision(), tf.keras.metrics.Recall()])
```

Hereafter compiling the model using the keyword of compile() which contains different parameters such as optimizers which are used to optimize our model by adjusting a learning rate using different techniques, the second parameter is the loss which tries to show us the loss value of the model and we have used categorical cross entropy for our multi-class classification, and binary cross entropy for our binary classification, then there is metrics which includes the accuracy, precision, and recall which tells how our model is performing in terms of performance. Then after the precision and recall of the model performance are known, the value of F1-Score can be calculated easily.

Generally, in this chapter, we have tried to show how we have implemented and designed the Tigrinya hate speech detection in a brief way.

CHAPTER SIX

6. Result and Discussion

6.1. Chapter Overview

In this chapter the results and discussion are discussed in detail, in the first section results of the evaluated models with their sub-topics are described, on the second section details about performance of different model in terms of accuracy are covered, then the comparison of our model with the existing works is the third section, then finally on the fourth section discussion about the whole research is described in detail..

6.2. Results of the Evaluated Models

As we have been describing in detail in the previous chapters, we have built and evaluated three different deep learning algorithms of Bi-LSTM, CNN-LSTM combined, and CNN using different feature extraction techniques. We have evaluated the performance of each model by splitting the dataset into train and test sets based on the stratified k-fold cross-validation techniques, in which the train set is used to train the model, and the test set is used for the purpose of evaluating the performance of the model.

6.2.1. CNN-LSTM Model Evaluation Result

As is described in detail in section 4.5.2, Most of the time LSTM is better for sequential data for it has its own mechanisms to handle long term dependences in a sentences of sequential data, and CNN was better in image classification, in which it classifies the image using the convolutional layer using different mechanisms. But this CNN can also be used in text classification, the only thing we need to do in order to use it for text classification is minimizing the convolutional layer from 3, or 2 dimensions to 1 dimension of convolution layer, the reason is sequential data are one dimensional. Therefore, as a researcher we have combined these two different deep learning algorithm techniques together to see if they can give us a good performance in the case of Tigrinya hate speech detection.

We have built this model using the word embedding layer as an input layer in which it provides vectorized words to the hidden layer, and at the hidden layer, we have used two hidden layers, in the first layer we have put the convolutional layer with its own filter size and max pooling. Then on the second layer, we used the LSTM layer with 10 units of neurons. And finally, since

our study is based on the binary and multi-class classification, for the binary classification of normal and hate classes we have used two neurons at the dense layer with sigmoid activation function, and for the multi-class classification which classifies Tigrinya written texts into one of the three categories called hate, normal, and offensive, we have used three neurons at the dense layer with softmax activation function. To reduce the model overfitting, we have also applied the L2 regularization techniques and we have evaluated our CNN-LSTM combined model for both binary and multi-class classification using different evaluation metrics, their result is presented as follows.

Table 11: CNN-LSTM Model Evaluation Result for Binary Classification

Metrics	Feature Extraction				
	Word2vec		Fast Text		Keras Embedding
	CBOW	Skip-gram	CBOW	Skip-gram	
Accuracy (In percent)	87.78	85.83	93.33	**95.28**	91.39

As a result, for the binary classification using the CNN-LSTM model the Fast Text with skip-gram model has outperformed the rest with an accuracy of 95.28 %.

Table 12: CNN-LSTM Model Evaluation Result for Multi-Class Classification

Metrics	Feature Extraction				
	Word2vec		Fast Text		Keras Embedding
	CBOW	Skip-gram	CBOW	Skip-gram	
Accuracy (In percent)	75	65.26	85.37	85	75
Precision (In percent)	78.10	72.03	86.23	86.34	75.09
Recall (In percent)	70	53.89	83.52	84.26	74.81
F1-Score (In percent)	73.82	61.65	84.85	85.28	74.94

Here for the multi-class classification, the information presented in the above table describes the detail of classifying the Tigrinya hate speech detection using the CNN-LSTM with different

feature extraction techniques. As a result, the feature extraction of Fast Text with the skip-gram model has outperformed the other feature extraction techniques with performances of Accuracy 85 %, Precision of 86.34 %, Recall of 84.26 %, and F1-score of 85.28 %.

6.2.1.1. HyperParameter Tuning of CNN-LSTM Model for Multi-Class Classification

For the CNN model, we have applied hyperparameter tuning to find an optimal value of parameters for the model. The major justification for tuning hyperparameters is the difficulty in determining a model's optimum performance when only one parameter combination is used. Therefore for our model, to find the optimal performance, we have manually tuned and applied different parameters including changing the number of epochs, changing the number of hidden layers, activation functions, optimizers and etc. hence the following table shows different combinations of parameters with their respective accuracy and F1-Score for CNN-LSTM.

Tested Parameters (TP)	***Hyperparameter values***							
	Epochs	***Batch -Size***	***Hidden layers***		***Optimizers***	***Activation Functions***	***Accuracy (In percent)***	***F1-Score (In percent)***
			CNN	***LSTM***				
TP1	10	64	1	1	Adam	Softmax	**85.37**	**84.85**
TP2	10	64	2	1	Adam	Softmax	83.12	82.36
TP3	10	64	1	2	Adam	Softmax	81.27	81
TP4	50	64	1	1	Adam	Softmax	84	83.69
TP5	50	64	2	1	Adam	Softmax	84.31	83.94
TP6	50	64	1	2	Adam	Softmax	83.44	83.23
TP7	10	128	1	1	Adam	Softmax	83.11	82.53
TP8	10	128	2	1	Adam	Softmax	84.52	84.24
TP9	10	128	1	2	Adam	Softmax	83.75	83.59
TP10	50	128	1	1	Adam	Softmax	82.86	82.36
TP11	50	128	2	1	Adam	Softmax	81.41	80.68
TP12	50	128	1	2	Adam	Softmax	81.12	80.87

Table 13: Hyperparameter Tuning of CNN-LSTM Model for Multi-Class Classification

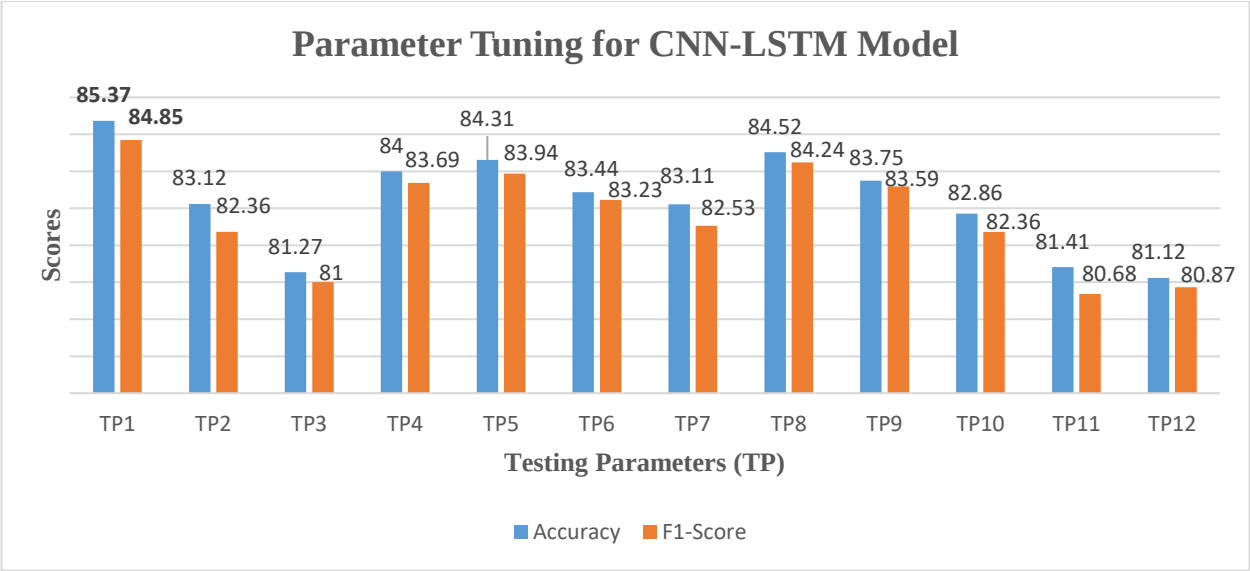


Figure 13: Parameter Tuning Results of CNN-LSTM Model for Multi-Class Classification

6.2.2. CNN Model Evaluation Result

In order to select the best performer amongst different models, we have also tried to implement our Tigrinya hate speech dataset using the convolutional neural network. As we have described in detail in the previous chapters in section 4.5.3 about how the convolutional neural network works for sequential data, we have applied it using different techniques, since there are three different layers in the concept of deep learning neural networks, on the input layer embedding layer is used for converting text to numbers and deliver it to the hidden layer. On the hidden layer, we have used one hidden layer of convolutional neural network with its different parameters such as filter size and max pooling. On the output layer, we have used three units of neurons for the multi-class classification and two units of neurons for the binary classification. Therefore we have evaluated our CNN-based model using different feature extraction mechanisms and evaluation metrics as follow.

Table 14: CNN Model Evaluation for Binary Classification

Metrics	Feature Extraction				
	Word2vec		Fast Text		Keras Embedding
	CBOW	Skip-gram	CBOW	Skip-gram	
Accuracy (In percent)	83.61	87.22	**93.61**	93.33	91.94

As a result, for the binary classification using the CNN model, the Fast Text with CBOW has outperformed the rest with an Accuracy of 93.61 %.

Table 15: CNN Model Evaluation for Multi-Class Classification

Metrics	Feature Extraction				
	Word2vec		Fast Text		Keras Embedding
	CBOW	Skip-gram	CBOW	Skip-gram	
Accuracy (In percent)	71.11	68.33	**83.33**	81.85	81.3
Precision (In percent)	75.66	72.27	**83.52**	81.78	82.18
Recall (In percent)	63.33	58.89	**82.59**	81.48	81.11
F1-Score (In percent)	68.94	64.89	**83.05**	81.62	81.64

Whenever the convolutional neural network model is applied to our dataset for the multi-class classification, it has provided promising results using different extraction techniques, as it is described in the above table the Fast Text with the CBOW has a better performance when it is evaluated using different evaluation metrics, and as a result it has provided an Accuracy of 83.33 %, Precision of 83.52 %, Recall of 82.59 %, and F1-score of 83.05 %.

6.2.2.1. HyperParameter Tuning of CNN Model for Multi-Class Classification

For the CNN model, we have applied hyperparameter tuning to find an optimal value of parameters for the model. Hence the following table shows different combinations of parameters with their respective accuracy and F1-Score for CNN model.

Table 16: Hyperparameter Tuning of CNN Model for Multi-Class Classification

Tested Parameters (TP)	***Hyperparameter values***						
	Epochs	***Batch -Size***	***Hidden layers***	***Optimizers***	***Activation Functions***	***Accuracy (In percent)***	***F1-Score (In percent)***
TP1	10	64	1	Adam	Softmax	**83.33**	**83.05**
TP2	10	64	2	Adam	Softmax	81.65	80.36
TP3	50	64	1	Adam	Softmax	81.13	81.27

TP4	50	64	2	Adam	Softmax	82.86	82.43
TP5	100	64	1	Adam	Softmax	80.34	80.18
TP6	100	64	2	Adam	Softmax	81.5	81.65
TP7	10	128	1	Adam	Softmax	82.55	82.05
TP8	10	128	2	Adam	Softmax	82.31	82.45
TP9	50	128	1	Adam	Softmax	80.30	80.15
TP10	50	128	2	Adam	Softmax	81.81	80.96
TP11	100	128	1	Adam	Softmax	81.39	80.54
TP12	100	128	2	Adam	Softmax	82.28	81.56

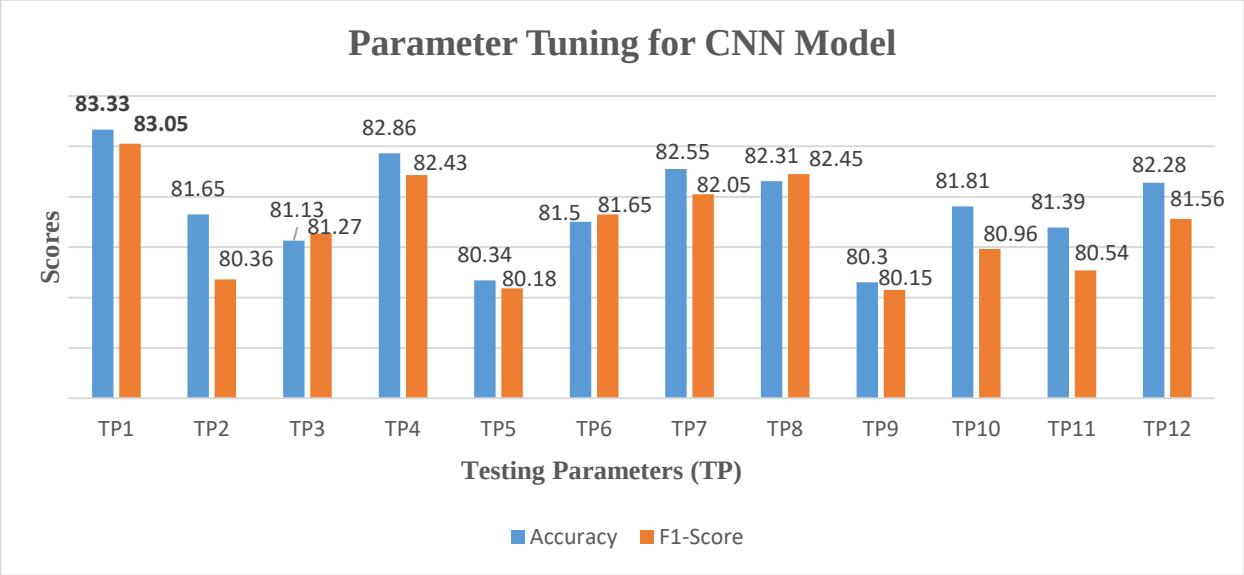


Figure 14: Parameter Tuning Results of CNN Model for Multi-Class Classification

6.2.3. Bi-LSTM Model Evaluation Result

Inside TensorFlow, there is a library called Keras API which provides us the Bi-LSTM model, and different evaluation metrics which helps us to evaluate the model of the Bi-LSTM. As we have been discussing earlier in the concept of deep learning there are three layers, therefore for the Bi-LSTM model we have used the embedding layer as an input layer, which provides the vectorized words to the hidden layers, on the hidden layer we have used one layer of Bi-LSTM in which the hidden layer contains 64 units of neurons, and finally, on the output of dense layer, we have used three units of neurons with softmax activation functions for the multi-class

classification. The reason we put only three units of neurons on the output layer is that our number of classes to be classified has only three categories called hate, offensive and normal, and the softmax activation function is the recommended type of activation function for the multi-class classification. On the other hand, we have used two units of neurons at dense layer for the binary classification with activation function of sigmoid. The following table describes the detailed results achieved using the Bi-LSTM model

Table 17: Bi-LSTM Model Evaluation for Binary Classification

Metrics	Feature Extraction				
	Word2vec		Fast Text		Keras Embedding
	CBOW	Skip-gram	CBOW	Skip-gram	
Accuracy (In percent)	84.72	85.28	95.83	**96.11**	95.83

As a result, for the binary classification using the Bi-LSTM model, the Fast Text with Skip-gram has outperformed the rest with an accuracy of 96.11 %. Therefore, for the binary classification the Bi-LSTM model with Fast Text of skip-gram has outperformed the CNN-LSTM and CNN with an accuracy of 96.11 %

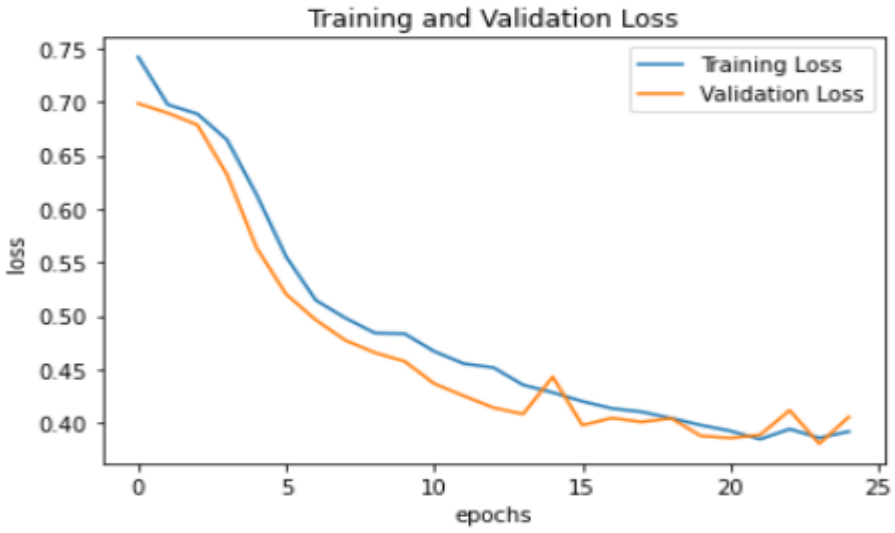


Figure 15 : Model Loss of Bi-LSTM for Binary Classification

Table 18: Bi-LSTM Model Evaluation for Multi-Class Classification

Metrics	Feature Extraction				
	Word2vec		Fast Text		Keras Embedding
	CBOW	Skip-gram	CBOW	Skip-gram	
Accuracy (In percent)	71.48	66.3	**87.41**	85	85.74
Precision (In percent)	76.83	69.66	**88.02**	85.26	86.19
Recall (In percent)	62.04	57.41	**85.74**	84.63	85.56
F1-Score (In percent)	68.64	62.94	**86.86**	84.94	85.87

Here, based on the result in the above described table, from the feature extraction techniques of word2vec, Fast Text and Keras embedding layer for the multi-class classification, the Fast Text with the Continuous Bag of Words has outperformed the word2vec and Keras embedding layer of feature extraction with a performance value of Accuracy 87.41 %, Precision 88.02 %, Recall 85.74 %, and F1-Score of 86.86 %. Generally, the Bi-LSTM model has outperformed the CNN-LSTM and CNN models using Continuous Bag of Words of Fast Text.

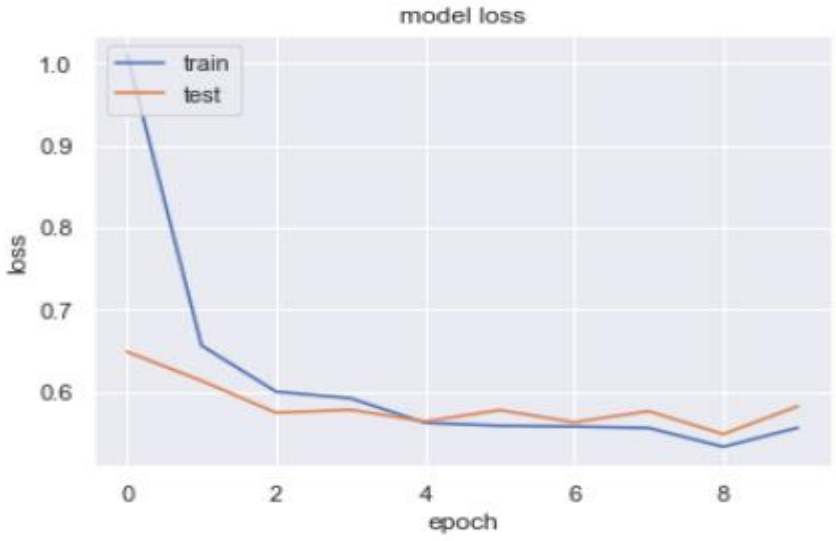


Figure 16: Model Loss of Bi-LSTM for Multi-Class Classification

6.2.3.1. Confusion Matrix of the Bi-LSTM Model for Binary Classification

The overall size of our dataset for binary classification is 3608, with 1793 and 1815 labeled as normal and hate respectively, and we have trained our model based on stratified K-fold cross-validation by splitting the dataset into 10 different k-folded groups. As a result using confusion matrix, the Bi-LSTM model has correctly classified 1779 as normal from the 1793 of the normal class, and incorrectly classified 14 of it as hate. From the 1815 hate classes, our model has classified 1804 of it correctly as hate, and wrongly classified 11 of it as normal. The following confusion matrix shows the prediction of binary classification using Bi-LSTM model

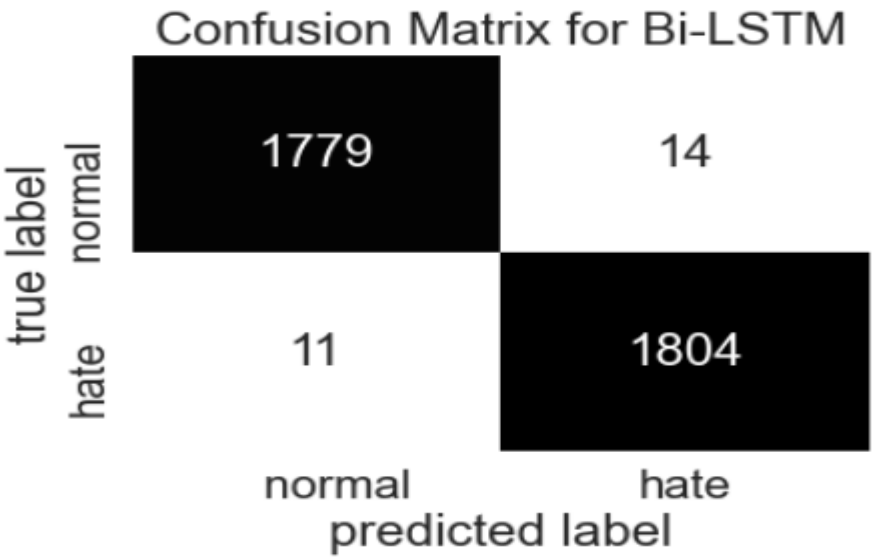


Figure 17: Confusion Matrix of the Bi-LSTM models for Multi-Class Classification

6.2.3.2. Confusion Matrix of the Bi-LSTM Model for Multi-Class Classification

A confusion matrix is a way that is used to show how our trained model got confused during predictions. Therefore, since our Bi-LSTM model has performed better than the CNN-LSTM and the CNN, we have seen a detailed classification of it using the confusion matrix. The confusion matrix has been applied to the Bi-LSTM model after the dataset is trained based on the K-fold cross-validation techniques. As described in earlier chapters, the overall size of our dataset is 5400, with 1793, 1815, and 1792 labeled as normal, hate, and offensive classes, respectively, and we have trained our model based on stratified cross-validation by splitting the dataset into 10 different k-folded groups. We have used the confusion matrices to determine

how our Bi-LSTM model classifies each class appropriately. The confusion matrix shows how a total of 5400 datasets are categorized correctly and incorrectly as hate, offensive, and normal. As a result, from the 1793 of the normal classes, our model has classified 1650 of them correctly as normal classes and incorrectly classified 27 of them as hate, and 116 of them as offensive. From the 1815 hate classes, our model has classified 1648 of them correctly as hate classes and incorrectly classified 59 of them as normal, and 108 of them as offensive. At the end, from the 1792 offensive classes, our model has classified 1626 of them correctly as offensive classes and incorrectly classified 81 of them as hate, and 85 as normal.

The following figure is the detailed result of the confusion matrix during predicting the Tigrinya written texts and classifying them as hate, offensive and normal accordingly.

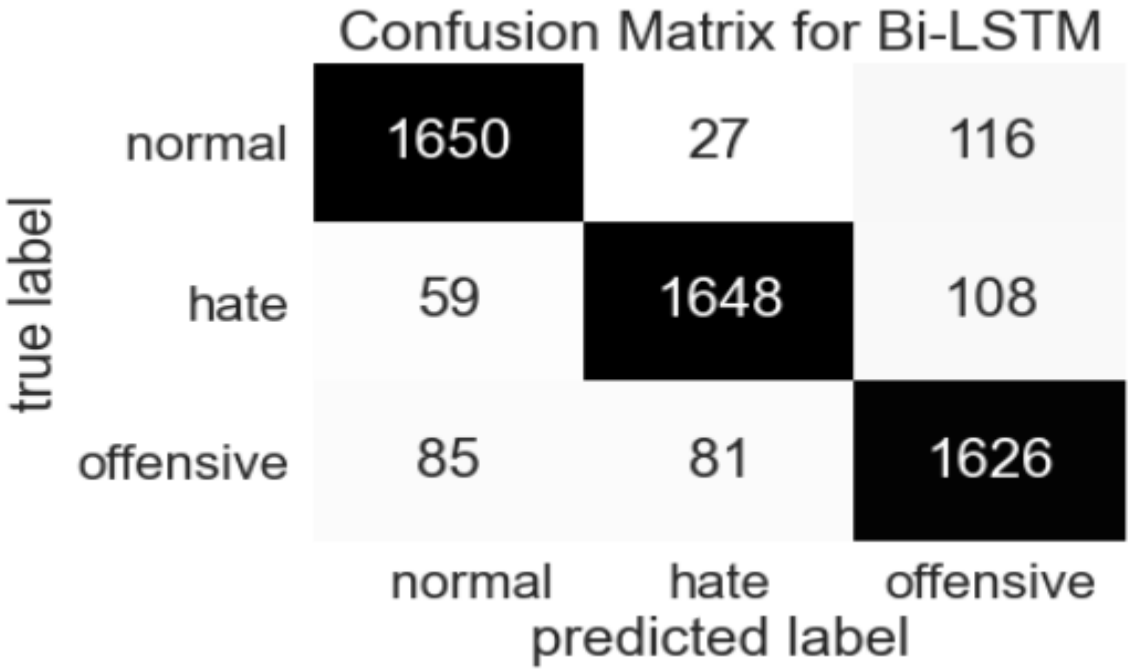


Figure 18: Confusion Matrix of the Bi-LSTM models for Multi-Class Classification

Generally, according to the report of the above confusion matrix, there has been a little confusion between identifying the hate from the offensive, and identifying the offensive speech from the normal speech, and hate speech was also a bit confusing for our model.

6.2.3.3. HyperParameter Tuning of Bi-LSTM Model for Multi-Class Classification

For our best-performing model of Bi-LSTM, we have applied hyperparameter tuning to find an optimal value of parameters for the model. Hence the following table shows different combinations of parameters with their respective accuracy and F1-Score of the Bi-LSTM model.

Table 19: Hyperparameter Tuning of Bi-LSTM Model for Multi-Class Classification

Tested Parameters (TP)	***Hyperparameter values***						
	Epochs	***Batch -Size***	***Hidden layers***	***Optimizers***	***Activation Functions***	***Accuracy (In percent)***	***F1-Score (In percent)***
TP1	10	64	1	Adam	Softmax	**87.41**	**86.86**
TP2	10	64	2	Adam	Softmax	86.23	85.59
TP3	50	64	1	Adam	Softmax	85.78	84.97
TP4	50	64	2	Adam	Softmax	86.11	85.33
TP5	100	64	1	Adam	Softmax	85.93	86
TP6	100	64	2	Adam	Softmax	86.11	86
TP7	10	128	1	Adam	Softmax	85.93	85.57
TP8	10	128	2	Adam	Softmax	84.44	84.24
TP9	50	128	1	Adam	Softmax	86.48	86.11
TP10	50	128	2	Adam	Softmax	86.33	86.13
TP11	100	128	1	Adam	Softmax	86.11	85.2
TP12	100	128	2	Adam	Softmax	86.43	86.45

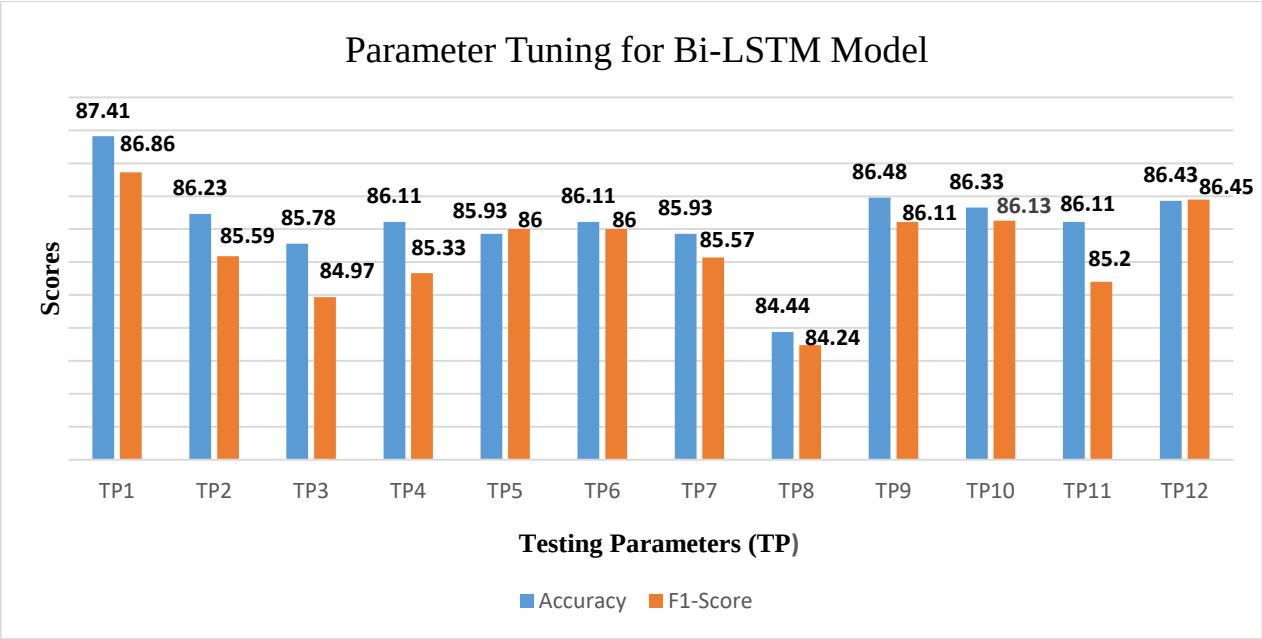


Figure 19: Parameter Tuning Results of Bi-LSTM Model for Multi-Class Classification

6.2.3.4. Bi-LSTM Model Evaluation Demo

We have demonstrated the ability of our model by giving new data to the model, and detect the text, and classify it as hate, offensive, and normal. The following code snapshot shows the demonstration for evaluation of our Bi-LSTM model.

Code 6. 1: Bi-LSTM Model Evaluation Demo

```
txt = ["መንነት ኣለኒ ኢልካ ዲኻ ትሓስብ ዘለኻ ጸ**ላሕ ባ*ያ"]

seq = tokenizer.texts_to_sequences(txt)
padded = pad_sequences(seq, maxlen=max_length)
pred = Bi_LSTM_Model.predict(padded)
labels = ['hate', 'offensive', 'normal']

print(pred)
print(np.argmax(pred))
print(labels[np.argmax(pred)-1])

[[3.7251116e-04 9.9923837e-01 3.8912270e-04]]
1
hate

txt = ["እግዚኣብሄር ጽንዓቱን ምሕረቱን ይሃብኩም ዘም ኣሕዋተይ"]

seq = tokenizer.texts_to_sequences(txt)
```

```
padded = pad_sequences(seq, maxlen=max_length)
pred = model.predict(padded)
labels = ['hate', 'offensive', 'normal']

print(pred)
print(np.argmax(pred))
print(labels[np.argmax(pred)-1])

[[0.50873274 0.14672324 0.344544  ]]
0
normal
txt = ["አንታ ለይባ ትም ኢልኻ ዘይትሰርቕ ላህማም"]

seq = tokenizer.texts_to_sequences(txt)
padded = pad_sequences(seq, maxlen=max_length)
pred = model.predict(padded)
labels = ['hate', 'offensive', 'normal']

print(pred)
print(np.argmax(pred))
print(labels[np.argmax(pred)-1])

[[0.36185002 0.05536045 0.5827896 ]]
2
offensive
```

6.3. Performance of our Models in terms of Accuracy

Three different deep learning approaches have been used throughout our work, and they have all produced surprisingly positive results for the Tigrinya hate speech detection for the binary and multi-class classification, especially when combined with Fast Text feature extraction techniques. As a result, for the multi-class classification the combination of LSTM and CNN models has produced an accuracy of 85.37 %; the CNN model likewise produced an accuracy of 83.33 %; and the Bi-LSTM model produced an accuracy of 87.41 %, which is a better result. Even though the Bi-LSTM model outperformed the others, the rest of the model still produced impressive results. The following graph shows the models with their respective results.

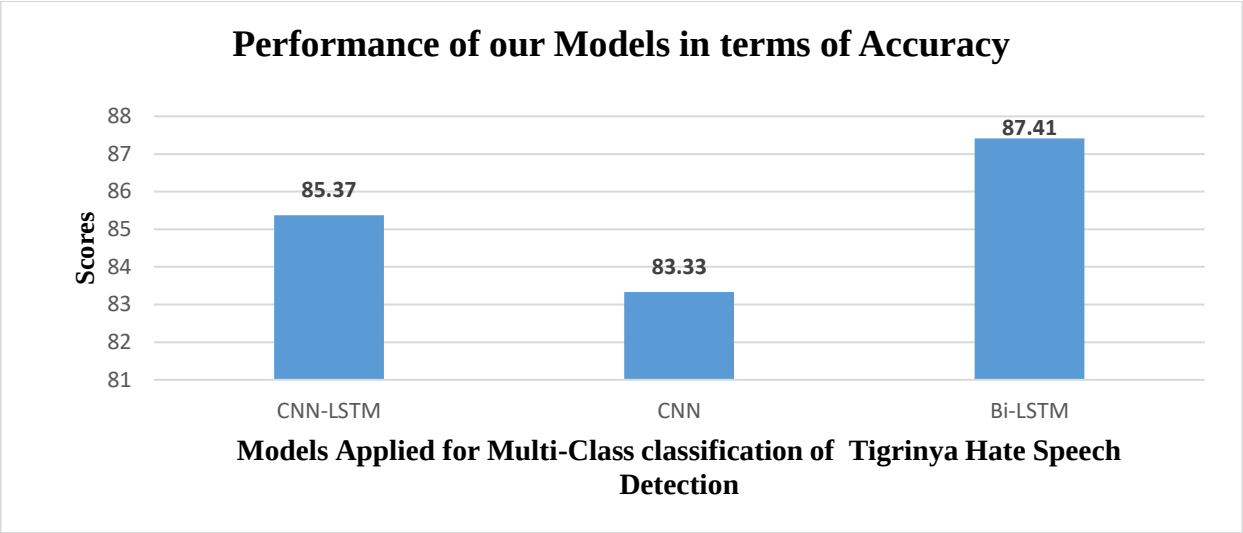


Figure 20: Performance of our Models in terms of Accuracy for Multi-Class Classification

Additionally, for the binary classification of the Tigrinya hate speech detection the Bi-LSTM model has outperformed CNN-LSTM and CNN with an accuracy of 96.11%.

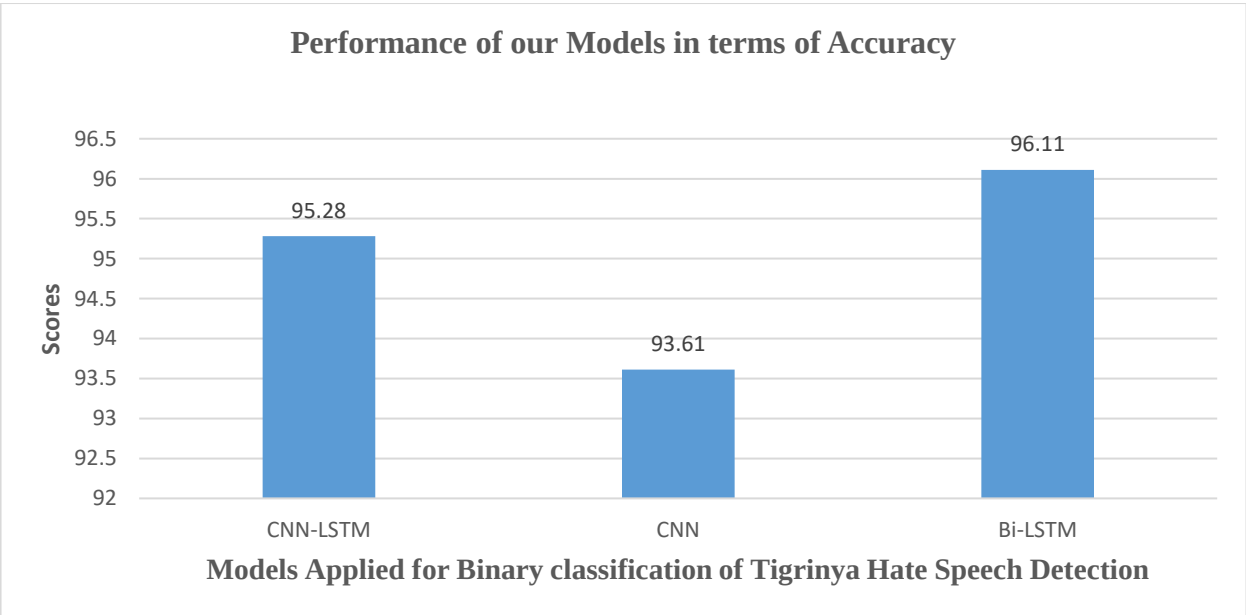


Figure 21: Performance of our Models in terms of Accuracy for Binary Classification

6.4. Comparison of our Models with the Existing Models

Even though there is no research studied regarding the Tigrinya hate speech detection, there are some related studies done in our country. These researches are hate speech detection for Amharic language and Afan-Oromo languages. As described in section 2.8 the first research

done for the Amharic language is done by (Zewdie Mossie and Jenq-Haur Wang, 2018), they have used machine learning techniques of Naïve Bayes to detect Amharic hate speeches from the non-hate using Word2vec and TF-IDF feature extraction techniques and they have achieved 79% of accuracy. When we compare it with our study, we have used different deep learning approaches than the traditional machine learning approaches using three different feature extraction techniques, namely Word2vec, Fast Text, and Keras Embedding layer. By using this mechanism, we have achieved a better performance of accuracy for both binary and multi-class classification than this Amharic hate speech detection. There is also another research done for the Amharic language by (Defar, 2019), the good part of this research is that, they have prepared two separate datasets one for the binary classification and the other for ternary classification, and also they have compared more than six feature extraction techniques namely, TF-IDF, Word2vec and by combining n-grams with TF-IDF using three different traditional machine learning algorithms. Finally, they ended up with 73% of F1-Score using SVM with Word2vec for the binary classification and 53% for the ternary classification. In our study we have also prepared two separated datasets for the binary and multi-class classification, and we have achieved a better model performance using various deep learning algorithms with three different feature extraction techniques

The other research studied for Amharic hate speech detection is done by (Surafel Getachew Tesfaye and Kula Kekeba Tune, 2020), this research has achieved better performance for Amharic hate speech detection with an accuracy of 97.9%. They have used two variants of RNN called LSTM, and GRU using Word2vec of feature extraction. This research was a benchmark for our study, which has helped us to think variants of RNN could provide a better performance in the area of hate speech detection. This research has used around 30,000 datasets and has a better performance for binary classification, and when we compare it with our study, for the binary classification we have provided comparable results using deep learning techniques with much lesser datasets. In addition to this we have applied multi-class classification for the Tigrinya hate speech detection by adding one more classes called offensive speech while the research done by (Surafel Getachew Tesfaye and Kula Kekeba Tune, 2020) has performed only for the binary classification. Additionally we have tried to combine LSTM with the CNN, even though it has achieved a lesser result than the Bi-LSTM model, it is still better, and we have

showed that it is possible to use different deep learning approaches than just Recurrent Neural Network for the Tigrinya hate speech detection.

For the Afan-Ororomo hate speech detection different studies have been done, the first one is (Naol Bakala Defersha and Kula Kekeba Tune, 2021), they have used six different traditional machine learning techniques to compare different results using TF-IDF combined with bigram for the feature extraction purposes, and they have got precisions of 66%, recall of 66% F-Score of 66%. But for our study we have built a model using different deep learning approaches with a better performance than this study.

Another researches conducted for Afan-Oromo hate speech detection is by (kanessaa, 2021), they have done a binary hate speech detection by comparing four different traditional machine learning algorithms using Word2vec, TF-IDF, and TF-IDF combined with Word2vec as feature extraction techniques and they have achieved an accuracy, F1-Score, precision and recall of 0.96%. Here when we compare it to our Tigrinya hate speech detection, for the binary classification we have achieved a better performance with accuracy of 96.11%. In addition to this we have applied multi-class classification for the Tigrinya hate speech detection by adding one more classes called offensive speech while the research done by (kanessaa, 2021) have only classified for the binary classification. Furthermore, we have explored different deep learning algorithms than using the traditional machine learning algorithm. Additionally (Kemal, 2021) has studied bilingual hate speech detection for Afaan Oromo and Amahric language, they have used six different deep learning approaches and the Bi-LSTM has achieved a better performance than the rest five deep learning approaches. As a gap they only have studied for binary classification, except for the number of classes they already did a great work.

6.5. Discussion

In this study, our main objective is to build the Tigrinya hate speech detection model from Facebook posts and comments, and therefore to fulfill our objective some research questions have been raised in section 1.4. Throughout this whole document, we have been trying to answer the research questions that have been raised at the beginning of the study, using different techniques. We have discussed in detail the scientific methods and techniques we have used to answer the research question, therefore for clarification, we have first put the research question, and then we have discussed in what way we have responded to each question.

The first research question was "How to identify the data sources and prepare the dataset for the study??" here to answer this research question we have applied different techniques. In order to prepare and build the datasets for the Tigrinya hate speech detection, primarily before we started to collect the dataset, we identified our data sources. As a result, the main sources for our datasets were popular activist pages, government official pages, and political parties' Facebook pages, and therefore based on these data sources we have collected the dataset from Tigray Media House (TMH), Dimtsi Woyane International (DW), center for research and documentation ማእከል ምርምርን ስነዳን ግዱሳት ሃገራውያን, Tigray Television official Facebook pages (TTV), Tigray Prosperity Party Official Pages, ኤርትራ ወይ ሞት Eritrea wey Mot and Tigray Communication Affairs Official pages. As a sampling technique, each data source of the Facebook page should have at least 15 thousand followers, and each post and comment need to be written in Tigrinya (Fidel) script. Hence, we have used a tool called Facepager which is an API of Facebook, therefore by using this tool we have collected different Tigrinya written posts and comments. After we have collected the data, we gave 1000 datasets to three annotators to label them as hate, offensive, and normal based on the Ethiopian hate speech law and guidelines of the Facebook community standard, then to check whether the labeling is correct and unbiased we applied the Fleiss Kappa inter-annotator agreement, and as a result with the value of 0.65 of Fleiss Kappa the level of agreement between annotators is a substantial level of agreement. Finally, we saved it in CSV file format, therefore these whole steps have been applied to prepare it, and this is how we identified the data source and prepared our dataset.

The second research question is "Which feature extraction techniques give good performance for building the model? To answer this question, we have applied three different feature

extraction techniques namely, Fast Text, Word2vec, and Keras Embedding layer. The main purpose of feature extraction is to vectorize each word into real-valued vectors or numeric forms of matrices to enable it to be understood by the machines. We have compared the results achieved using these three different techniques, and as a result, the Fast Text and Keras Embedding layer have resulted with better performance than the Word2vec. More specifically to answer the research questions, for the multi-class classification the Fast Text using the continuous bag of words has outperformed the rest of the feature extraction techniques applied throughout our study. On the other hand, for the binary classification the Fast Text using the skip-gram model has outperformed the rest of the feature extraction techniques

The third research question is “Which deep learning approaches can be used to build a best performing model that can accurately predict and classify the Tigrinya texts as hate, offensive and normal speech?” to answer this question we have applied three different deep learning techniques and these are Bi-LSTM which is variants of Recurrent Neural Networks, CNN-LSTM which is a combination of LSTM and CNN, and finally we have also applied the CNN to our dataset. As a result, the Bi-LSTM model has outperformed the CNN-LSTM and CNN for both binary and multi-class classification. But the CNN and CNN-LSTM has also achieved very promising result in detecting Tigrinya hate speeches, and therefore they could be applied in the Tigrinya text classification for the future.

Conclusions and Future Works

Conclusion

The main objective of our study is to develop and build a model that could detect Tigrinya written texts and classify it as hate, offensive or normal, and therefore we have built it using different techniques. To build the Tigrinya hate speech detection model, collecting the data from different Facebook pages was our first task, and we have collected 5400 posts and comments with unique words of 19,507 for the multi-class classification. For the binary classification we have used 3608 posts and comments, and it has a unique word of 15017, and saved it in a CSV file format. After collecting the dataset, we have applied some preprocessing techniques which includes data cleaning, normalization, tokenization, and removing stop words, and this preprocessing technique has helped the model to only consider on the main keywords from the dataset. Then different feature extraction techniques have been applied, these are the Word2vec, Fast Text, and the Keras Embedding layer. The reason we need to use the feature extraction technique is, that each word in the dataset has to be understood by the machine, and in order to do so they need to be converted into numbers, and we have applied comparison of feature extraction and selected the best performing feature extractor.

We have used three different deep learning algorithms of Bi-LSTM, CNN-LSTM combined, and CNN for both binary and multi-class classification to implement the Tigrinya hate speech detection. In our study, we have tried each of these deep learning algorithms using the three feature extraction techniques. As a result, for the multi-class classification the Bi-LSTM model using the Fast Text of CBOW has outperformed the CNN-LSTM combined, and the CNN with an Accuracy of 87.41 %, the precision of 88.02 %, recall of 85.74 %, and F1-Score of 86.86 %. On the other hand, for the binary classification which classifies Tigrinya texts as hate and normal, the Bi-LSTM model using the Fast Text of skip-gram has outperformed the CNN-LSTM combined, and the CNN with an Accuracy of 96.11 %. As a researcher even though the CNN-LSTM, and CNN has performed lesser than the Bi-LSTM model, we have noticed that the CNN-LSTM and CNN are also good for both binary and multi-class classification of Tigrinya hate speech detection, additionally from the extraction techniques the Fast Text and Keras Embedding layer has performed better than the Word2vec for the Tigrinya hate speech detection.

Future Works

The main goal of this study was to build a model that detects the Tigrinya written texts and classify it as hate, offensive and normal for multi-class classification and classifying as hate and normal for binary classification, and we have achieved this by using different techniques. Even though our model could detect the Tigrinya hate texts from the non-hate, there are a lot of missing techniques to be included in our research, in which these missing techniques can have an effect on improving the performance of the model. The following are future works to be included in the Tigrinya hate speech detection

- Building a dataset that includes all of the Tigrinya dialects, here in our study the main challenge was all the Tigrinya speaker writes using based on their dialect which might vary from the well-known Tigrinya texts, and therefore trying to correct each text is not the right thing, preparing datasets which include the whole Tigrinya dialect would be much better, and therefore this could be our future work.
- Building multilingual hate speech detection model is also our future plan, because people might diffuse hate speech by mixing Tigrinya, Afan-Oromo and Amharic languages. Additionally, instead of building different models for each language, building one model which is trained on different languages would be better.
- Including emojis and graphics reaction in a sentence, using emoji or graphics people might express their feeling negatively or positively towards something, therefore for the future we will include emojis and graphics reaction in the Tigrinya hates speech detection.
- When we build these Tigrinya hate speech detection models, we didn't have applied stemming or lemmatization techniques, which might provide a change in the performance of our model, therefore including stemming and lemmatization is our future plan in the Tigrinya hate speech detection.

. . .

Special Acknowledgment: This research was funded by Adama Science and Technology University under the grant number **ASTU/SM-R/495/22**

Adama, Ethiopia

References

Abrha, M. (2019). Mapping Online Hate Speech among Ethiopians: The Case of Facebook, Twitter and YouTube . *unpublished*.

Adhanom, I. B. (Dec 2019). A First Look into Neural Machine Translation for Tigrinya. *research gate*.

al., P. S. (2020). Reasons for Facebook Usage: Data From 46 Countries.

Al-Khalifa, R. A. (2020). A Deep Learning Approach for Automatic Hate Speech Detection in the Saudi Twittersphere.

Ashwin Geet D'Sa, Irina Illina, et.al. (n.d.). Classification of Hate Speech Using Deep Neural Networks.

Auliya Rahman Isnain, Agus Sihabuddin, et.al. (2020). Bidirectional Long Short Term Memory Method and Word2vec Extraction Approach for Hate Speech Detection. *Indonesian Journal of Computing and Cybernetics Systems*, 169-178.

Awet Fesseha, Eshete Derb Emiru et al. (2021). Text Classification Based on Convolutional Neural Networks and Word Embedding for Low-Resource Languages: Tigrinya.

Ayalew, Y. E. (2020). Defining 'Hate Speech' under the Hate Speech Suppression Proclamation in Ethiopia.

BERHE, G. A. (2011). A TWO STEP APPROACH FOR TIGRIGNA TEXT CATEGORIZATION . *unpublished*.

Birhanu, G. A. (2017). Automatic Text Summarizer for Tigrinya.

Birhanu, G. A. (2017). Automatic Text Summarizer for Tigrinya Language. *unpublished*.

Defar, Y. K. (2019). Hate Speech Detection For Amharic Language on Socila Media Using Machine Learning Techniques. *unpublished*.

Eddy Muntana Dharma, Ford Lumban Gaol. et.al. (2022). THE ACCURACY COMPARISON AMONG WORD2VEC, GLOVE, AND FASTTEXT TOWARDS CONVOLUTION

NEURAL NETWORK (CNN) TEXT CLASSIFICATION. *Journal of Theoretical and Applied Information Technology*.

G/Medhin, M. T. (2018). Trilingual Sentiment Analysis on Social Media . *unpublished*.

Hunde, F. E. (2021). Designing and Developing Bilingual Chatbot for Assisting Ethio-Telecom Customers on Customer Services. *unpublished* .

Jitendra Singh Malik, Guansong Pang, et.al. (Feb 2022). DEEP LEARNING FOR HATE SPEECH DETECTION: A COMPARATIVE STUDY.

kanessaa, L. G. (2021). Hate Speech Detection Framework from Social Media Content: The Case of Afaan . *unpublished*.

Kemal, B. S. (2021). Bilingual Social Media Text Hate Speech Detection For Afaan Oromo and Amharic Languages Using Deep Learning. *unpublished*.

Mendel, T. (2006). *Study on International Standards Relating to Incitement to Genocide or Racial Hatred.*

MercyCorpus. (2019). *How social media can spark violence and what.*

Museum, U. S. (n.d.). *HATE SPEECH AND GROUP-TARGETED VIOLENCE: The Role of Speech in Violent Conflicts.*

Naol Bakala Defersha and Kula Kekeba Tune. (2021). Detection of Hate Speech Text in Afan Oromo Social Media using Machine Learning Approach. *INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY*, 2567–2578.

Peace, E. I. (2021). *Fake news Misinformation and Hate Speech in Ethiopia: avulnerabilty assesment.*

Piotr Bojanowski. et.al. (2016). *fasttext.cc*. Retrieved from https://fasttext.cc/docs/en/pretrained-vectors.html

Pradeep Kumar Roy,Asis Kumar Tripathy et.al. (November 20, 2020.). A Framework for Hate Speech Detection Using Deep Convolutional Neural Network.

Putra, Dewa Ayu Nadia Taradhita and I Ketut Gede Darma. (2021). Hate Speech Classification in Indonesian Language Tweets by Using Convolutional Neural Network. *J. ICT Res. Appl*, 225-239.

Raghad Alshalan and Hend Al-Khalifa. (2020). A Deep Learning Approach for Automatic Hate Speech Detection in the Saudi Twittersphere. *applied sciences*.

Regassa, M. G. (2017). Topic-based Tigrigna Text Summarization Using WordNet. *unpublished*.

Scheffler, A. (2015). *The Inherent Danger of Hate Speech Legislation: A Case Study from Rwanda and Kenya on the Failure of a Preventative Measure.*

Sindhu Abro, Sarang Shaikh, et.al. (2020). Automatic Hate Speech Detection using Machine Learning: A Comparative Study. *International Journal of Advanced Computer Science and Applications*, 484-491.

Sindhu Abro, Sarang Shaikh. et.al. (2020). Automatic Hate Speech Detection using Machine Learning: A Comparative Study. *International Journal of Advanced Computer Science and Applications*.

Snaddon, B. (2017). *Hate speech and inter-ethnic violence in Nigeria.*

Son T. Luu, Hung P. Nguyen, et.al. (Jan, 2020). Comparison Between Traditional Machine Learning Models And Neural Network Models For Vietnamese Hate Speech Detection. ResearchGate.

Stevens, J. P. (June 2012). Should Hate Speech Be Outlawed?

Surafel Getachew Tesfaye and Kula Kekeba Tune. (2020). Automated Amharic Hate speech Posts and Detection Model using Recurrent Neural Network.

Tela, A. F. (May 2020). Sentiment Analysis for Low-Resource Language: The Case of Tigrinya. *unpublished*.

Waldron, J. (2012). *The Harm in Hate Speech.*

Yemane Keleta Tedla , Kazuhide Yamamoto, et.al. (2016). Tigrinya Part-of-Speech Tagging with Morphological Patterns and the New Nagaoka Tigrinya Corpus. *International Journal of Computer Applications* , 33-41.

Yohannes, A. (n.d.). *chilot.* Retrieved from https://chilot.me/2020/09/15/hate-speech-on-facebook-is-pushing-ethiopia-dangerously-close-to-a-genocide/

Zemicheal Berihu, Gebremariam Mesfin, et.al. (2020). Enhancing Bi-directional English-Tigrigna Machine Translation Using Hybrid Approach. *NIK 2020 conference.* Oslo, Norway .

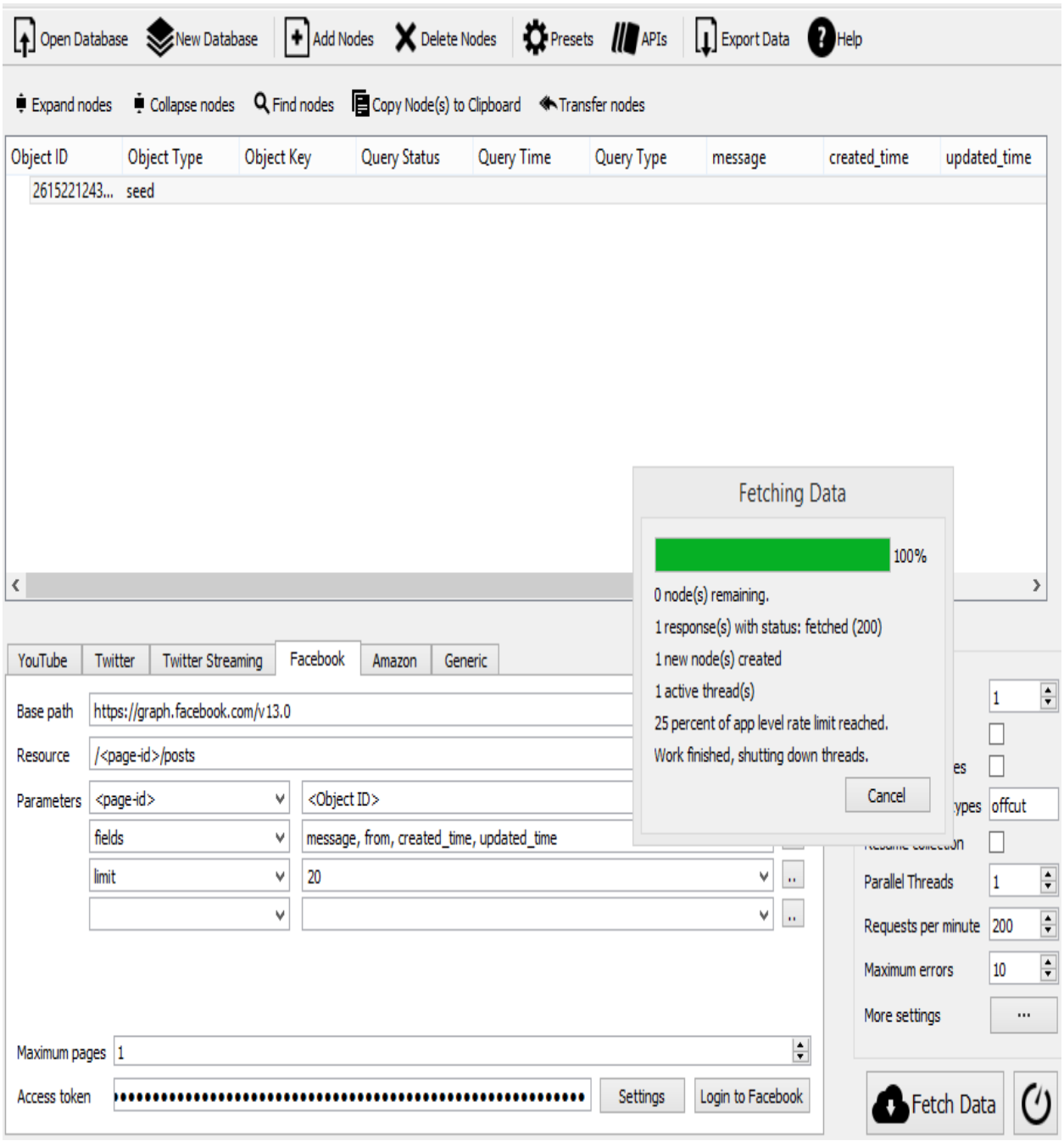
Zemicheal Berihu, Gebremariam Mesfin. et. al. (2020). Enhancing Bi-directional English-Tigrigna Machine Translation Using Hybrid Approach. *NIK 2020 conference.* Oslo, Norway .

Zewdie Mossie and Jenq-Haur Wang. (2018). SOCIAL NETWORK HATE SPEECH DETECTION FOR AMHARIC LANGUAGE. 41-55.

APPENDIX

Appendix A: Data Collection Sample

Appendix A.1: Data Collection Sample using Facepager Tool



level	id	parent_id	object_id	object_type	object_key	query_status	query_tim	query_type	message
0	1	None	3757132896403	seed			None	None	
1	2	1	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<page-id>/posts	Qafar Xaagu 4-06-2014/ም ----- Website www.tigraitv.com Email tigraimas
1	3	1	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<page-id>/posts	ዜና ትግርኛ ሰዓት 6:30 - 4 ለካቲት 2014 ዓ/ም ----- Website www.tigraitv.com Email
1	4	1	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<page-id>/posts	በቆለ ተምቤን ዮተፈፀመው ግፍ
1	5	1	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<page-id>/posts	ካብ ቤተ ክህነት ቤተ ክርስትያን ኦርቶዶክስ ተዋህዶ ትግራይ ዝተውሃበ መግለፂ
1	6	1	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<page-id>/posts	English News Feb 10, 2022 ----- Website www.tigraitv.com Email tigraima
1	7	1	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<page-id>/posts	ኣማርኛ ዜና 2:00 - የካቲት 3, 2014 ዓ/ም ----- Website www.tigraitv.com Email ti
1	8	1	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<page-id>/posts	ዜና ትግርኛ ሰዓት 1:00 - 3 ለካቲት 2014 ዓ/ም ----- Website www.tigraitv.com Em
1	9	1	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<page-id>/posts	ካብ ቤተ ክህነት ቤተ ክርስትያን ኦርቶዶክስ ተዋህዶ ትግራይ ዝተውሃበ መግለፂ:: ----
1	10	1	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<page-id>/posts	ካብ ቤተ ክህነት ቤተ ክርስትያን ኦርቶዶክስ ተዋህዶ ትግራይ ዝተውሃበ መግለፂ:: ----
1	11	1	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<page-id>/posts	መዋእል ቃልሲ ---
1	12	1	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<page-id>/posts	ዜና ትግርኛ ሰዓት 6:30 - 3 ለካቲት 2014 ዓ/ም --- Website www.tigraitv.com Email T
1	13	1	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<page-id>/posts	የአሜሪካ ኮንግረስ የውጭ ጉዳይ ኮሚቴ ----- በአሜሪካ ኮንግረስ የውጭ ጉዳይ ኮሚቴ በፋሽስቱ #አብ
1	14	1	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<page-id>/posts	ሰብ ሞያ ዓሰምለኸ ሕጊ፣ ኣብይ ኣሕመድ ኣብ ልዕሊ ትግራይ ዝፈፀሞ ገበን ኩናትን ጥሕሰት ሰብኣዊ መሰ
1	15	1	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<page-id>/posts	መሰል ህዝቢ ትግራይ ንምርግጋፅ ደጀንነት ብምጥንሻርን ደቅና ብምስላፍን ፅላእትና ከንመስት ይብባእ -
1	16	1	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<page-id>/posts	ልምዓትና እናልማዕናን ፅላእትና እናመሽትናን ----- ዓቕምና ፀንቂቕና ልምዓት ከንከን ሃፍቲ ተፈጥሮን
1	17	1	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<page-id>/posts	ኣማርኛ ዜና 2:00 - የካቲት 2, 2014 ዓ/ም ----- Website www.tigraitv.com Email ti
1	18	1	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<page-id>/posts	ዜና ትግርኛ ሰዓት 1:00 - 2 ለካቲት 2014 ዓ/ም ----- Website www.tigraitv.com Em
1	19	1	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<page-id>/posts	Qafar Xaagu 2-06-2014 --- Website www.tigraitv.com Email Tigraimassm
1	20	1	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<page-id>/posts	ዜና ትግርኛ ሰዓት 6:30 - 2 ለካቲት 2014 ዓ/ም --- Website www.tigraitv.com Email T

2	5100	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	ከቡር ኣቦና ብትግርኛ ከንብቡ ተሸኒሮም : ቋንቋ ደመኛታትና ከንደየናይ ዘኽክል ፅዕንቶ ከምዝገበረልና ኦ
2	5101	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	ተመስገን ኣምላከየ
2	5102	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	መሾስ ምን ይበላል እንግዲህ ምንም መንግዲህ እርሶና መስሎኟ ኣሑቲ ቤተክርስቲያንን ለማፍረስ ለመቢ
2	5103	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	*ኣሜን ፫ ሚዜ*።
2	5104	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	ትክክለኛ ውሳኔ ነው ! በዛው በዋልድባ ገዳም ውስጥ የሚኖሩ የሶስፋራው ቡድን አበሳት በርካቶች ናቸዉ
2	5105	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	ተመስገን አቦና
2	5106	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	ተመስገን ፈጣሪ ናብ ብረከ ትግራይ ስልጣኑ አቹሉም ንምለስ አለና ይበል #ወከመ_ተሰፍስሂ_ልሰ
2	5107	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	Amen amen Tigray deserve peace prosper in the Tigray homelands
2	5108	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	ተመስገን አሜን አሜን አሜን
2	5109	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	Amen
2	5110	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	Amen Amen Amen
2	5111	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	
2	5112	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	Temesgen Arska mkial eyu
2	5113	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	ካሐልፍ እዩ
2	5114	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	Asmara Kidanemariam Amen
2	5115	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	እምላኽ ይሕግዝ መፈጸምትኡ የርእየና ጽቡቕ ጅማሮ።
2	5116	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	♥♥♥♥♥♥♥♥♥♥♥□□
2	5117	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	ንኩሉ ነገር እግዚኣብሄር ይመስገን እግዚኣብሄር ይተሓወሰ ቃለ ሂወት የስምኣልና አቦና
2	5118	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	ተመስገን ብጹኣን ከቡራት አቦታትና ብዕድመን ጸጋን ይሓልወልና እግዚኣብሔር እምላከ ይሓግዝኩም ከ
2	5119	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	እግዚኣብሔር ይመስገን □□□□
2	5120	5	3757132896403	data	data.*	fetched (200)	2022-02-1	Facebook:/<post-id>/comments	ተመስገን ኣሜን ኣሜን ኣሜን

Appendix B: Dataset Sample

Appendix B.1: Tigrinya Hate Speech Dataset Sample

comments	Label
… ,, @..# (' ኣምላክ ይመስገን[]፣ .ንኩሉ ግዝየ ኣለዎ።""""""" 2345544, ???	normal
mister ዋሕዋሕታ፡ ናይ መሰል ኣሰር ዘትሐዝክና መሪር ቅርሲ ወለዲ ሓሞት ዘምዖግክና እቶን መፍተን ወርቂ ለካቲት ኣዴና 🩹❤️ ✋✋✋	normal
ኣህባይ ካብ እዚኦም ዝመጽእ ለውጢ የለን	hate
ኣኽባ ኣናጩ ፅብላሊዕስያ መዓንኾራ ዘይከደነትስ መሬት ከደነት ይባሃል ኣብ ኣፍሪካ ዞሎ ዘይፈትሑስ ኣብ ፍልስጤም ዞሎ ክፈትሑ ንኽብዶም እምበይ ንሓቂ ዘይተዓደሉ ሓሰውቲ	offensive
እወ እንታይ ዶ ኣትይዎሞ እዩ ንጸጋም ይገድፎዎ እንበር እንታይ ዳኣ ☹️ ዳሓን ግን እግዚኣብሔር ኣምላኽ ባዕሉ ይኣኽለኩም ኢሉ ሰላምና ካውርደልና እዩ ኣጆና :: ካብ ሰብ ፍታሕ እንተዝነበር እማ ኣብ ከምዚ እዋን ኣይምተበፅሐን 😢 ዝሮዝሮ ንዚ ካብ ምጽሓፍ ሱቕ መሐሸኩም መሕረቕ እዩ:: ☹️	normal
ዋኢጋት ተኣከባ ኣይተኣከባ ዘምጽእኦ ለዉጢ የለን ንሞኺኑ ኤርትራ ካብ ማዓዝ እያ ናይ ኣፍሪካ ሕብረት ኣባል ዝኮነት ገና ብዙሕ ክንሰሞዕ ኢና ኣፍሪካውያን ኹሎም ዋኢጋት እዮም	hate
ኣእዱግ ተኣከባን ተበንጨራን ዘምጽእኦ ለውጢ የለን ኩሎም ተዘይኮኑ መብዛሕቲኦም ዲክታተራት እዮም:: ኣፍሪካውያን ኣእዱግ እዮም	hate
ፈጣሪ ደኣ ምሕረቱ ይዉረደና እምበር ካብዞም ኣህባይ መፍትሒ የለን።😔😔 ርጉማን	hate
የሕዝን እዩ	normal
ኩሉ ፈሊጥኩም ተጋሩ ለመንቲ ሓሰውቲ ሙኺንኩም ወረ ነዓኩም ኣፍሪቃ እንታይ ትብጻሓኩም ኣሜሪካ ዘይትቕርበኩምን ለመንቲ ብስርናይ ናብ ኩናት ተኣትወኩም ዘላ	hate
ዓለም ዓውሪታ ሓቂ ዘይትፈትው ፈጣሪ ሰላም ምሕረት የውርደልና ንሃገርና ንህዝብና ትግራይ ትስዕር	normal
ዞም ቀርጫፋት ታይ ኣእትይዎም ድኣ	offensive
ሓቅኺ ኽብርቲ ሓፍትና ኣምላክ ይኣኽለኩም ይበለና እምበር ካብ ኣህዛብ ንቡድል ዝኸውን መፍትሒ የብሎምን ዞም በካዓት	offensive
ካብዞም ፀማማት ምንም ነገር ኣይንፅበን ሻሕራማት ሓኸኽቲ ፀሊምኮ ርእሲ የቡሉን ብቅልፅምና ጥራይ ንቐፅል	hate
ኦ ኣምላኽ ህዝብኻ ሓሉ 🫂	normal
ኣምላክ ሆይ ሓደራኻ ህዝብና ምሕረትካ ልኣኽልና🫂🫂	normal
ኣሜን መሬት ዓድና ብሰላም ይምለስኩም	normal
ኣሜን ኣሜን ኣሜን	normal
ገዛውቲ ተቓጸሉ እምበር ምኽንያት መቃጸሊኡ ንምንታይ ዘይንገር ምኽንያቱ ብቓንጫ ዝስራሕ ገዛውቲ ብቐሊል ክቃጸል ይኽእል እዩ ይኹን እምበር ህዝብና ክንድዚ ግፍዒ ኣምተገብኦን ምናልባት ኣብ ውሽጦም ስደተኛታት መሲሎም ዑሱባት ከይህልዉ ይጠራጠር ምኽንያቱ ኣብ ወርሒ ክልተ ግዜ ማለት ቀሊል ኣይኮንን	normal
ልኡኽ ሰላሊ ከይህሉ ተወዲብኩም ባዕልኹ ንባዕልኹም ሓልው::	normal
ህዝበይ ማዓረይ ከሓልፍ እዩ ኩሉ ፈጣሪ ግልፅ ይበለና	normal
ኣሜን ኣሜን ኣሜን	normal
ትግራይ ትሰረር	offensive
እንኳዕ ደስ በለኩም ተጋሩ ናይዚ ሓሳድ ኣብይ ሚዲያ ዋልታ ቲቪ tv ሃክ /hacked ተገይራ✋	offensive
ድምፂ ህዝብና ትዕፍን ዝነበረት ዋልታ ቲቪ ተሰሪቓ	normal
ሰላም እንኳዕ ዳሓን መፃእኹም ❤️❤️❤️❤️❤️🫂🫂🫂	normal

ዝፈትወኩም ጀጋኑ ኣሕዋት	normal
ንባዓል ኣምባሳደር ካሳ ተከለብርሃን ብጣዕሚ ነይረ ዝፈትዎ	normal
ህዝበይ ማዓረይ ኣጆኻ ክሓልፍ እዩ ኩሉ ትግራይ ትስዕር	normal
ኣየ ህዝበይ ንስኻ ዶ እዙይ ትፍደ ኣሕሕሕ	normal
ተጋሩ ርጉማት ብሓይሊ ምዉሳድ ዘይናትኩም እዩ ነታ ኣምሓረይቲ ብሓይሊ ወሲድኩምላ ትኾኑ ለኻብጥ ሰረቕቲ	hate
ነውሪ ብብሄርኪ ክትኮርዒ ኣለኪ ። ተጋሩ ናይ ኣተሓሳስባ እምበር ናይ ብሄር ጽልኢ የብልናን ።	normal
እንታይ ዘይገበረት እዩሞ እታ ትስማዕ ስምዒት እያ ዝተናገረት እምበር ከምቲ ንሪኣ ዘለና ምንም ዘይትፈልጥ ገባር እያ ዘላ ግን እቲ ዝወረዳ ችግር እና መከራ ግፍዒ ከመይ ከምዘሕምም እታ ሕማማ እያ ዝተናገረት እቲ ውሽጣ ዝተስምዓ ስምዒት ይመስለኒ ትዛረብ ዘላ	normal
Alsa YemaneAlsa Yemane ዝተወለድኩሉ ብሄረይ ንክጽውዖ እውን የጽልአኒ ዩ ምሽ ኢላ ኣነ ከኣ ኣብኡ ተደሪኽ እየ ጽሒፈ	normal
ቆሻሻ ሓሳድ ዘይተረብሕ	offensive
ፅቡቕ ኣለካ ፀሊኦዮ ትብል ኣላ ግን እቲ ውሽጣ ዝተሰምዓ ስምዒት እያ ትዛረብ ዘላ ወገነይ ዝበልካዮ ስብ ዘይተፀበኻዮ ግፍዒ ክፍፅመልካ እንተሎ ይቅረ ንቲ ሰብ ንቲ ኽባቢ እውን ኢኻ ትፀልኦ እና ከምኡ ተሰሚዕዋ ይኽውን ኽቢድ እዩ ፈጣሪ ኣምላክ ብቻ ሰላሙ ይመልሰልና እምበር	normal
ትም በሉ ቶካናት ቕማላት ዓጋመ ሓሰዉቲ	hate
ኣቲ ቆማል ረሳሕ ደጋል ዓጋመ	hate
Sara Fsha ትም በሊ ቆማል ዓጋመ	hate
ኣለናዶ? ደቂ ዓደይ ኽቡራት	normal
እታ ፣ ሎጎ ፣ ፅብቕቲ ፣ ነይራ ፣ ግን ፣ እታ ፣ ናይ ፣ ተባዕታይ ፣ ምስሊ ፣ ልምንታይ፣ ናይ፣ ወዲ ፣ጥራይ ፣ኮይና ፣እተን ፣ዋዕሮታት ፣ ትግራይ፣ ልምንታይ ፣ዝዉክለን ፣ምስሊ ፣ዘይለ? ኣስተኻኽልዋ፣ ናይ፣ ተባዕታይን ፣ናይ ጓል ፣ኣንስተይቲ፣ ምስሊ ፣ሐቢሩ ክቕመጥ፣ ኣለዎ፣ ብህፁፅ ፣ ኣትግብርዋ።!!!	normal
ትግራይ ትዕወት ኣጆኹም ኣቱም ጀጋኑ♥♥♥	normal
ክወግሕ እዩ ህዝበይ😂😂😂	normal
ትግራይ ሰብ ኣለዋ	normal
ተጋሩ እንታይኾን ኣምጸ እዝኹሉ ሓለኽመለኽ ኣብ ክንዲ ኣብ ጀኖሳይድ ምእታዉኹም ኣብዉሽጠኹም ዘሎ ረሳሕ ኣተሓሳባ ምሕጻቡ ምሐሽ መእሰሪ ኾነ መጠርነፊ ዘይብሉ ህውከት ዘእተወኩም ሕወሓት እዩ እሞ ንሶም እዮም ጀኖሳይድ ዝበሃሉ ገዛኹም ኣጽፍፉ ኣዳምነት ተርክብሉ ለኻብጥ	hate

Appendix C: List of Stop Words

Appendix C.1: List of Tigrinya Stop Words

ሀዚ	ለሎ	ሆይ	ለካ
ለለ	ሐዚ	ልብል	ልናይ
ሕዚ	ሕጂስ	ልዚለስ	ልዕሊ
ሕጂ	መበል	መቼም	መን
ማለተይ	መዓዝ	መኣዝ	መንዩ
ማለት	ማለትሲ	ማራ	ምሳኻ
ምስ	ነበሩ	ንየው	ኣቲ
ምስቲ	ነቲ	ኣለካ	ኣታ
ምስታ	ነታ	ኣለኻ	ኣነ
ምሽ	ነቶም	ኣለዉ	ኣነ
ምብሃልና	ነዘን	ኣለዎ	ኣነስ
ምንም	ነዚ	ኣላ	ኣነኳ
ምእንቲ	ነዛ	ኣሎ	ኣንታ
ምኽነ	ነዞም	አረ	ኣዝየ
ምኽና	ነይራ	ኣበይ	ኣየናይ
ምኽን	ነይርና	አብ	ኣይኮነን
ምኽኖም	ነይርወን	ኣብ	ኣይኾነን
ምኽኖም	ነይሮም	ኣብኡ	ኣይኾነይ
ስለ	ናበይ	አብዚ	እህሞ
ስለምንታይ	ናብ	ኣብዚኦም	እማ
ስለዚ	ናትና	አቦ	እምበር
ስለዝኾነ	ናትኩም	ኣቱም	እምበኣር
ስለዝኾነውን	ናትካ	አነ	እምበይ
ስነ	ናይ	ኢለ	እሞ
ቅድሚ	ናይተን	ኢለካ	እሞሲ
በለ	ናይቲ	ኢሉ	እሞዳኣ
በሉ	ናይቶም	ኢላ	እስኪ
በል	ናይዘን	ኢልኒ	እስካብ
በተለይ	ናይዚ	ኢልና	እስኻ
በቲ	ናይዞም	ኢልኪ	እባ
በቶም	ኔረ	ኢልካ	እተን
ብኦኦም	ኔሩ	ኢልካዮ	እቲ

በየናይ	ኔርና	ኢልኻ	እታ
በየን	ኔርካ	ኢሎም	እቶም
በጃኪ	ንሓደ	ኢኻ	እና
በጃኻ	ንሕና	ኢኽን	እናበለ
ባእ	ንመን	ኢየ	እናበሉ
ባዕልካ	ንምባል	ኢዩ	እናበላ
ብቲ	ንምንታይ	ኣለኩ	እንበር
ብቖ	ንሱ	ኣለኩም	እንተ
ብናይ	ንሳ	ኣለካ	እንተለኪ
ብኢሂን	ንስኪ	ኣለኹም	እንተዘየለ
ብከምዚ	ንስካ	ኣለኻ	እንተዝኾና
ብኹሉ	ንስክስ	ኣለዉ	እንተድኣ
ብዘይ	ንስኺ	ኣለዋ	እንታይ
ብዚ	ንስኻ	ኣለዎ	እንዲኺ
ብዛዕባ	ንስኻትኩም	ኣለዎም	እንዲኻ
ብዝኮነ	ንስኻትክን	ኣላ	እንዳ
ብዝኾነ	ንስኻን	ኣሎ	እንድየን
ብጀካ	ንቲ	ኣበይ	እንድዩ
ታማ	ንናይ	ኣቢሉ	እንድያ
ታይ	ንኩሉ	ኣብቲ	እንድዮም
ትበል	ንዓኺ	ኣብታ	እንዶ
ትብሉ	ንዘለዉ	ኣብክንዲ	እንዶሞ
ትብል	ንዘሎ	ኣብዘይ	እኮ
ትብሎ	ንዙይ	ኣብዚ	እኳ
ትኾኑ	ንዚ	ኣብዚኣ	እወ
ነበረ	ንዞም	ኣብዛ	እዉን
እዋን	ከለኹም	ከኣ	ክብልዎ
እዋእ	ከለዉ	ኩለን	ክትብሉ
እዋይ	ከለዋ	ኩሉ	ክትብሊ
እውንዶ	ከሎ	ኩሎም	ክንብል
እዙይ	ከመይ	ኩኑና	ክንኣቱ
እዚ	ከማና	ኩን	ክንዲ
እዚ	ከማን	ኩኖ	ክኣ
እዚኣ	ከማካ	ካበይ	ክኾና

እዚኣስ	ከማኺ	ካባና	ክኾን
እዚኣም	ከማኻ	ካባይ	ኮላይ
እዛ	ከማይ	ካብ	ኮታስ
እዝኹሉ	ከም	ካብተን	ኮነ
እዝይስ	ከምቲ	ካብቲ	ኮንኩም
እዞም	ከምታ	ካብቶም	ኮንኪ
እየ	ከምቶም	ካብዚ	ኮይኑ
እየን	ከምኡ	ካብዛ	ኾይኑ
እዩ	ከምኡውን	ካኣሎ	ኮይና
እዮሞ	ከምዘለኒ	ክህልዎ	ኮይንካ
እያ	ከምዘለኩም	ክሳብ	ኮይንካ
እዬም	ከምዘለኪ	ክሳዕ	ወላ
እዮ	ከምዘለኺ	ክሻብ	ወላኳ
እዮም	ከምዘለዎ	ክብሉ	ወዘተ
ኦ	ከምዚ	ክብል	ወይ
ዝኮነት	ዘይብሉ	ዘለክን	ወይስ
ዝኮኑ	ዝበሃሉ	ዘለኹም	ወዮ
ዝኾነ	ዝበሃል	ዘለወን	ዋእ
ዝደሊ	ዝበለ	ዘለዉ	ዋይ
ዞም	ዝበለና	ዘለዋ	ዓም
ይበለና	ዝባሃላ	ዘለዎ	ዘለኒ
ይበሉ	ዝባሃል	ዘለዎም	ዘለና
ይበል	ዝብል	ዘላ	ዘለኩም
ይብለካ	ዝነበረ	ዘሎ	ዘለኪ
ይብሉ	ዝኮነ	ዘብል	ዘየለ
ይብላ	ዲኺ	ድማ	ገለ
ይኩን	ዲኻ	ድኣ	ገለዶ
ይኹን	ዳኣ	ድዩ	ገዲም
ደኣ	ዳኣሞ	ድያ	ጌረ
ዲካ	ድሕሪ	ድዮም	ግን
ግንኮ	ጥራሕ	ግዳ	