

2021-03

DESIGNING AGGRESSION DETECTION MODEL IN SOCIAL MEDIA FOR AMHARIC LANGUAGE

SHIFERAW, NIGUSSIE

<http://ir.bdu.edu.et/handle/123456789/12401>

Downloaded from DSpace Repository, DSpace Institution's institutional repository



BAHIR DAR UNIVERSITY
BAHIR DAR INSTITUTE OF TECHNOLOGY
SCHOOL OF RESEARCH AND GRADUATE STUDIES
FACULTY OF COMPUTING
COMPUTER SCIENCE

MSC THESIS ON:

**DESIGNING AGGRESSION DETECTION MODEL IN SOCIAL MEDIA
FOR AMHARIC LANGUAGE**

BY

SHIFERAW NIGUSSIE

BAHIR DAR, ETHIOPIA
MARCH, 2021



DESIGNING AGGRESSION DETECTION MODEL IN SOCIAL MEDIA FOR AMHARIC LANGUAGE

By

Shiferaw Nigussie Alamirew

A thesis submitted to the school of Research and Graduate Studies of Bahir Dar Institute of Technology, BDU in partial fulfillment of the requirements for the degree of Computer Science program in the Faculty of Computing.

Advisor Name: Gebeyehu Belay (PhD)

Bahir Dar, Ethiopia

March 24, 2021

DECLARATION

This is to certify that the thesis entitled, **Designing Aggression Detection Model In Social Media For Amharic Language** submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science under **Computing Faculty**, Bahir Dar Institute of Technology, is a record of original work carried out by me and has never been submitted to this or any other institution to get any other degree or certificates. The assistance and help I received during the course of this investigation have been duly acknowledged.

Shiferaw Nigussie



03/23/21

Name of the candidate

signature

Date

© 2021

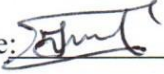
Shiferaw Nigussie

ALL RIGHTS RESERVED

Bahir Dar University
Bahir Dar Institute of Technology-Bahir Dar University
School of Research and Graduate Studies
Faculty of Computing

APPROVAL OF THESIS FOR DEFENSE RESULT

I hereby confirm that the changes required by the examiners have been carried out and incorporated in the final thesis.

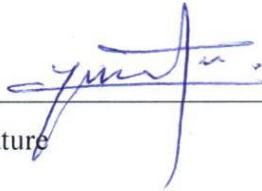
Student Name: Shifieraw Nigussie Alamiraw Signature:  Date: 23/03/2021

As members of the board of examiners, we examined this thesis entitled “**Designing Aggression Detection Model in Social Media for Amharic Language**” by Shifieraw Nigussie Alamiraw. We hereby certify that the thesis is accepted for fulfilling the requirements for the award of the degree of Masters of Science in Computer Science.

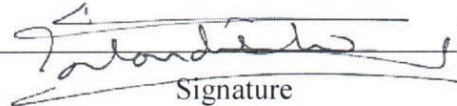
Board of Examiners

Approved By:

Advisor:

Geberghis Belay (Dr.) 
Name Signature Date

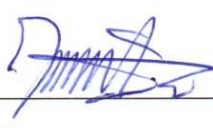
External Examiner:

Wondwossen Mulugeta (PhD)  March 2021
Name Signature Date

Internal Examiner:

Mekonnen Wagaw (PhD)  03/23/2021
Name Signature Date

Chair Person:

Seffu G (Ass. prof)  02/23/2021
Name Signature Date

Chair Holder:

Haileyesus A.  15/07/13 E.C
Name Signature Date

Faculty Dean:

Belete B.  
Name Signature Date

ACKNOWLEDGEMENTS

Firstly, I would like to thank the Almighty God, for his grace, strength and good health as I undertook this research.

Secondly, I would like to thank Dr. Gebeyehu Belay for being my advisor and helping me develop the aptitude and interest towards research. He provided the perfect amount of counseling in deciding the research problem and direction. The thesis would not be possible without his continuous guidance and insights. Working under his supervision was an enlightening experience and his dedication to work and research provided a perfect ideal to work towards. His courses of Machine Learning opened up new methods to work towards.

Thirdly, I would also like to thank “ሊ.ገ ተግባሩ አዳነ”, a linguist expert at Debre Markos University, for his support for data annotation. Thank you for your supports that you gave me even if you have many works to do.

Finally, Special thanks to Dr. Zewdie for providing hint and awareness about aggression detection and supporting me some related resources on the area. I would also like to thank my friends, family and classmates who have support me on along this journey. A special thanks goes out to my dear friend Yihun Alemu, you helped me get through some of the toughest months of my life.

ABSTRACT

As a side effect of increasingly popular social media, aggression has emerged as a serious problem afflicting individuals or groups. However, aggression classification plays an important role in the user's feedback evaluation and decrease the challenge. At present, deep learning techniques such as long short-term memory (LSTM), gated recurrent unit (GRU), bidirectional long short-term memory (BiLSTM), or convolutional neural network (CNN) are prevalently preferred in aggressive comment classification. Moreover, word embedding such as Word2Vec and FastText is closely examined in text for mapping closely related to the vectors of real numbers. In this thesis, we present BiLSTM model with FastText word embedding to classify Amharic Facebook and Twitter posts and comments into non-aggressive, covertly aggressive, and overtly aggressive comments to reduce the challenges of Amharic aggression using deep learning approach. The proposed model extracts features from FastText word embedding and classifies texts in terms of aggression levels. The FastText word embedding capable to handle out-of-vocabulary problems, spelling error for words and another challenging issue lack of efficient in representation of rare words. By comparing, the performance of the proposed word embedding with word2vec, the proposed word embedding better for aggression detection classification performance with 1.23% accuracy difference score. We developed BiLSTM model using FastText word embedding to classify Amharic social media posts and comments into neutral, covertly, and overtly aggressive. We compared with BiGRU and BiLSTM-CNN based on word2vec and FastText word embedding, the proposed model based on FastText embedding performed best with 0.7631, 0.71, 0.71, and 0.71 accuracy, F1-score, Precision and Recall respectively.

TABLE OF CONTENTS

DECLARATION	Error! Bookmark not defined.
ACKNOWLEDGEMENTS	iv
ABSTRACT	v
Table of Contents	vi
LIST OF FIGURES	ix
LIST OF TABLES	x
LIST OF ABBREVIATIONS	xi
CHAPTER ONE	1
INTRODUCTION	1
1.1 Background	1
1.2 Motivation	4
1.3 Statement of the Problem	5
1.4 Objective	6
1.4.1 General Objective	6
1.4.2 Specific Objectives	6
1.5 Scope and Limitations	7
1.5.1 Scope	7
1.5.2 Limitations	7
1.6 Significance of the study	8
1.7 Organization of the Thesis	8
CHAPTER TWO	9
LITERATURE REVIEW	9
2.1 Introduction	9
2.2 Amharic language	9
2.2.1 The Amharic Character Representation	9
2.2.2 Challenge in an Amharic Writing Scheme	10
2.3 Verbal Aggression	11
2.4 Offensive Speech	11
2.5 Hate Speech on Social Media	12
2.6 Verbal Aggression in Social media	12
2.7 Methods for Detecting Aggression	16
2.7.1 Rule Based Approaches	16
2.7.2 Deep Learning Used in Aggression Detection	16
2.7.2.1 Convolutional Neural Networks (CNN)	17

2.7.2.2	Long Short Term Memory networks (LSTM)	17
2.7.2.3	Bi-directional Long Short Term Memory networks (Bi-LSTM)	19
2.7.2.4	Bi-directional Gated Recurrent Unit (Bi-GRU).....	19
2.7.2.5	Attention-based LSTM Networks	20
2.7.2.6	Ensemble Classifier	20
2.8	Word Embedding Used in Aggression Detection	20
2.8.1	Word2vec	21
2.8.2	Glove	22
2.8.3	FastText	22
CHAPTER THREE		25
METHODOLOGY		25
3.1	Introduction	25
3.2	Dataset building	25
3.2.1	DATA COLLECTION.....	25
3.2.2	Dataset Preparation	28
3.2.3	Data Annotation	29
3.2.4	Amharic Text Preprocessing	31
3.2.4.1	Tokenization.....	32
3.2.4.2	Data cleaning	32
3.2.4.3	Token removal	33
3.2.4.4	Normalization of Amharic words	34
3.3	Input representation	39
3.4	Imbalanced learn in deep learning.....	42
CHAPTER FOUR.....		45
EXPERIMENT AND RESULT DISCUSSION		45
4.1	Introduction	45
4.2	Experimental Environment Setup	45
4.3	Implementation of FastText.....	47
4.4	Implementation of Deep learning model.....	51
4.4.1	Library.....	51
4.4.1.1	TensorFlow.....	51
4.4.1.2	Keras.....	51
4.4.1.3	Deep learning Layer	52
4.5	Train-test split	53
4.6	Evaluation Measures.....	53
4.6.1	Accuracy	53
4.6.2	Precision	54

4.6.3	Recall	54
4.6.4	F1 Measure	55
4.6.5	Confusion Matrix	55
4.7	Experimental Results.....	56
4.8	Error Analysis of the Experiment	59
CHAPTER FIVE		60
CONCLUSION AND RECOMMENDATION.....		60
5.1	Conclusions	60
5.2	Contribution of the study.....	61
5.3	Recommendation.....	61
References.....		62
Appendix.....		65
Appendix A: Sample Source Codes		65
Appendix A.1 Sample Code for Word2Vec word embedding.....		65
Appendix A.2 Sample Code for TSNE to visualize FastText word embedding		65
Appendix A.3 Sample Code for Building and Evaluating Models.....		65
Appendix A.4 Sample Code for preprocessing of Amharic text.....		66
Appendix A.5 Sample Code for imbalance learning.....		67
Appendix A.6 Sample Code for model evaluation using confusion matrix.....		67
Appendix B. Sample preprocessing data removing stop word and emoji		68
Appendix C. List of stop words		68
Appendix D. Annotation Guidelines		69

LIST OF FIGURES

Figure 2.1 Bi-directional Long Short Term Memory architecture (Kulshrestha et al., 2020).....	19
Figure 2.2 Model architecture of (A) CBOW and (B) Skip-gram (Mikolov et al., 2013).....	21
Figure 3.2 Preprocessing flow chart	35
Figure 3.3 Amharic Aggression Detection Architecture	41
Figure 4.1 Sample Code for Building FastText word embedding.....	47
Figure 4.2 Visualization of 50,000 words from the fastText vector produced by t-SNE	49
Figure 4. 3 Top 10 similar words in FastText.....	50
Figure 4.4 Sample code and result to evaluate the Model	57
Figure 4.6 Confusion matrix of BiLSTM with FastText and word2vec word embedding.....	58
Figure 4.7 Confusion matrix of BiLSTM-CNN with FastText and word2vec word embedding.	58

LIST OF TABLES

Table 2.1 Amharic Characters Alphabet Example	10
Table 2. 2 Amharic Characters with the Same Sound	10
Table 2.3 Summary of related work in aggression detection research	24
Table 3.1 Summary of social media account lists in the dataset	26
Table 3.2 aggression showing the three categories.....	29
Table 3.3 Example of aggression showing the three categories	31
Table 4.1 Description of the Tools and Python Package Used During the Implementation	46
Table 4.2 Parameters of FastText	50
Table 4. 3 BiLSTM parameters	52
Table 4.5. Results with accuracy metrics evaluation.....	57

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
BiGRU	Bidirectional Gated Recurrent Unit
BiLSTM	Bidirectional Long Short-term Memory
CAG	Covertly Aggressive
CBOW	Continuous bag-of-words
CNN	Convolution Neural Networks
CPU	Central Processing Unit
F1	F1-Score
FN	False Negative
FP	False Positive
KNN	K-Nearest Neighbors
LSTM	Long Short Term Memory
NAG	Non-aggressive
NLP	Natural Language Processing
OAG	Overtly-aggressive
ReLU	Rectified Linear Activation Function
RF	Overtly aggressive
SG	Skip- Gram
SGD	Stochastic gradient descent
SMOTE	Synthetic Minority Oversampling Technique
SNE	t-distributed stochastic neighborhood embedding
SPSS	Statistical Package for the Social Sciences
SSWE	Sentiment Specific Word Embedding
SVD	Singular value decomposition
SVM	Support Vector Machine
TF-IDF	Term frequency-inverse document frequency
TN	True Negative
TP	True Positive
VIM	Variable Importance Measures

CHAPTER ONE

INTRODUCTION

1.1 Background

Social media is a very popular way for people to express any hot news, advertising something, establish communication with a new friend, and sharing what group members feel at the time about the issue. People in Ethiopia also spend a significant amount of time on social media to express opinions, interact with friends and share ideas (Sibhat, 2018). Reported that among other social media outlets Facebook is the predominant one utilization in Ethiopia (Sibhat, 2018). More than four million peoples utilize Facebook in Ethiopia (Sibhat, 2018). However, social media also provide the means for distributing abusive and aggressive content easily. Incidents of aggression are serious and widespread issues affecting increasingly more social media users especially Facebook in Ethiopia (Sibhat, 2018). Aggressive comment can cause anxiety and other unexpected reactions of readers (Chen et al., 2017). Many of them believed that Facebook has a part in intensifying and catalyzing online hate speech in Ethiopia. In Ethiopia, aggression languages are commonly come from an unpleasant conditions such as political extremists, using social media, especially Facebook, as a tool of communication to spread aggressive and hate content. As a result, the rise of racism is disrupting the country's sense of belonging, resulting in diversification and degradation. It actively facilitates hateful material debate and racial tensions between different ethnic groups. To this concern, a political activist who wanted anonymity asserted that social media is contributing a great deal to the extension of aggressive and hate speech in the country (Zewdie & Wang, 2018).

Aggressive language varies greatly, ranging from simple profanity to much more severe types of high toxic language. The more troublesome types of language are hate speech and cyber bullying and the presence of these languages on social media platforms showed in correlation with hate crimes in real life settings (Galery et al., 2018).

The general consensus that hates speech can be defined as a language that targets a group with the intent to be harmful or to cause social chaos and insult targeted at an individual is commonly known as cyber bullying. This targeting is usually done based on some characteristics such as race, color, ethnicity, gender, sexual orientation, nationality or religion (Sahay et al., 2018).

Aggression is aggressive online behavior that uses digital media in a way that is intended to cause harm to another individual or group (Sahay et al., 2018). It is intentional actions (Salawu et al., 2020) performed by an individual or a group of people using digital communication like social media such as sending messages and posting comments are some of them. Aggression can take many forms and definitions in social media. Sometimes, the researchers expressed it as cyber bullying or cyber aggression. This will often have the risk of being targeted or harassed via aggressive language, which may severely impact their online experience and the community in general. Thus, incidents of online aggressive behavior have become a major source of social conflict, with the potential of forming criminal activity.

Furthermore, aggression in social media is targeted to a particular person or group to damage the identity and lower their status. It is often expressed in two ways: directly expressed or hidden in the posts (Madisetty & Desarkar, 2018). According to (Kumar et al., 2018), aggression in social media can be broadly categorized into three groups: Overtly Aggressive (OAG), Covertly Aggressive (CAG), and Non-aggressive (NAG). Overtly-Aggressive (OAG) refers to the texts in which aggression is openly expressed either through the use of specific kind of lexical items or lexical features which are considered aggressive structures. Covertly-Aggressive (CAG) refers to texts which are an indirect attack against the user. Detecting aggression from social media is an intriguing research topic. This is because, it is important for law enforcement agencies which have been covering behind in terms of technology, to collaborate with the relevant people who are currently working in the area of criminal action and solution so that they can detect the usage of social media in spreading aggressive messages to put an end to the discrimination. Aggression also contains hate speech (Singh et al., 2019a).

According to (Zewdie & Wang, 2018), Amharic is the second largest Semitic language in the world after Arabic. However, it is under-resourced in that it has few or no language processing tools and resources. So, using the Amharic language for aggression detection is necessary to take action in this serious issues that affect Internet users in this language. The growth of aggression activities is increasing as equally as the growth of social networks (Srinandhini & Sheeba, 2015). To address this issue researchers implement abusive language detection which is done using the Amharic language for interests of web users and to contribute to the maintenance of the civility of the online space in a more general sense.

In this thesis, we use a deep learning approach to detect aggression for Amharic language in Ethiopian social media in order to address the issue and classify the aggressive text into three classes: “Overtly Aggressive (OAG)”, “Covertly Aggressive (CAG)”, and “Non-aggressive (NAG)”. We use Bidirectional Long Short-Term Memory (BiLSTM) as deep learning method with word embedding (word2vec and fastText). Word embedding is a vector representation of the sequence input that can improve classification performance including LSTM (Galery & Charitos, 2018). BiLSTM is one of deep learning approach that can give an excellent performance when used for text classification (Elnagar et al., 2020). BiLSTM is a powerful recurrent neural network (Sharma & Patel, 2018), which alleviates the problem of gradient diffusion and explosion. BiLSTM has been applied in various text classification tasks, such as sentiment analysis, fake news detection and hate speech detection on social networks with good performance (Jang et al., 2020). It is a robust, widely-used, and high-performing classifier (Jang et al., 2020) (Elnagar et al., 2020). It can be trained easily due to its complex structure and ability to handle vanishing gradient phenomena.

1.2 Motivation

The motivation behind this study is the hot issues aggressive comments which are given by the user in the Amharic language that has a direct negative effect with the social media. The Amharic language has no tool to detect aggression and under resourced languages, social media is bringing its negative consequences to the Ethiopian population than other developing countries (Sibhat, 2018). In fact majority of the world's languages are still under resourced this is particularly true for most African country languages. However, the evolution of the Internet and of social media texts, such as Twitter, YouTube and Facebook messages, has created many new opportunities for creating such tools, but also many new challenges.

Secondly, the scientific study of aggression comments from a computer science point of view is recent, and there is a general lack of data about aggression detection. For this reason, the development and systemization of shared resources, such as guidelines, annotated datasets in multiple languages, and algorithms, is a crucial step in advancing the automatic detection of aggression. In addition, many companies and institutions might be very interested in aggression detection over their platforms. After all, they provide a service which they profit, consequently, they assume public obligations with respect to the contents transmitted. Additionally, users can be interested in blocking discourse with aggression to avoid being exposed to it.

Aggressive messages are prevalent and challenging in the Amharic online community as individuals spread aggressive message hiding behind their screens. The aggressive messages have effects of violence or hate crimes and destroys the lives of individuals, families, communities and the country at large. Therefore, critical importance to monitor and identify instances of aggressive messages, as soon as possible to prevent their spread on social network. Therefore, it is necessary to conduct a research on aggression for the social media for all language to decrease the negative psychological effect of the user.

1.3 Statement of the Problem

The emergence of social media (Mouheb et al., 2019), there has been an exponential increase in the user-generated content being made available over the web. Nowadays, information has the power to reach billions of people within a matter of seconds. This has resulted in good communication and exchange of ideas anywhere at any time. However, with the lack of knowledge, many people use this social media to transmit aggressive and potentially harmful contents intentionally over social media, especially Facebook and Twitter (Abdhullah-Al-Mamun & Akhter, 2019).

Aggression is harassing that happens on the internet through different mediums including on web visits and messages. Racial and religious attacks are the most prominent form of aggression (Fauzi & Yuniarti, 2018). This potentially harmful incidents are the source of crimes and conflicts rising from the places the hash tag originate from, and in turn informing the police as well as warning the community about how much undesirable results it generates. Thus the incidents of aggression and ungratified verbal behavior have not remained a minor nuisance, but have acquired the form of a major criminal activity that affects a large number of people, more harmful and, maybe, uncontrollable (Si et al., 2019). The article by (Yimam et al., 2019) pointed out that the dissemination of aggressive content endangering the democratic rights, jeopardize the long-standing social fabrics and ultimately create political and socio-physiological destruction, unfolding polarized political climate, ethnic violence and displacement in Amharic language. These incidents of aggression intends to damage the social identity of the target person and lower their status and has the ability to cause depression and sometimes even suicides by deteriorating the victim both mentally and physically. Therefore, preventive measures is needed to cope with aggressive text in social media for Amharic language.

Aggression in the Amharic Language content monitor in social network perform through one time interruption of the internet service. In which, blocking public to use social media will harmful that involves with participation in social media for the means of socializing, education and business purpose. On the other hand, instead of blocking the sites, if measures are taken to detect aggression through machine learning, then actions could be held against a certain sector of people rather than the mass public. It is an open secret that the recent widespread aggressive and hate speech call for violence particularly targets individual or group for particular purpose intentionally. As such automatic methods to identify such behavior has become important.

All previous researches on the incident of Amharic aggression detection like Amharic hate speech detection use word2vec word embedding which has the problem of out-of-vocabulary problems and another challenging issue lack of efficient in representation of rare words or phrases (Verma & Khandelwal, 2019) . Also, in the previous work spelling error and sarcastic abuse of the language in the social media doesn't consider.

The research issues are summarized as the following questions.

1. How to develop an aggression detection model for Amharic texts?
2. Which word embedding, FastText or word2vec is better for Amharic aggression detection?
3. To what extent the classification rate is achieved?

1.4 Objective

1.4.1 General Objective

The main objective of this research is to design an aggression detection model in social media for Amharic texts using deep learning.

1.4.2 Specific Objectives

To accomplish the general objective of the study, the following specific objectives are formulated.

- ❖ To develop the dataset using Amharic posts and comments from social media
- ❖ To develop detection models using deep learning algorithms
- ❖ To identify effective features for aggression detection model
- ❖ To evaluate the model designed using the evaluation metrics

1.5 Scope and Limitations

1.5.1 Scope

This research is done on aggression detection for Facebook and Twitter in Amharic posts and comments using BiLSTM method and FastText word embedding as vector space. In this research, we focus on Amharic language texts. We use posts and comments by the users in social media and capture in order to determine aggression level, and classify them into three classes (Overtly aggressive, covertly aggressive and Non-aggressive). Furthermore, since the research focusing on the text posts and comments by the social media users, the video, audio and image will not be taken into account.

1.5.2 Limitations

This study comes across limitations on a different phase of the research process. Since there is a lack of other studies for comparison aggression detection for the Amharic language, also lack share public dataset and model for aggression detection. As a result, this study creates a new dataset. The other constraint of this study is listed as follows.

- Due to the limitation of resources for the dataset annotation process of the dataset was challenging and a lack of aggression related law experts to consult.
- The study only implemented the proposed deep learning classifier and limited to develop and evaluate algorithms not test other ensemble methods.

1.6 Significance of the study

The main beneficiaries of this study are Amharic speaker stakeholders that use social media platforms for their day to day activities. Another important application of the detection of cyber aggression is the detection of cyber extremism, cybercrime and cyber hate propaganda. On the one hand, Social media platforms benefit by taking this work results as input for developing better aggression detection or monitoring models for the Amharic language on their platform, because they are struggling for developing aggression detection or abusive content moderation system that can support most language used on the platform. Also, it helps the user to be protected from aggressive content during the time they spent on this social media platform. Additionally, researchers can replicate the proposed research for other languages used in Ethiopia and can serve as a baseline for related work on this issue or use the dataset to improve the research. Considering the gravity of the consequences that cyber aggression has on its victims and its rapid spread amongst internet users specially kids, there is an imperious need for research aiming at understanding how cyber aggression occurs, in order to prevent it from escalating. Also understanding specificity when it comes to types of aggression and aggressive behavior can help with implementing appropriate countermeasures to those who demonstrate aggressive behavior with in their speech and behavior on social media platform.

1.7 Organization of the Thesis

The remaining parts of this thesis are organized as follows. Chapter Two presents a literature review and related work on aggression detection. Chapter three presents explain a details description of the architectural design and components, and network model of the proposed system. Chapter Four presents the experimental results and corresponding interpretation of the proposed system. Finally, chapter Five concludes the thesis with the research contributions and future works.

CHAPTER TWO LITERATURE REVIEW

2.1 Introduction

Due to the numerous victims of aggression and the fatal consequences it causes, today there is a need to study this phenomenon in terms of its detection and prevention. The consequences of aggression are worrisome when victims cannot cope with the emotional stress of abusive, threatening, humiliating, and hate messages. Presently there are several types of research to avoid or reduce online violence; even so, it is necessary to develop more precise techniques or online tools to support the victims on different languages. The problem is more for the language which have very few computational linguistic resources like Amharic language.

2.2 Amharic language

Amharic is the second most widely spoken Semitic language next to the Arabic language and spoken around a 27 million estimated population (Emuye & Tesfa, 2020). Despite the relatively large number of speakers, Amharic is still a language for which very few computational linguistic resources have been developed for the language (Zewdie & Wang, 2018), (Kenenisa & Melak, 2019)

2.2.1 The Amharic Character Representation

Amharic utilizes Geez characters; the characters trace back to 4th century A.D. The first forms of the Geez script included only consonants, while the subsequent variants of the characters represent phoneme pairs of consonant-vowel. Like Geez, Amharic writing uses characters formed by a consonant-vowel combination. In Amharic, seven vowels are used, each in seven distinct forms that reflect the seven vowel sounds they are ኢ ፣ ኦ ፣ ኡ ፣ ኣ ፣ ኤ ፣ ኦ ፣ ኧ. There are 33 basic characters with seven forms representing a consonant and a vowel at the same time, which makes the Amharic script pronounced in the syllable.

Table 2.1 Amharic Characters Alphabet Example

	a/e	u	i	a	e	a/ e	o
h	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
l	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
h	ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ

2.2.2 Challenge in an Amharic Writing Scheme

Amharic writing scheme has some issues that are difficult to process Amharic text. One of these challenges is the redundancy of characters used in Amharic, more than one character to represent the same sound. The various forms have their meaning in Ge'ez, but there is no clear rule that shows their purpose in Amharic (Worku, 2013). The problem of the same sound with various characters is not only observed with core characters but also exhibited in the same order of characters. Those are, ሀ and ሃ; ሐ and ሓ; ኀ and ኃ; አ and ኣ. A word formed by using this character has the same meaning. For example, the word 'sun' could be written in a different way like ጸሀይ, ፀሀይ, ጸሃይ, ፀሃይ, ጸሐይ, ፀሐይ, ፀሣይ.

Table 2. 2 Amharic Characters with the Same Sound

Characters	Other forms of characters	Other
ሀ (he)	ሐ and ኀ	ሃ ፣ ሓ ፣ ኃ
ሠ (se)	ሰ	
አ (a)	ዐ	ኣ ፣ ዓ
ጸ (tse)	ፀ	

2.3 Verbal Aggression

Verbal aggression is a fairly broad term with many definitions, but there is no clear definition of what aggression is and what is not (Potapova & Gordeev, 2016). For this reason, many researchers study expression of aggression both on all levels nowadays. In this thesis, we adopt the definition of verbal aggression based on the work of (Tommasel et al., 2018) and (Kumar et al., 2018). Verbal aggression is any kind of linguistic behavior which intends to damage the social identity of the target person and lower their status and prestige (Kumar et al., 2018). Verbal aggression could be understood as intentional harm delivered via internet to a person or a group of people who perceive such acts as offensive, derogatory, harmful, or unwanted involving rude, insulting, offensive, teasing or demoralizing comments through on line social media that target educational qualifications, gender, family or personal habits to attack the self-concepts of other people instead of, or in addition to, their positions on topics of communication (Tommasel et al., 2018) . Aggression also include hate speech and offensive texts (Haddad et al., 2020). Verbal aggression could be classified into two basic types based on how it is expressed i.e. overt aggression and covert aggression (Kumar et al., 2018). While these incidents are still isolated and do not represent the norm, their gravity demand deeper understanding. Additionally, cyber aggressive comments make their targets feel demoralized and frustrated, thus acting as a barrier for participation and socialization.

2.4 Offensive Speech

Offensive speech can be defined as a language or expression that offends and negatively characterizes an individual or group of people. This kind of speech causes someone to feel upset, annoyed, insulting, angry, hurt, and disgusting. The difference between hate speech and offensive language is often based upon understated linguistic distinctions (Mouheb et al., 2019). Offensive speech occurs when there is a low degree of hate speech characterizations occurs. The speech may contain a target but do not direct incite violence, attack or diminish based on the target group characters because people often use offensive language to make a point in the debate, on heated conversation, and condemn another violent act performed by other people. Any speech that contains a sarcastic, mocking, and joke that offend can be considered offensive if it conducts using language that contains high negative, abusive, dirty words or phrases in both oral and text.

2.5 Hate Speech on Social Media

Social media technology makes people able to express their opinion, including hate speech, quickly, then spread widely and become viral if the topics covered are interesting with different people from every race, religion, and nationality. Social media sites such as Twitter and Facebook have connected billions of people and allowed them to share their ideas and opinions instantly. That being said, there are several hostile consequences as well such as online harassment, trolling, and hate speech (Chetty & Alathur, 2018). Hate speech is any communicative acts that used to express hatred towards a person or a group based on some characteristic such as race, ethnicity, gender, sexual orientation, nationality, religion, or other characteristics (Fauzi & Yuniarti, 2018). These online platforms are often exploited and misused to spread content that can attacks or diminishes, that incites violence or hate against groups or individuals. The anonymity and mobility of social media platform features for security and privacy, enable their users to hide their true identity behind the screen and express or spread hateful content than they might otherwise.

2.6 Verbal Aggression in Social media

Aggressive speech threatens the peaceful social life, the long-lasting unity of people and even may lead to a massive massacre between ethnic as well as religious groups if not managed by regulation. Therefore the need for a regulation to govern hate speech in Ethiopia is very critical and timely (Dibaba, 2019). However, the criminal law is criticized as being poorly drafted with profound implications for human rights in general and freedom of expression as well as the right to privacy in particular. Criminal law is also blamed for confusing social media with the conventional media and how can control this hate speech. It also fails to impose clear criminal responsibility on hatred social media users and many other vague and confusing even unseen scenarios that should seriously be considered (Abraha, 2019). So the work is important to control these type of incidents.

Evaluating past research on cyber aggression detection through deep learning and other related research will give a better understanding of the best approach for this study. This section presents a comprehensive review of basic related works to the area of automatic aggression detection on social media to clearly understand the general technique, method, and result of existing studies. Cyber aggression detection has captured the interest of researchers in the last years due to its proliferation across social media and its harmful effect on people (Singh et al., 2019a) .

There have been several studies on computational methods to detect aggressive language disseminated on social media in the last few years. Different studies fight against aggression by supporting the classification of situations, topics, or types of it with different angles, considering the various causes leading to it, and taking into account its various forms of manifestation. The study (Gordeev, 2016) aimed at learning ways of automatic detection of verbal aggression for the most popular American (4chan.org) and Russian (2ch.hk) image boards. The study material consists of 1,802,789 messages. The machine learning algorithm word2vec was applied to detect the state of aggression. A decent result is obtained for English (88%) and 59% for the Russian to classify the message as aggressive or not aggressive. Other authors, covers implementations of various models. According to the results, they select three classifiers: K-Nearest Neighbors (KNN), SVM, and SGD Logistic Regression which have good performance worthy enough for further experiments to classify aggression into aggressive and unaggressive and they got 0.84 F-score.

Likewise, in Spanish-language, (Gutiérrez-Esparza et al., 2019) proposed a model to classify cyber-aggression cases applying machine Learning and detect offensive comments on social media, in order to intervene by filtering. They applied Random Forest, Variable Importance Measures (VIMs), and OneR to support the classification of offensive comments in three particular cases of cyber-aggression: racism, violence based on sexual orientation, and violence against women and got 90% accuracy. Also, (Sharma & Patel, 2018) worked on the research for toxic comment classification using neural networks and machine learning. They classify the level of toxicity into 6 labels: toxicity, severe toxicity, obscenity, threat, insult or identity hate with CNN and RNN. They got accurate and promising results using LSTM with word embedding 98.77% accuracy.

In 2018 on the Aggressive detection track of Authorship and Aggressiveness Analysis on Twitter, (Enrique Muñiz Cuza et al., 2018) focuses on the detection of aggressive comments in tweets that come from Mexican users to determine whether a tweet is aggressive or not. Their approach is an Attention-based Long Short-Term Memory Network Recurrent Neural Network where the attention layer helps to calculate the contribution of each word towards targeted aggressive classes. In particular, they build a Bidirectional LSTM to extract information from the word embedding over the sentence, then apply attention over the hidden states to estimate the importance of each word and finally feed this context vector to another LSTM model to estimate whether the tweet is aggressive or not. The experimental results show that their model achieves outstanding results.

Also, (Julian & Ralf, 2018) proposed a deep learning-based with ensemble based system to classify an aggression of input post(comment) into one of three classes, namely, Overtly Aggressive, Covertly Aggressive, and Non-aggressive. Their approach uses three deep learning methods, namely, Convolutional Neural Networks (CNN) with five layers (input, convolution, pooling, hidden, and output), Long Short Term Memory networks (LSTM), and Bi-directional Long Short Term Memory networks (Bi-LSTM). A majority voting based ensemble method is used to combine these classifiers (CNN, LSTM, and Bi-LSTM). They trained the method on Facebook comments dataset and tested on Facebook comments (in-domain) and other social media posts (cross-domain). Their system achieves the F-score (weighted) of 0.604 for Facebook comments.

Recently, (Sazany & Budi, 2018) identified abusive and offensive language using deep learning approach, Long Short-Term Memory (LSTM) with word embedding. The experiment result shows that LSTM with word embedding can increase the F1-Score from the previous work until 19.44%, from 70.06% to 83.68% for Indonesian language. Aggression detection model (Singh et al., 2019b) designed to classify the aggression to one of the three predefined categories: “Overtly Aggressive”, “Covertly Aggressive” and “Non-Aggressive” using deep neural networks from Facebook post/comment for Hindi-English Code-Mixed text. They wanted to use the dataset for experimenting with different approaches to make the most top-level classification of aggression on social media. They experimented with 6 classification models and their CNN model on a 10 K-fold cross validation gave the best result with the prediction accuracy of 73.2% and performed comparably to the state-of-the-art algorithms of the time including emoji image with text.

Some researches on the incident of aggression like hate speech detection has been done with aid of machine learning approach. However, all works in Amharic does not consider the problem of spelling error and sarcastic abuse of the language in the social media. Until now, most of the research has focused on resource rich languages like English while resource-poor languages could not gain the attention of the researchers because of the lack of language resources like annotated datasets. With the advent in the algorithms in the fields of Machine Learning and AI, the researchers are equipped with the right set of computational methods, which can be used to detect and report aggressive contents in Social Media Networks. Detection of aggression is predominantly a classification problem. To solve this we need good data sets which are labeled (Kumar et al., 2018).

Amharic Hate speech (Zewdie & Wang, 2018) explored hate speech detection for the Amharic language in social media and the classification of hates into two categories (hate and not hate) using machine learning approach. Authors use Random forest and Naïve Bayes by means of Word2Vec and TF-IDF feature selection. To train the model, they used comments with hate speech words from Facebook. The best classification performance was achieved by Word2Vec with 79.83% accuracy using Naïve Bayes.

The recent thesis study from (Kenenisa & Melak, 2019) also did great effort in classifying texts written in Amharic, focusing on hate contents. The label was made into three, i.e. Hate, Offensive and Neither. Using combination of n-gram features, word2vec and two classical machine learning classifier i.e. Support Vector Machine, and Random Forest Decision Tree, the system reached 75.39 % accuracy with RF classifier.

Also (Emuye & Tesfa, 2020) proposed a deep learning-based with BiLSTM based model to classify Amharic hate speech of input post(comment) into one of four classes, namely, Hate, Offensive, both Hate and Offensive and Neutral. Their approach uses four deep learning methods, namely, BiLSTM, CNN-LSTM, LSTM and GRU. They trained the method on Facebook and Twitter comments dataset. Their system achieves the F-score (weighted) of 0.89 with BiLSTM using word2vec word embedding.

2.7 Methods for Detecting Aggression

In the wider field of natural language processing, the identification of aggressive content can be considered as a classification issue. So basically the process of detecting the aggression can be approached in two different ways, one is rule based classification and the other one is a deep learning based classification algorithms.

2.7.1 Rule Based Approaches

In rule based process the classification is done based on some handcrafted linguistic rule. This approach is also known as the lexical approach. There are a considerable amount of works have been done in the classification of aggression a rule based approach including (Chen et al., 2017) which proposed to rephrase profanity in Chinese text. The authors have framed 29 rephrasing rules. In rule based systems, one needs to prepare a set of list of words. Lexicon-based features could also be included, such as lists of hate or aggressive words. Then on receiving a new document, it can be classified as the type from which the document has the maximum number of words. The advantage of this process is, it is comprehensible and can be improved over time. But the major disadvantage with this approach is, it lacks speed and scalability is always an issue. In the era of social media networks, where the data received are characterized by the properties of big data, if an approach cannot provide speed and scalability, the approach becomes one of those which are preferred least.

2.7.2 Deep Learning Used in Aggression Detection

Deep learning techniques have made impressive advances in many fields. Following this trend, recent NLP research is now increasingly focusing on applying deep learning architectures based on dense word vector representations like word2vec and fastText. In particular, LSTM and its various models are one of the main stream architectures for aggression detection since they can handle sequences of any length and capture long-term dependencies (Nikhil et al., 2018) .

2.7.2.1 Convolutional Neural Networks (CNN)

Convolution Neural Networks (CNNs) are special neural network models that are built to find relationships and patterns between data items according to their relative position. It was mainly used for classifying imagery data. However, CNNs have been achieving excellent results in various NLP tasks also. CNN is useful for extracting and classifying features because it can pass values to the next layer without losing spatial information. These features of CNN are suitable for utilizing spatial information such as the semantic similarity between words in a sentence (Nikhil et al., 2018). CNN consists of input, multiple hidden, and output layers. Different authors achieved the best results using a convolutional neural network with fastText or word2vec word embedding (Potapova & Gordeev, 2016), (Galery & Charitos, 2018). Galery & Charitos, (2018) identified abusive language in the English portion of the corpus as their initial survey found code-switching between Hindi and English. To address this code switching, they used fastText embedding for both languages and a SVD transformation of these two sets of embedding are to generate multilingual embedding's using sub-word units. There are two channels in CNN architecture: static and non-static. In a static channel, the word vectors are static and are not changed whereas in the non-static channel, the word vectors are tuned according to the task. Word vectors are low dimensional dense representations of one hot vectors. These word vectors or word embedding can be treated as universal feature extractors i.e., they can be applied to any task and achieve comparable performance.

2.7.2.2 Long Short Term Memory networks (LSTM)

Recurrent Neural Networks (RNNs) is a type of deep neural network designed for sequence modeling. LSTM is one type of RNN. These kinds of models are greatly studied due to their flexibility in capturing nonlinear relationships. However, traditional RNNs suffer from problems known as exploding or vanishing gradients and, therefore, have difficulty in capturing long-term dependencies. Long Short-Term Memory networks (LSTM) are one of the most used in Natural Language Processing (NLP) to overcome this limitation. They are able to learn the dependencies in lengths of considerably large chains. Previous works demonstrate that LSTMs have achieved remarkable performance in natural language processing tasks. This structure is naturally suited for many NLP tasks; hence, LSTMs have become increasingly popular for NLP applications in recent years. Many research studies use LSTMs to extract sentiment information from text data.

In conventional neural networks, we assume that all inputs are independent of each other. However, this assumption may not be valid for several problems where the input is inherently sequential, or there are explicit dependencies between the input segments. For example, to predict a next word in a sentence, it is better to know the previous words in the sentence. Recurrent Neural Networks (RNN) can predict the next word in the sentence based on the previous words which come before it. RNNs work well if the context is short. If the context is long RNNs may not work properly because they cannot remember the long dependencies and also because of vanishing gradient problem. To overcome the drawbacks of RNNs, Long Short Term Memory networks (LSTM) were introduced. These are special kind of RNNs which are capable of learning long-term dependencies and long contexts. LSTMs have the capability to add or remove the information to the cell state C_t with the help of structured gates. The first component of LSTM is forget gate. The function of forget gate is to decide which information to remember and which information to forget. Sigmoid activation function is used in this layer.

2.7.2.3 Bi-directional Long Short Term Memory networks (Bi-LSTM)

Bi-directional Long Short Term Memory networks (Bi-LSTM) use two LSTMs instead of one LSTM for training. The first LSTM is applied on the normal sentence and the second LSTM is applied on the reversed copy of the given input sentence. This can provide faster learning and additional context for better training.

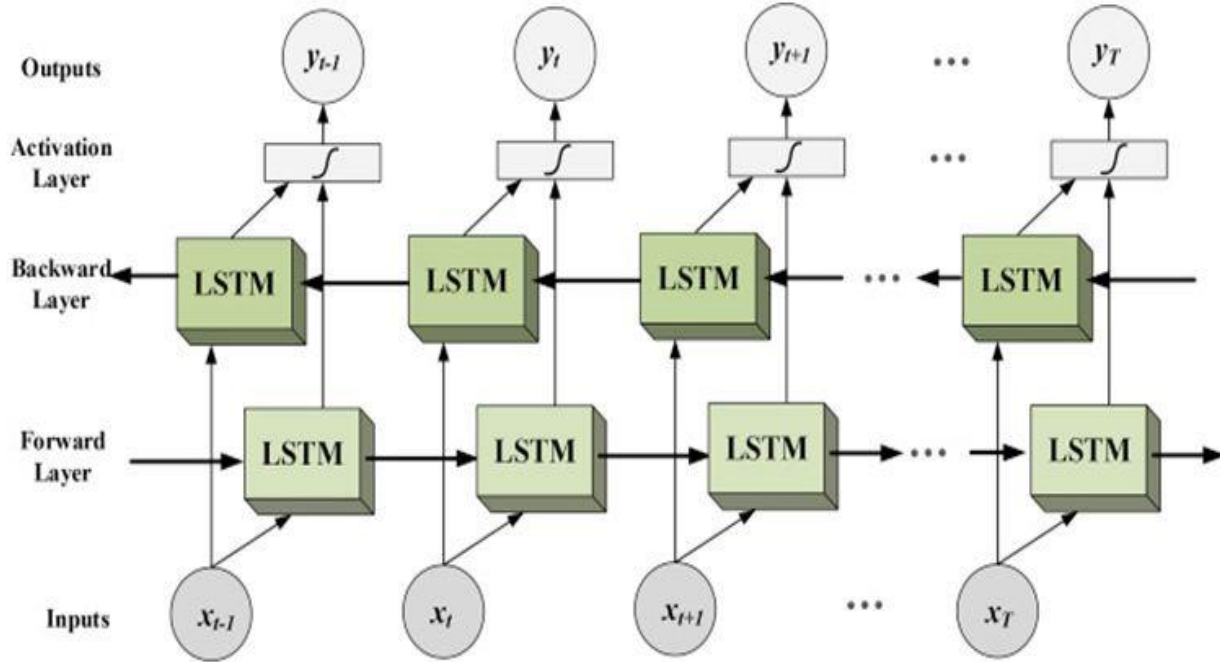


Figure 2.1 Bi-directional Long Short Term Memory architecture (Kulshrestha et al., 2020)

2.7.2.4 Bi-directional Gated Recurrent Unit (Bi-GRU)

The idea of BiGRU is to split the regular GRU neurons into a forward state and a backward state. Based on the co-occurrence information with seed sentiment words, we could enlarge the original seed words iteratively. Besides, BiGRU connects two hidden layers with opposite transmission directions to the same output layer, so that the output layer obtains information both from past and future states. Therefore, the BiGRU is able to learn the information from two different directions, and make more accurate analysis of the e-commerce product quality reviews based on clause-rules recognition rules

2.7.2.5 Attention-based LSTM Networks

A recent trend in deep learning has emerged in attention mechanisms. Attention has resulted in a lot of successes in various NLP tasks because it can be viewed as a method for making neural networks learn better by letting the network know where to look as it is performing its tasks (Nikhil,, Pahwa, Nirala, & Khilnani, August 25, 2018). They deploy an LSTM model with an attention unit over it. In this regard, attention-based LSTM models have achieved exciting advancements.

2.7.2.6 Ensemble Classifier

There are different ensemble methods such as bootstrap aggregating (bagging), boosting, stacking, simple averaging, majority voting, etc. to combine the classifiers. In bagging ensemble method, each classifier is trained on random sub samples which are drawn from the original dataset using the bootstrap sampling method. In boosting, each classifier is trained on the same samples but the weights of the instances are adjusted according to the error of the last prediction. In stacking, the classifiers are combined using another machine learning algorithm. In simple averaging, the predictions of classifiers are averaged to get the final predictions. Finally, in the majority voting ensemble method, for each test instance, the final prediction is the one, which is predicted by the majority of classifiers. Generally, the performance of the ensemble method is often better than the performance of individual methods in the ensemble.

2.8 Word Embedding Used in Aggression Detection

Word embedding is a numerical vector representation of each word in a text as a dense vector of typically a few hundred dimensions, such that when projected onto a vector space, words with similar meanings have similar vector representations and closer together while words with different meanings are more far apart. These vector representations of words can be trained on a large amount of unlabeled data and achieved by learning the vector representation of a word through the context in which it appears. Word vectors used as initializations for the word vectors of deep learning models with such semantic relationships that can improve many existing NLP applications (Mikolov et al., 2013). Currently, Word2Vec, Glove and FastText are the most widely used tools for learning word embedding.

2.8.1 Word2vec

Word2vec is a method for learning the meaning of words by representing them in vector space. Word2vec uses a two layered neural network to learn vector representations of words. It looks at neighboring words to determine which words are similar. Word2vec is an unsupervised algorithm that allows finding semantic relations and distances between words without any annotation or other data preprocessing. It is a predictive embedding model. Nowadays this method is considered to be the best for determining semantic relations between words.

Two main Word2vec architectures that are used to produce a distributed representation of words (continuous bag of words (CBOW) and continuous skip gram). Continuous bag of words (CBOW) — the order of context words does not influence prediction (bag-of-words assumption). In the continuous skip-gram architecture, the model uses the current word to predict the surrounding window of context words. Continuous skip-gram weighs nearby context words more heavily than more distant context words. While order still is not captured each of the context vectors are weighed and compared independently to find word representations that are useful for predicting the surrounding words in a sentence or a document (Bojanowski et al., 2017).

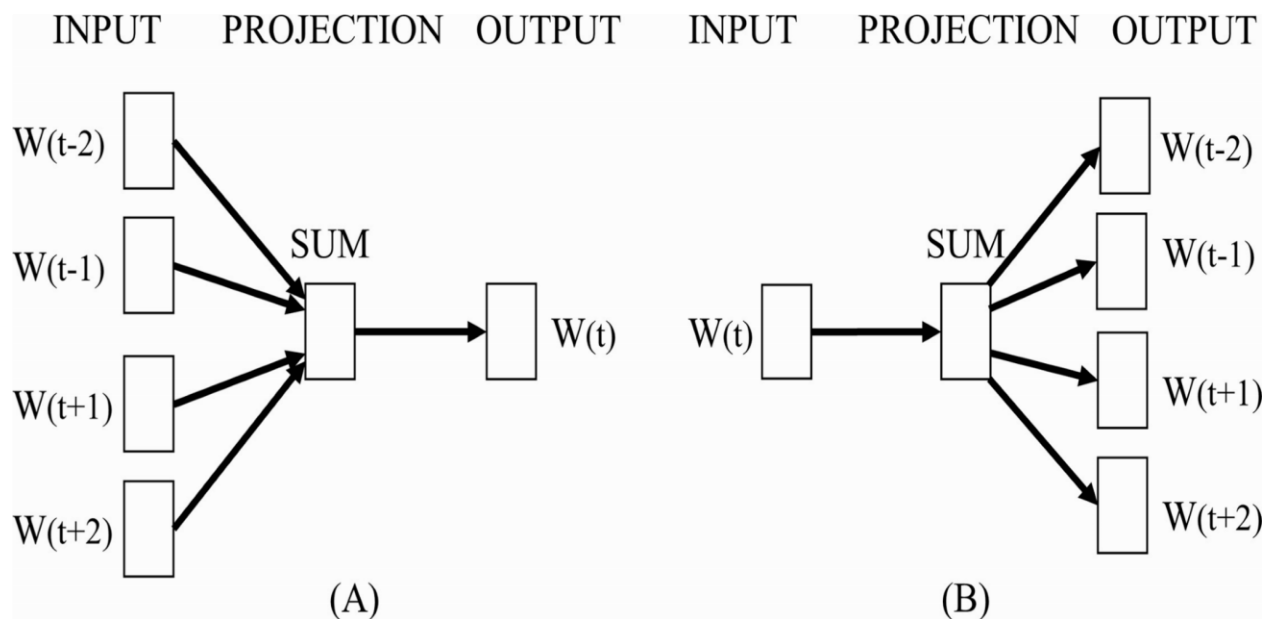


Figure 2.2 Model architecture of (A) CBOW and (B) Skip-gram (Mikolov et al., 2013)

2.8.2 Glove

Both CBOW and Skip-Grams are “predictive” models, in that they only take local contexts into account. Word2Vec does not take advantage of global context. Glove embedding by contrast leverage the same intuition behind the co-occurring matrix used distributional embedding, but uses neural methods to decompose the co-occurrence matrix into more expressive and dense word vectors. While Glove vectors are faster to train, neither Glove nor Word2Vec has been shown to provide definitively better results rather they should both be evaluated for a given dataset. In contrast both Word2vec and Glove leave unseen words as out-of-vocabulary words.

2.8.3 FastText

Facebook released an embedding model that recently attracted a great deal from the machine learning community especially for morphologically rich languages called FastText in 2016. The main goal of the FastText embedding is to take into account the internal structure of words while learning word representations; this is especially useful for morphologically rich languages like Amharic, where otherwise the representations for different morphological forms of words would be learnt independently. The semantic and syntactic information of words that are contained in these vectors make them powerful features for NLP tasks. One issue FastText criticized is its memory consumption to construct character n-gram level, it takes longer to generate FastText embedding compared to word2vec model. However, word2vec and Glove treats each word in corpus like an atomic entity and generates a vector for each word. In contrast, FastText treats each word as composed of character n-grams. So the vector for a word is made of the sum of these characters n grams. With this manifest it benefits to generate better word embedding's for rare words. Moreover, FastText predict vector for out of vocabulary words from its character n-grams even if the word doesn't appear in the training corpus. So, with this intuition, we proposed to use FastText as a word vector generating model for this work.

As it is an extension to the Word2Vec model, FastText also has two architectures for computing word representations called Skip-gram and CBOW (continuous-bag-of-words). The Skip-gram model learns to predict a target word given a nearby word. On the other hand, the CBOW model predicts the target word according to its context. The key difference between FastText and Word2Vec is the use of n-grams. Word2Vec learns vectors only for complete words found in the training corpus. FastText, on the other hand, learns vectors for the n-grams that are found within each word, as well as each complete word. For morphologically rich languages like Amharic treating each variety of words as n-grams is an effective approach and it is summarized as table 2.3.

Table 2.3 Summary of related work in aggression detection research

Authors	Year	Title	Method	Classification	Result	Limitation
Emuye & Tesfa	2020	Amharic text hate speech detection in social media using deep learning approach	BiLSTM, CNN-LSTM, LSTM, GRU	Hate, Offensive, both Hate and Offensive and Neutral	0.89 F-score with BiLSTM	-Not good for infrequent word and spelling error -Out-of-Vocabulary problem
Zewdie & Wang	2018	Hate speech detection for Amharic language in social media	RF and NB by means of Word2Vec and TF-IDF	Hate and not hate	79.83% Accuracy with RF using word2vec	-Not good for infrequent word and spelling error -Out-of-Vocabulary problem
Kenenisa & Melak	2019	Hate speech detection for Amharic language on social media using machine learning techniques	SVM and RF with word2vec	Hate, Offensive and Neither	75.39 % accuracy with RF classifier	-Not good for infrequent word and spelling error -Out-of-Vocabulary problem
Sharma & Patel	2018	Toxic comment classification using neural networks and machine learning	CNN and LSTM with word2vec	toxicity, severe toxicity, obscenity, threat, insult or identity hate	98.77% Accuracy With LSTM	-Not good for infrequent word and spelling error -Out-of-Vocabulary problem
Julian & Ralf	2018	Aggression identification using deep learning and data augmentation	LSTM,CNN, BiLSTM and ensemble method with word2vec	Overtly Aggressive, Covertly Aggressive and Non-Aggressive	0.604 F-score	-Not good for infrequent word and spelling error -Out-of-Vocabulary problem
Singh et al.,	2019	Aggression detection on social media text using deep neural networks	CNN,LSTM, SVM, MLP , Decision Tree with word2vec	Overtly Aggressive, Covertly Aggressive and Non-Aggressive	73.2% accuracy with CNN	-Not good for infrequent word and spelling error -Out-of-Vocabulary problem
Sazany & Budi	2018	Identify abusive and offensive language in Indonesian Twitter	LSTM with FastText	Not abusive, abusive but not offensive and offensive language	0.8368 F-score	-One direction only

CHAPTER THREE METHODOLOGY

3.1 Introduction

This chapter explains a research methodology to build a dataset, design and implementation of Amharic aggression detection to achieve research objectives and answer the research question. The first step is collecting and preprocessing the dataset. Second is transforming the texts into vectors that are comprehensible by the deep learning algorithms. Third is building the prediction model that is a deep learning model. Finally, is the evaluation of the model in detecting aggressive texts? The following section in this chapter explains and justifies the methodology used in conducting the study on aggression detection for Amharic languages.

3.2 Dataset building

3.2.1 DATA COLLECTION

To classify the aggression level across Facebook and Twitter for Amharic language users, we have built a corpus of users' comments and replies in different groups or pages retrieved from social media public pages of Ethiopian Broadcasting pages, Political Party, Politician and Government Office Official Pages, activist, Religious Media and Religious Group Pages, Bloggers and Journalists pages and groups. In order to collect the post and comments, a list of the Facebook and Twitter public pages is gathered based on the content this page disseminates to the public view to have a diverse source of pages and have a representative dataset such as Fana broadcasting corporation, Amhara mass media agency, ESAT tv, EBS tv, Fana tv, OMN tv, Mereja.com, Abbay media, Berera newspaper, Dire Tube, VOA Amharic, ASRAT tv, Asrat mediahouse, Abbay media, Wolkayt Tegedie Amhara Revolution, Yehabesha.com, EPRDF Official, Amhara Democratic Party /ADP/ CC , FDRE Communication Affairs Office Ethiopia, Amhara Press, Bete Amhara, Amhara Media Service, Tamagne Beyen, Pm Abiy Ahmed-daily news, Jawar Mohamed page, Eskinder Nega pages, ፤ህግና ፍትህ በኢትዮጵያ(higina fitihi be'itiyēp'īya), የአማራ ወጣቶች ማህበር በጎንደር(ye'āmara wet'atochi mahiberi begonideri), Mahiberekidusan page, የአማራ ብሔራዊ ንቅናቄ National Movement of Amhara , Soccer Ethiopia ET , Minister of Transport, FDRE Et , Tigray tv, Gamo media group, social pages collected from February 2020 to July 2020. These pages typically post discussions on political, social, economic, religious, and environmental topics and issues.

Table 3.1 Summary of social media account lists in the dataset

Social media account	Number of data in the dataset
ESAT tv	400
EBS tv	420
Fana tv	400
OMN tv	400
Mereja.com	275
Abbay media	350
Berera newspaper	430
Dire Tube	215
VOA Amharic	225
ASRAT tv	200
Asrat mediahouse	200
Wolkayt Tegedie Amhara Revolution	245
Yehabesha.com	300
EPRDF Official	300
Amhara Democratic Party /ADP/ CC	260
FDRE Communication Affairs Office Ethiopia	300
Amhara Press	400
Bete Amhara	400
Amhara Media Service	400
Tamagne Beyen	270
Pm Abiy Ahmed-daily news	220
Jawar Mohamed page	340
Eskinder Nega pages	420
ህግና ፍትህ በኢትዮጵያ	330
የአማራ ወጣቶች ማህበር በጎንደር	300
Mahiberekidusan page	360
Tigrai tv	270
Gamo media group	250
የአማራ ብሔራዊ ንቅናቄ National Movement of Amhara	300
Amhara Mass Media Agency	500
Soccer Ethiopia ET	220
Minister of Transport, FDRE Et	100
Total	10000

We used Facebook and Twitter comments for this study for the following reasons:

1. Facebook and Twitter is used by different users to express their opinions about different posts which can be a valuable source to estimate the opinions of people.
2. The number of users is increasing day by day and diversity of opinions is also increasing.
3. Facebook and Twitter provides different ways to create a page, group or community which may include education, political, business and others.

There are several sampling metrics or criteria that can be used to select these pages or a user on a social media platform. This study considered the following sampling criteria and metrics used for selecting a public page from the categories:

- Active public pages included in categories which uses Amharic language frequently
- The page that posts news or hot issues on politics, ethnicity, religion, or gender daily.
- The page that uses the Amharic language most frequently for posts and comments. By doing so, authors could capture both casual conversations and politically aggressive posts and comments

3.2.2 Dataset Preparation

After the data collected, a preparation process follows, which are collecting, cleaning, filtering, and consolidating data into one file or data table. This process used preparation tools such as MS Excel and others. Filtering and Cleaning the data primarily use for the next stage, which is the annotation of the post and comments in the dataset and then after used for designing a training model for Amharic aggression detection. The following tasks performed to prepare the dataset for annotation:

- Removing all non-Amharic texts
- Removing non-textual posts and comments.
- Removing numbers, all null, blank value, and whitespace.
- Normalizing Amharic texts
- Replacing Amharic abbreviation by its long forms
- Removing stop words and
- Removing duplication to ensure the uniqueness of each text in a dataset.

The details of the annotation procedure is in (Kumar et al., 2018). The dataset consists of posts collected from public page in Facebook and Twitter. The posts and comments are related to entities (organizations, individuals or issues) in Amharic text. After filtering and manually annotating, a benchmark corpus of 10,000 comments was created. Comments were assigned into three classes each representing a certain category of antisocial behavior as shown in Table 3.1. The data set size is imbalanced, given no such study to detect and to classify antisocial behavior into separate classes has been previously conducted. Three mark tagsets are generated in the dataset, namely: covertly aggressive, overtly aggressive and non-aggressive. Each post is annotated with one of these tagsets. The dataset consists of, 80% training, 10% validation, 10% test set. On the dataset set, the number of examples per class is unbalanced. The overtly aggressive class is approximately 40.01% of the training examples; covertly aggressive class is approximately 14.16% of the training examples and non-aggressive aggressive class is approximately 45.83% of the dataset.

3.2.3 Data Annotation

Annotation is a procedure for adding information to the collected data or document at some level. In this case, the annotation process needs to label a post or comment for building aggression datasets. The study uses a simple random sampling technique to select posts and comments to be annotated. The technique allows all the filtered posts and comments on each page to get an equal chance to be annotated. The annotation conducted based on the instruction guideline provided by the researcher. The labeling conducted by three annotators.

Table 3.2 aggression showing the three categories

Class	Label	Number of comments
Non-aggressive (NAG)	0	4583
Covertly aggressive (CAG)	1	1416
Overtly aggressive (OAG)	2	4001

To make the annotation process clear, the instruction or a guideline was prepared and listed for three classes in Appendix D. The instruction is based on the definition of aggressive comment based on Amharic language expert. Using this guideline, the Amharic language experts label each post and comment to three different class ‘Non-aggressive (NAG)’, ‘Covertly aggressive (CAG),’ ‘Overtly aggressive (OAG)’ text data.

The dataset annotation process done by the Amharic language expert. The number of annotators limited because of a lack of resources, mainly budget and time need for more annotators to participate in the process. The annotators were selected based on their willingness to perform the task and Amharic language skills. All the annotators are postgraduate in Amharic language. The annotators were instructed to annotate a random subset of an equal number of posts and comments from the dataset. The three annotators instructed to annotate the same number of equal instances of posts and comments to measure the inter-rater agreement using SPSS, which shows kappa agreement level (Viera & Garrett, 2005) their annotation decisions match. Due to the challenging nature of the manual annotation process of aggression datasets, the annotators were allowed to annotate in their own time freely. However, in order for the annotation task to be easier, the researcher gives brief insights into the annotation guidelines provided for labeling posts and comment into three different classes, as defined in Table 3.1.

We extracted several thousand comments from public pages in Social media using exportcomments.com for the raters to label. For the "Non-aggressive", "covertly aggressive" and "Overtly aggressive", the Kappa value for agreement is 0.91, thus showing that annotators can achieve a relatively high agreement for this task. Kappa agreement < 0 less than chance agreement, 0.01–0.20 Slight agreement, 0.21- 0.40 Fair agreement, 0.41-0.60 Moderate agreement, 0.61-0.80 Substantial agreement, 0.81-0.99, almost perfect agreement and chance agreement would equate to 0 (Viera & Garrett, 2005). In view of the fact that most of the comments were annotated by more than one annotator, we also measured the kappa inter-annotator agreement metric, which calculates the degree of agreement between the various annotators on a job. In this case, considering all comments that received annotations from the three annotators and obtain from SPSS to be used for this work.

Table 3.3 Example of aggression showing the three categories

Text	Category
ይህች ባቄላ ካደረች አትቆረጠምም እንዘፍዝፋት	CAG
የሁለት ዝሆኖች ፀብ ብዙ ሳር ያጠፋል!!! አንዱን በለው	CAG
ባዶ ጭንቅላት ማን አባረራቸው በእራሳቸው ፈርተው ወጡ እንጂ	OAG
ተቃጠል ቆሻሻው	CAG
ገለባን ማብነን ነው	CAG
ጭራቆች	OAG
ወይ ኢትዮጵያ የጅቦች ዋሻ መሆኗ ያሳዝናል	CAG
ማህበረ ቅዱሳን የትግራይ ደመኛ ጠላቶች የተደራጀ የፖለቲካ ድርጅት መሆኑ መገንዘብ ደግ ነው	OAG
ክብር ለብሄር ብሄረሰቦች ሀገር ኢትዮጵያ	NAG
አንተ አህያ ዝም ብለህ ተጫን 😊😊😊😊	CAG
እንከፍ	OAG
አንተ እባብ አሁን ደግሞ ማንን ልትነድፍ ነው	CAG
ውሻው ቢሄድ ጅቡ መጣ	CAG

3.2.4 Amharic Text Preprocessing

This subcomponent performs preprocessing of Amharic post and comment for the training and testing detection model. This preprocessing is performed based on the Amharic language and basic text preprocess techniques such as removing (cleaning) punctuations and special characters), normalization, removing stop words and tokenization automatically.

Stop words are common words, which are a portion of natural language that do not add meaning to text documents but make the text appear heavier. Since they do not add meaning, stop words can be easily removed without affecting the analysis process. Their removal reduces the number of features to be considered and as such can improve the performance of a classifier.

3.2.4.1 Tokenization

Splitting the input text yields the parts used to calculate features. The introduced general pipeline to extract NLP features uses words as atomic units and groups those into sentences. Thereby tokenization must divide the text into words, sentences and in some cases into larger utterances like paragraphs. Since the separation into paragraphs is not required for the short texts analyzed during this study it is not described in detail. Additionally, paragraphs depend heavily on the used convention, e.g. two-line feeds as separator. Those conventions lack of ambiguity and are consequently easy to separate.

3.2.4.2 Data cleaning

One crucial part of all practical machine learning tasks is data quality. Typical real-world data has a low quality due to missing measurements, faulty measurements or simple inaccurate measurements. These errors introduce unwanted noise. Therefore, data cleansing is required to improve the quality and boost the outcome. Textual data is no exception to this general rule. In addition to data specific cleaning steps textual data faces commonly a problem with dimensionality. Two properties of text contribute to this problem. On one side some words occur frequently in almost any text without adding information on the other side some words are so unique that they do not occur in other texts. Both aspects lead to a larger vocabulary and consequently to more dimensions in the initial BoW model. Therefore the curse of dimensionality is supplementary handled by removing and generalizing specific tokens.

3.2.4.3 Token removal

Removing irrelevant or redundant features is frequently used for most ML methods. At this point, tokens are the most basic features of text. Consequently, irrelevant tokens are removed before more complex features are calculated. The removal of tokens depends on the actual task and is thereby a domain specific task. The three token types listed below are commonly removed.

Stop words -not all words contribute the same amount of meaning, therefore words not contributing to the requested meaning can be ignored. In this thesis, we prepared stopwords lists manually from our corpus manually. Stop words are identified through assembled corpus and are entirely removed using python codes. Typical stop word corpora contain words like "ግን", "ነገር", "ነፃ", "ናቸው", and "ነበር".

Numbers -the meaning of a number depends on the context and format of it. This makes a number particular difficult to interpret. For instance, numbers might be nominal or numerical and different units can measure the same property as meter and centimeter. This required knowledge is difficult to acquire and numbers cannot be represented in the same way words are encoded, therefore numbers are predominantly ignored or replaced with one special token for all numbers. Some specific methods introduce additional hand-crafted rules for numbers.

Special Characters -apart from punctuation marks many texts use other special characters as well. Commas, parenthesis, bullet points, etc. are used to structure a text. Logically those structure elements do not carry any meaning if the structure is disregarded. Nevertheless, certain texts assign some meaning to certain sequences of special characters. The most prevalent case is text emoticons and non-Amharic characters.

Punctuation Marks Removal- The Amharic writing system uses the same punctuation marks, however, only a few of them are practically used, especially in computer-written text. The word-separator (HuletNetib), two square dots arranged like colon (:), and sentence-separator (aratnetib), four square dots arranged in a square pattern (::), in a writing system that are used consistently. This day, the use of HuletNetib is not seen in modern typesetting. In typesetting its place is almost completely taken over by space. Lists in text are separated by an equivalent of comma, netelaserez (፣) followed by ASCII space and deribsereze (፤), which is the equivalent of the semi-colon. The question mark (?) is used. Therefore in this study punctuation marks are removed to prepare collected data for the indexing process. This is because the same word that is attached to a punctuation mark and that is not attached to a punctuation mark is considered as different words.

3.2.4.4 Normalization of Amharic words

Amharic writing system contains characters or symbols, which have the same sound during reading. For example, in the case of Amharic the characters “ሀ, ሐ, ኀ, አ and ቀ” produces the same sound during reading with characters “ሃ, ሓ, ኃ, ኣ and ዓ” respectively. On the other hand, some alphabets share the same sound, such as, “ሀ, ሐ, and ኀ”, “ሰ and ሠ”, “ኣ and ዓ”, “ፒ and ጸ” and “ጸ and ፀ”. As a result, words which have the same meaning may have different spelling structure. For example, the Amharic words “ፀሐይ” can be written in a different way such as, “ፀሀይ” and “ጸሀይ”. In this thesis, such kinds of problems will be solved by changing the characters and the word to their common standard form.

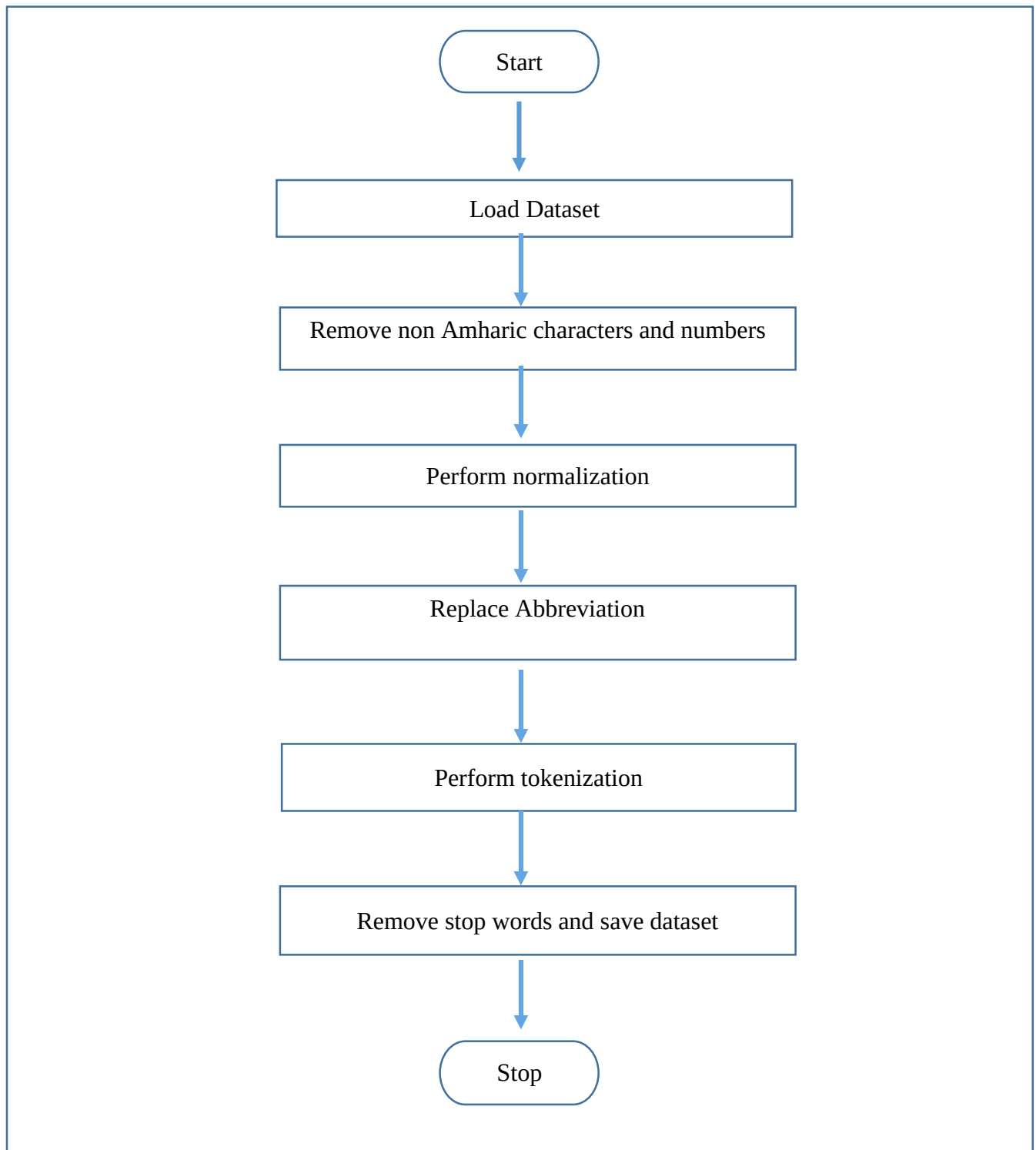


Figure 3.2 Preprocessing flow chart

Pseudo code: 3.1 Normalization

Input: text in a dataset

Output: clean text

Begin: 1. Read the text in the dataset;

2. While (! end of the text in a dataset):

If text contains [ሉ[ዋኦ]] [ሙ[ዋኦ]] [ቱ[ዋኦ]] [ሩ[ዋኦ]] [ሱ[ዋኦ]] [ጩ[ዋኦ]] [ፉ[ዋኦ]]; then

Replace characters with [ሊ] [ሚ] [ቲ] [ሪ] [ሲ] [ጢ] [ፊ] # respectively

If text contains [ሸ[ዋኦ]] [ቂ[ዋኦ]] [ቡ[ዋኦ]] [ቼ[ዋኦ]] [ሀ[ዋኦ]] [ኦ[ዋኦ]] [ጹ[ዋኦ]]; then

Replace characters with [ሺ] [ቋ] [ቢ] [ቺ] [ኦ] [ኧ] [ጺ]

If text contains [ኘ[ዋኦ]] [ኩ[ዋኦ]] [ከ[ዋኦ]] [ገ[ዋኦ]] [ደ[ዋኦ]] [ጡ[ዋኦ]] then

Replace characters with [ኸ] [ከ] [ከ] [ገ] [ደ] [ጡ]

If text contains [ሐ] [ሐ] [ሐ] [ሐ] [ሐ] [ሐ] [ሐ] [ሐ]; then

Replace characters with [ሀ] [ሀ] [ሀ] [ሀ] [ሀ] [ሀ] [ሀ] [ሀ]

If text contains [ኀ] [ኀ] [ኀ] [ኀ] [ኀ] [ኀ] [ኀ], then

Replace the characters with [ሀ] [ሀ] [ሀ] [ሀ] [ሀ] [ሀ] [ሀ]

If text contains [ሠ] [ሠ] [ሠ] [ሠ] [ሠ] [ሠ] [ሠ], then

Replace the characters with [ሰ] [ሰ] [ሰ] [ሰ] [ሰ] [ሰ] [ሰ]

If text contains [ዐ] [ዐ] [ዐ] [ዐ] [ዐ] [ዐ] [ዐ], then

Replace the characters with [አ] [አ] [አ] [አ] [አ] [አ] [አ]

If text contains [ጸ] [ጸ] [ጸ] [ጸ] [ጸ] [ጸ] [ጸ], then

Replace the character with [ፀ] [ፀ] [ፀ] [ፀ] [ፀ] [ፀ] [ፀ]

3. Return replaced text;

End:

Algorithm 3. 1 Normalization Amharic Character with the Same Sound (adopted from (Yonas & Tilahun, 2019))

Pseudo code: 3.3 Removing stop words

Input: text in a dataset

Output: clean text

Begin: 1. Read the text in the dataset stop word list;

2. While (! end of the text in a dataset):

 If a word is in the stop word list then

 Remove the word from the corpus

 End

3. Return clean_text;

End:

Algorithm 3.3 Stopwords removal adopted from((Emuye & Tesfa, 2020))

3.3 Input representation

The embedding layer is the representation of inputs to the deep neural network models. Each term in the vocabulary used in the model is encoded by the embedding layer. We experimented with our pre-trained word vectors for the embedding layer including word2vec and FastText. Thus, instead of using a pre-trained word vector file to look up the vector representation for each word, we used the FastText word embedding (Bojanowski et al., 2017) to infer the vector representation for each word in the vocabulary of our model. This allows us to use the sub-word level information to derive vector representation for words not in the vocabulary of the corpus on which the fastText model was pre-trained. This facility is especially important for social media posts which usually contains typos and abbreviations. In addition, we used a randomly initialized embedding which is trained with the network to learn task-specific word representation. So, the input representation to our deep learning model is a 400 dimensional word vectors derived from fastText model.

Recently, several representations have been proposed for representing texts in a way that is understandable by machine or deep learning algorithms. Taking an example is the one-hot encoding approach. In the one-hot encoding, all unique words are extracted and represented in a vector. Then each word is encoded into a numerical vector of a length equals to the number of unique words. Where the numerical vector is a vector of zeros except at the index of the respective word is set to one. This model is inefficient since the actual sense of words cannot be grasped, whereas words are interpreted primarily as vectors of zeros.

Word embedding is more robust mechanism than one-hot encoding approach (Verma & Khandelwal, 2019). In word embedding approach, all words of similar meaning have an analogous encoding. Embedding refers to a dense vector, where the length of the vector is a previously specified parameter. Also, parameters that are learned during the training phase are the components of the dense vector. A higher embedding dimension refers to a greater ability to learn the semantic meaning of sentences, but also to the need for larger training data (Verma & Khandelwal, 2019).

One of the models that is developed by the word embedding approach is the Word2Vec (Mikolov et al., 2013). Word2Vec is a neural network of three- layers, the input layer, the hidden layer, and the out- put layer. However, this architecture of neural networks is merely used to learn the weights of the hid- den layer that are the words embeddings. Mainly, Word2Vec has two structures of implementations; the Continuous Bag-of-Words (CBoW) and the Skip- Gram (SG) models. The CBoW model depends on several neighboring words to predict a certain middle word. Whereas, SG model is the opposite, that is used to predict the neighboring words for a single determined word. In which, the CBoW model takes a set of input words, where the number of words is determined by window size. While in the SG model, a single word is considered to predict the neighboring words. Predicting the neighboring words is accomplished by finding the highest probability words that are the most similar to the corresponding word.

The word embedding inside the deep learning framework is represented by an embedding layer. Each embedding layer is also known as a lookup table that has an input length and a dimension length. The input length is the size of the unique vocabulary of the respective dataset. While, the dimension length is a parameter specifies the length of the embedding. The dimension length should be fit to suite the problem and the size of the data

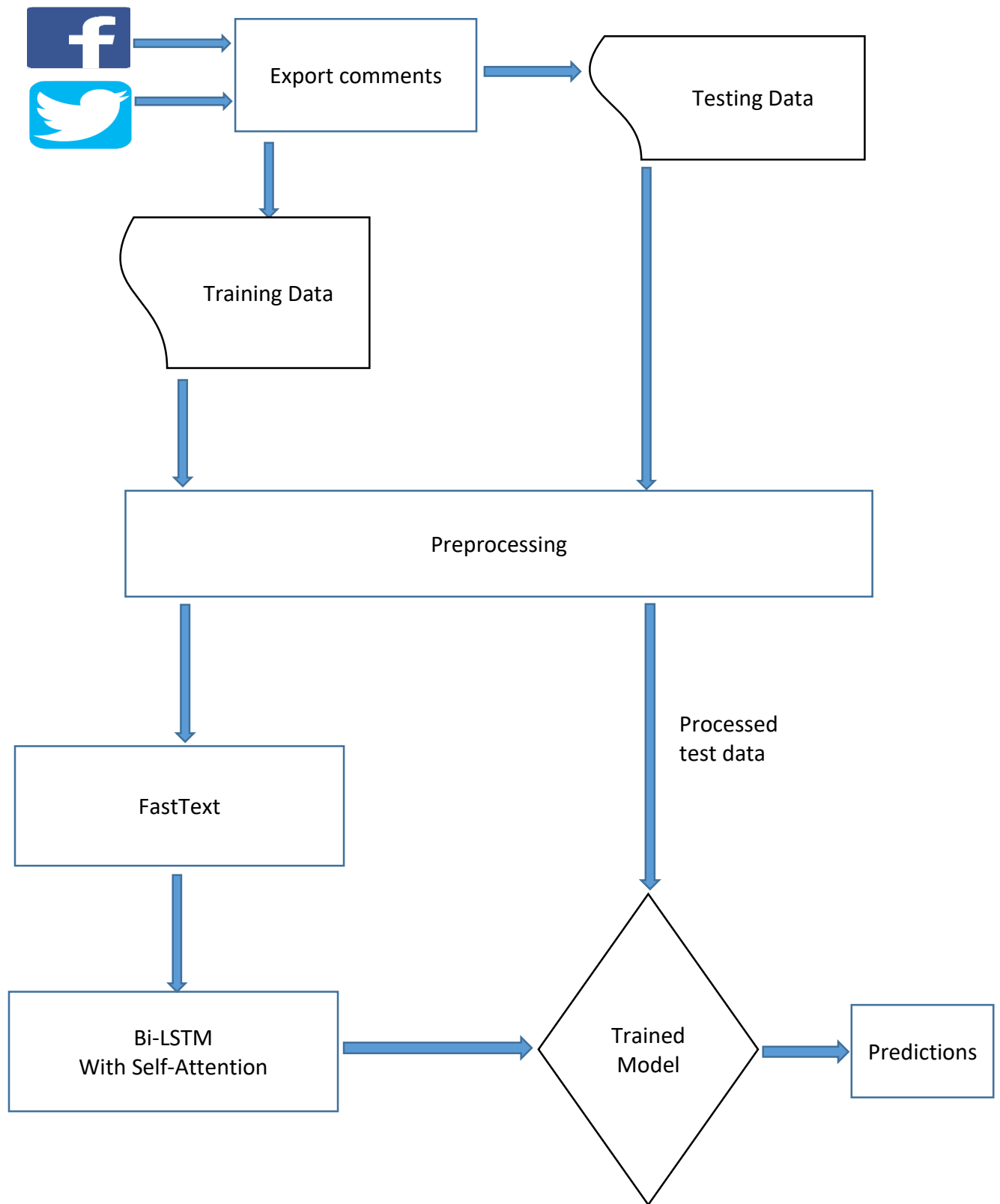


Figure 3.3 Amharic Aggression Detection Architecture

3.4 Imbalanced learn in deep learning

The learning phase and the subsequent prediction of deep learning algorithms can be affected by the problem of data imbalance. The balancing issue corresponds to the difference between the numbers of samples in the different classes. In this situation, the decision function of the different learning models could be highly impacted, with an important imbalanced ratio, favoring the class with the greater number of samples usually known as the majority class. There are two methods to address the imbalance problem of datasets, we can either choose an appropriate performance metric or resample the datasets. The metrics such as Confusion Matrix, Precision, Recall and F1 Score are designed to give a more truthful result when working with imbalanced classes.

There are two main methods of resampling that can use to even-up the classes: add copies of instances from the underrepresented class called over-sampling; delete instances from the over-represented class, called under-sampling. The random sampling approach is very easy to implement and fast to run. But the random under-sampling always leads to loss of information. And random over-sampling adopts a simple replicated sample strategy to increase the minority samples, it can easily lead to overfitting problem, that is, the features learned by the model are too specific and not generalized.

An improvement based on a random over-sampling algorithm is the SMOTE Synthetic Minority Oversampling Technique. By taking every minority class sample and introducing synthetic examples along the line segments joining any or all of the nearest neighbors of the k minority class, the minority class is over-sampled. Depending upon the amount of oversampling required, neighbors from the k nearest neighbors are randomly chosen (Kovács et al., 2020).

SMOTE based algorithms implicitly consider SMOTE for Learning from Imbalanced Data: 15-year Anniversary a mechanism to counteract both the class imbalance and the small disjoints. By means of creating new instances in between close examples, it allows to reinforce the representation within the clusters. The premise for the good behavior of SMOTE is related to the fact that the nearest examples should be selected within that very area. SMOTE (Kovács et al., 2020) algorithm achieves the increment of low-frequency samples by performing random and linear interpolation between low-frequency samples and their congeneric nearest neighbors. The advantage of the SMOTE algorithm is that it generates new samples during the oversampling process, instead of monotonously replicating the original samples, which avoids the increase of erroneous samples and facilitating the classification process of the classifiers.

Since an aggressive comment is a real, but limited phenomenon is frequent facing with datasets where non-aggressive samples are predominant. Therefore, we exploit the oversampling techniques.

Problem with Imbalanced dataset:

- Algorithms may get biased towards the majority class and thus tend to predict output as the majority class.
- Minority class observations look like noise to the model and are ignored by the model.
- Imbalanced dataset gives misleading accuracy score.

We use two methods to address the imbalance problem of datasets, we choose different performance metric and resample the datasets with SMOT resample algorithm. The metrics such as Confusion Matrix, Precision, Recall and F1 Score are designed to give a more truthful result when working with imbalanced classes. There are two main methods of resampling that can use to even-up the classes: add copies of instances from the underrepresented class called over-sampling; delete instances from the over-represented class, called under-sampling. The random sampling approach is very easy to implement and fast to run. But the random under-sampling always leads to loss of information. And random over-sampling adopts a simple replicated sample strategy to increase the minority samples, it can easily lead to overfitting problem, that is, the features learned by the model are too specific and not generalized.

CHAPTER FOUR

EXPERIMENT AND RESULT DISCUSSION

4.1 Introduction

This thesis aim to develop aggression detection for the Amharic language. The output of this study can be a good opportunity for other studies to develop NLP tools for the Amharic language that classify aggression into three labels, non-aggressive, covertly and overtly aggressive comments. In this chapter, the working or developing environment of the proposed system, experimental details and evaluation results along with a discussion of results are presented.

4.2 Experimental Environment Setup

This study used several development tools and packages to implement the proposed solution, Amharic aggression detection. This study uses python programming language for implementing and experimenting with each proposed solution from the data preprocessing to the model building phase. Also, to evaluate the implemented proposed classifiers model. Python used because of its programming language of choice for developers, researchers and data scientists who need to work in machine learning models. Table 4.1 shows the list of tools and python packages with their version and description which is used in this study.

Table 4.1 Description of the Tools and Python Package Used During the Implementation

Tools		
Tools	Version	Description
Jupyter notebooks	3.7.6	A web-based integrated development environment for interactive computing.
TensorFlow	1.13.2	TensorFlow is widely used for machine learning, deep learning and numerical computations. In this study it was used as the backend for training the LSTM on CPUs
Keras	2.3.1	Keras greatly simplifies deep learning tasks in Python. It runs on top of TensorFlow and has built-in and easy to work with implementations of various neural networks models. In this study Keras was used to create, train, store, load and do all other necessary operations with LSTM.
Anaconda Navigator	3	Allows us to launch development applications and easily manage conda packages, environments, and channels without the need to use command-line commands.
Pandas	1.0.1	Pandas is a library that greatly simplifies datasets processing, display and analysis.
NumPy	1.18.4	NumPy is a fundamental Python library for scientific computing, making it simpler to work with arrays, matrices, use high-level math functions etc. This library was heavily used in this work for operations on data structures and various computations.
Gensim	3.8.3	It is a free python library designed to extract semantic topics from documents by extracting statistical co-occurrence patterns within a corpus of training documents in an unsupervised way automatically.
matplotlib	-	Matplotlib is a plotting library. It provides a vast number of built-in plots for various data graphical representation purposes. In this study it was used to visualize dataset sequences and LSTM train history.
python	3.7.6	Easy to learn, powerful programming language to develop a machine learning application.
Excel	2013	Used data preparation tasks in cleaning, filtering and sorting the gather data. Also, used to manage the annotation task.
t-distributed stochastic neighborhood embedding(t-SNE)	1.4.0	An algorithm for visualization word embedding based on Stochastic Neighbor Embedding
imblearn	0.7.0	Imbalanced-learn Python library, to use oversampling approach or undersampling approach to rebalance the original training set.
IBM SPSS	2020	It is a software that offers advanced statistical analysis and kappa agreements.

4.3 Implementation of FastText

To implement FastText, this thesis uses a python Genism module which is used to implement different embedding methods. We train two word embedding, word2vec and FastText. First import and instantiate word2Vec class and FastText with necessary parameter and builds the vocabulary, and training the Word2Vec model using the posts and comments that are gathered from the select pages. We use 221,568 Amharic comments to train FastText and word2vec. FastText predict vector for out of vocabulary words from its character n-grams even if word doesn't appear in training corpus. FastText also learn vectors for sub-parts of words called character n-grams ensuring that e.g., the words “መተባበረ”, “ለመተባበረ”, “ቢተባበረ ”, “ተባበረ” and “ስንተባበረ” all fall into same dimension in vector space, even if they tend to appear in different contexts. This feature enhances learning on heavily inflected languages.

With this manifest it benefits to generate better word embeddings for rare words. Moreover, FastText predict vector for out of vocabulary words from its character n-grams even if word doesn't appear in training corpus. In contrast Word2vec leaves unseen words as out-of-vocabulary words. So, with this intuition, we proposed to use FastText as word vector generating model for this work. As it is extension to Word2Vec model, FastText also has two architectures for computing word representations called Skip-gram and CBOW (continuous-bag-of-words). The Skip-gram model learns to predict a target word given a nearby word. On the other hand, the CBOW model predicts the target word according to its context. For instance, given the sentence “በርቱ በየትኛውም ቦታ ያለን ሰዎች የሚከፈለውን መስዋእትነት ከፍለን ኢትዮጵያችንን ከፍ እናደርጋለን” and the target word “መስዋእትነት”. The Skip-gram model predicts the target using a random close-by word, like “ከፍለን” or “እናደርጋለን”. Whereas the CBOW model takes all the words in a surrounding window, like {ሰዎች: የሚከፈለውን, ከፍለን: ኢትዮጵያችንን, ከፍ: እናደርጋለን}, and uses the sum of their vectors to predict the target word “መስዋእትነት”.

```
In [9]: import gensim
        from gensim import models
        from gensim.models import FastText

In [10]: model = FastText(review_lines, size=400, window=5, min_count=2, workers=4, sg=1, iter=10)
```

Figure 4.1 Sample Code for Building FastText word embedding

At the time of training, FastText trains by sliding a window over the input text and either learning the target word from the remaining context (CBOW), or all the context words from the target word (“Skip-gram”), and learning can be viewed as a series of updates to a neural network with two layers of weights and three tiers of neurons, in which the outer layer has one neuron for each word in the vocabulary and the hidden layer has as many neurons as there are dimensions in the embedding space. In this way, it is similar to Word2Vec. Unlike word2vec, FastText also learn vectors for sub-parts of words called character n-grams ensuring that e.g., the words “መተባበረ”, “ለመተባበረ”, “ቢተባበረ”, “ተባበረ” and “ስንተባበረ” all fall into same dimension in vector space, even if they tend to appear in different contexts. This feature enhances learning on heavily inflected languages. Despite use of sub-word information, training FastText is same as Word2Vec for both CBOW and Skip-gram models. The key difference between FastText and Word2Vec is the use of n-grams. Word2Vec learns vectors only for complete words found in the training corpus. FastText, on the other hand, learns vectors for the n-grams that are found within each word, as well as each complete word. At each training step in FastText, the mean of the target word vector and its component n-gram vectors are used for training. The adjustment that is calculated from the error is then used uniformly to update each of the vectors that were combined to form the target. This adds a lot of additional computation to the training step. At each point, a word needs to sum and average its n-gram component parts. The trade-off is a set of word-vectors that contain embedded sub-word information. These vectors have been shown to be more accurate than Word2Vec vectors by a number of different measures. Treating character n-gram manifests FastText the following advantage over Word2Vec:

Generate better word embeddings for rare words (even if words are rare their character n grams are still shared with other words - hence the embeddings can still be good). This is simply because, in word2vec a rare word (e.g., 10 occurrences) has fewer neighbors to be pulled by, in comparison to a word that occurs 100 times whereas FastText has more neighbor context words and hence is pulled more often resulting in better word vectors.

Out of vocabulary words - they can construct the vector for a word from its character n grams even if the word doesn't appear in the training corpus. Both Word2vec and Glove can't.

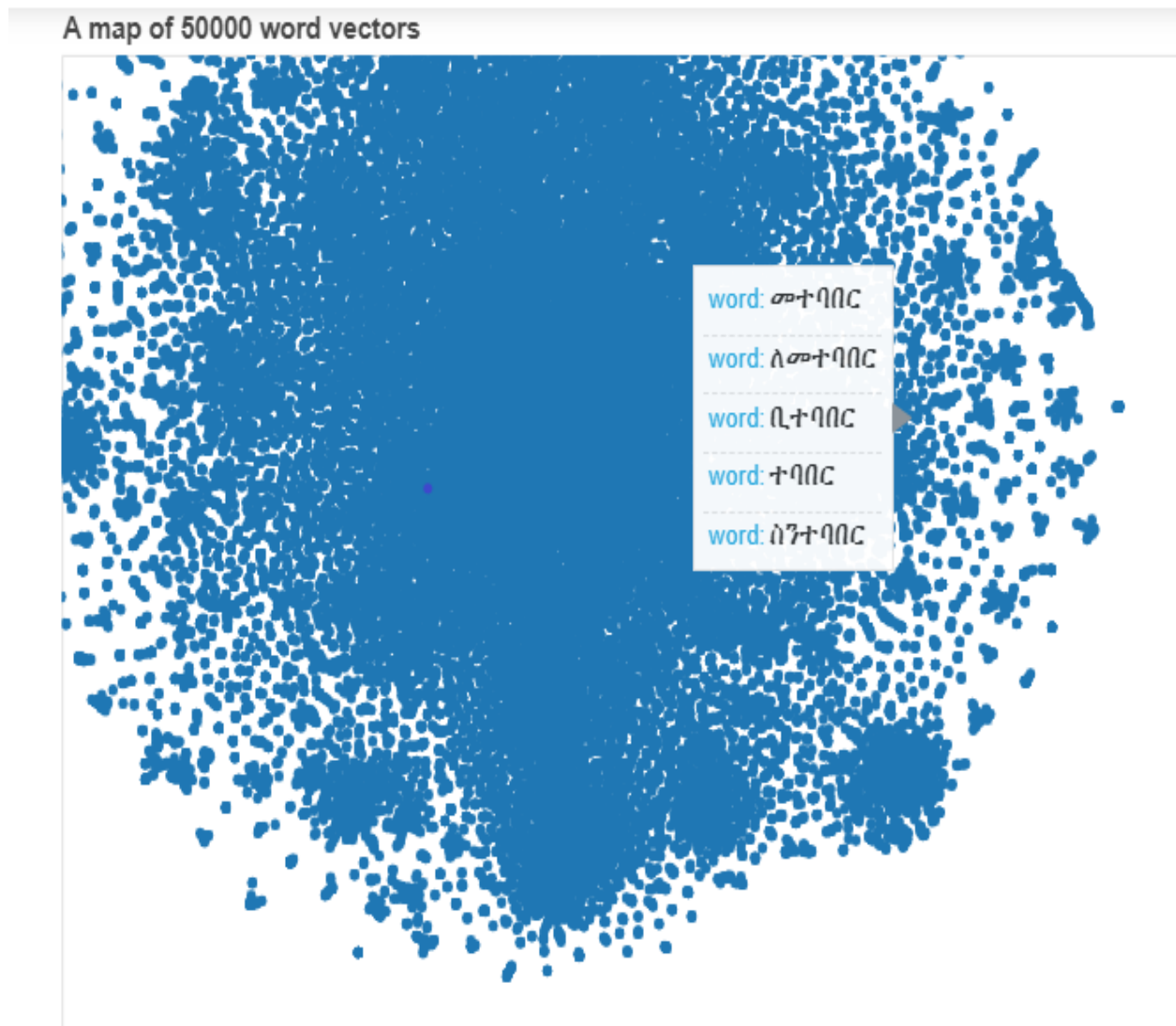


Figure 4.2 Visualization of 50,000 words from the fastText vector produced by t-SNE

In the above Figure, Figure 4.2 “መተባበር”, “ለመተባበር”, “ቢተባበር”, “ተባበር” and “ስንተባበር” exist in the same space vectors. This clearly shows FastText word embedding is importance for morphologically rich language like Amharic.

The notion of word similarity in vector space is very useful in larger semantic tasks. Knowing how similar two words can help in computing how similar the meaning of two phrases or sentences are, a very important component of natural language understanding tasks like hate speech detection, and aggression detection. FastText put similar words in one vector space which is important for the identification of aggressive comments contextually with spelling errors also.

```
In [19]: model.wv.most_similar("ለማጥላት", topn=15)

Out[19]: [('ለማጥላትና', 0.9265490174293518),
          ('ማጥላት', 0.8752273917198181),
          ('ለማጋጨት', 0.8588778376579285),
          ('ለማጋደል', 0.8491545915603638),
          ('ለማባላት', 0.8464323878288269),
          ('ለማጋጫት', 0.8460648059844971),
          ('ለማጫረስ', 0.8411413431167603),
          ('ለማናከስ', 0.839849591255188),
          ('ለማጫረስ', 0.8330080509185791),
          ('ለማጋጨትና', 0.8294110298156738),
          ('ለማጋጭት', 0.8229188323020935),
          ('ለማበሳጨት', 0.8221513032913208),
          ('ለማፋጀት', 0.8113781809806824),
          ('ለማበጣበጥ', 0.8108035922050476),
          ('ለማጋ', 0.8107782602310181)]
```

Figure 4. 3 Top 10 similar words in FastText

In the above Figure, Figure 4.3 “ለማጋጨት”, “ለማጋጨትና”, and “ለማጋጨት” refers the same word “ለማጋጨት” with typing error. This clearly shows FastText word embedding can handle typing error, and words which has similar meanings also considered easily with typos problem.

We use the following hyperparameters for FastText:

Table 4.2 Parameters of FastText

Hyperparameters	Value
Size	400
Window	5
Min_count	2
workers	4
sg	1
Iteration	10

4.4 Implementation of Deep learning model

The implementation of proposed models are constructed based on using Python programming language working on Jupyter Notebook. Python has a large number of scientific libraries for data processing and deep learning approaches. Some libraries are presented in the following sections and their relevance to this thesis are described.

4.4.1 Library

4.4.1.1 TensorFlow

TensorFlow is an open source software library for numerical calculations using data flow graphs. Nodes represent math operations in the figure, and edges in the figure represent multidimensional data arrays, i.e., tensors, that relate to each other. Its flexible architecture allows you to expand computing on a variety of platforms, such as one or more CPUs (or GPUs), servers, mobile devices, and more in desktop computers. TensorFlow was originally developed by researchers and engineers from the Google Brain Group (part of the Google Institute for Machine Intelligence) for machine learning and deep neural networks, but its versatility makes it widely used in other computations field.

4.4.1.2 Keras

Keras (Manaswi, 2018) is a high-level neural network API, Keras is written in pure Python and based on TensorFlow, Theano and CNTK backend. Keras was born to support rapid experimentation and quickly convert your ideas to results. Keras is highly modular, minimalist, and scalable, so it supports simple and fast prototyping and model CNN and RNN, or a combination of the two models, switch CPU and GPU seamless.

Keras's design principle is user-friendly. Keras is an API designed for humans. The user experience is always the primary and central part of the consideration. Keras provides a consistent and concise API that drastically reduces the workload for users in general use, while Keras provides clear and practical bug feedback. A model can be understood as a sequence of layers or as a graph of data. Fully configurable modules can be freely combined at the least cost. Specifically, the network layer, loss functions, optimizers, initialization strategies, activation functions, and regularization methods are all separate modules.

4.4.1.3 Deep learning Layer

In our evaluation of the proposed scheme, each classifier is implemented as a deep learning model having five layers and is described as follows:

- The Input (Embedding) Layer. The input layer's size is defined by the number of inputs for that classifier. This number equals the size to the word vector plus the number of additional features. The word vector dimension was set to 400 so that to be able to encode every word in the vocabulary used.
- The Bidirectional LSTM layer. BiLSTM extract word information from both directions. This layer is fully connected to both the Input and the subsequent layer.
- The Attention Layer. We add an attention layer for finding the contribution of each word by assign a weight to each word feature to the whole sequence.
- The dense layer. The output of the BiLSTM with attention was run through an additional layer to improve the learning and obtain more stable output. The ReLU activation function was used. Its size was selected equal to the size of the input layer.
- The output layer. This layer has 3 neurons to provide output in the form of probabilities for each of the three classes neutral, covertly, and overtly aggressive. The softmax activation function was used for this layer.

Hyperparameters used:

Table 4. 3 BiLSTM parameters

Hyperparameters	Value
Word embedding size	400
Max sentence length	100
Dropout after word embedding	0.6
Num of BiLSTM cells	256
optimizer	Adam
Batch-size	50

4.5 Train-test split

The final preprocessing step splits the data into a training, validation, and test set. The deep learning modules use the training set to update the weights of the model. Subsequently, the validation set is utilized to assess models during the optimization processes. Finally, the models predictions are evaluated on the test set. The aggression corpus consists of 4583, 1416 and 4001 for neutral, covertly and overtly aggressive comment respectively with a total number of 10,000 sentences. Therefore, this thesis have approximately 80% of the sentences in the training set, 10% in the validation set, and 10% in the test set. A summary of the number of sentences, the number of cases and the number of annotations in the dataset are shown in Table 3.1.

4.6 Evaluation Measures

4.6.1 Accuracy

Accuracy, being the most intuitive performance measure, provides information regarding how well the algorithm can correctly classify inputs. Following this knowledge, accuracy will only be used as a reference to overall performance instead of the main performance measure. Accuracy is the most popular performance measure used and for good reason. It's extremely helpful, simple to compute and to understand. It is the proportion of the correctly classified samples and all the samples.

Accuracy: It is simply a ratio of correctly predicted observation to the total observations. Accuracy is a great measure but only while having symmetric datasets where values of false positive and false negatives are almost similar (Devishree & Marseline, 2018).

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

4.6.2 Precision

The next performance measure to be reviewed is precision, which gives information regarding the algorithm's performance in avoiding false positives. This metric tells us how many elements were correctly identified as belonging to a particular class out of the total number of elements that the classifier claim belong to that class. This means that precision shows how many entries are correctly labelled as aggressive, meaning the value will get lower if the algorithm is set to be too sensitive to aggression detection (Devishree & Marseline, 2018).

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

4.6.3 Recall

The next performance measure is used to evaluate how well the algorithm can classify input with aggression or how many of the actual aggression can the algorithm detect. It is expected that higher recall corresponds to a stricter rule of detecting aggression, meaning that possible neutralizer might actually lower recall score.

Recall: Recall is the ratio of correctly predicted positive observations to the all observations in actual class, which identify all relevant instances (Devishree & Marseline, 2018).

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

4.6.4 F1 Measure

In some cases, precision might get really low when the main concern is maximizing recall, therefore a more balanced approach is needed. The F1 measure gives a more balanced approach towards recall and precision. By assessing the F1 measure, the overly conscious approach can be avoided because maximizing the algorithm's ability to correctly identify aggression is the main concern of this research. Correctly identifying aggression means that the algorithm is expected to be able to balance between finding aggression and non-aggression instead of blindly labelling inputs as aggression without checking for a possible neutralizer.

The F1 score metric, single metric that combines recall and precision with the weighted average. Therefore, this score takes both false positive and false negatives (Devishree & Marseline, 2018).

$$F1 = 2 \times \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

4.6.5 Confusion Matrix

The evaluation of text classification techniques can be obtained in terms of correctness by computing statistical measures namely the True Positives (TP), True Negatives (TN), False Positive (FP) and False Negatives (FN). These components form the Confusion Matrix as shown in Fig. 1. The dataset was split into test and train type datasets. Python was used as the environment in which this combined data had to run. Using Python, the required packages had to be called. The dataset was opened on the Python on Jupyter notebook platform and, by setting various parameters related to, the code was run on the two deep learning algorithms on the dataset. The results in confusion matrix, Precision, F1-score, accuracy and we used matplotlib to draw the results.

The Confusion Matrix for a 3-class aggression can be seen in Table 2.2, which displays the result for each class for a text classifier. We can use the results to calculate some other evaluation metrics (Precision, Recall and F1 Score) for each class, and also to calculate the overall versions of these metrics.

4.7 Experimental Results

Many experiments were implemented to confirm the effectiveness of the proposed model. Two different experimental approaches are presented to evaluate the performance of the proposed model. The first approach is to compare the performance of the word2vec and FastText word embedding models in deep learning created through the classification success of the proposed model. In the second approach, the performance of the proposed model is compared with other deep learning known algorithms with combination, BiLSTM-CNN on this data set.

Many libraries and tools are available for developing deep learning models. Keras is one of the most preferred software frameworks. TensorFlow is used in the backend of Keras, which provides support for the CPU or GPU environment.

Table 4.4: Evaluation results with precision, recall and F1-score evaluation metrics

Model	Aggression	Precision	Recall	F1-score	#Samples
BiLSTM + FastText	NAG	0.73	0.84	0.79	481
	CAG	0.59	0.25	0.35	144
	OAG	0.69	0.71	0.70	375
	Avg/Total	0.71	0.71	0.71	1000
BiLSTM + Word2vec	NAG	0.75	0.80	0.78	481
	CAG	0.41	0.25	0.31	144
	OAG	0.69	0.73	0.71	375
	Avg/Total	0.70	0.70	0.70	1000
BiLSTM- CNN + FastText	NAG	0.79	0.77	0.78	481
	CAG	0.37	0.45	0.41	144
	OAG	0.72	0.68	0.70	375
	Avg/Total	0.69	0.69	0.69	1000
BiLSTM- CNN + Word2vec	NAG	0.73	0.81	0.77	481
	CAG	0.46	0.38	0.42	144
	OAG	0.72	0.66	0.69	375
	Avg/Total	0.70	0.70	0.70	1000

In Table 4.4, Precision, Recall, F1-score and Accuracy was analyzed for BiLSTM and BiLSTM-CNN with FastText and Word2Vec word embeddings, and BiLSTM achieved highest classification accuracy, precision, recall and F1-score.

```
In [32]: print(classification_report(test_labels, pred2))
print(confusion_matrix(test_labels.argmax(axis=1), pred2.argmax(axis=1)))
```

```

              precision    recall  f1-score   support

     0       0.73       0.84       0.79         481
     1       0.59       0.25       0.35         144
     2       0.69       0.71       0.70         375

 micro avg       0.71       0.71       0.71        1000
 macro avg       0.67       0.60       0.61        1000
weighted avg       0.70       0.71       0.69        1000
samples avg       0.71       0.71       0.71        1000

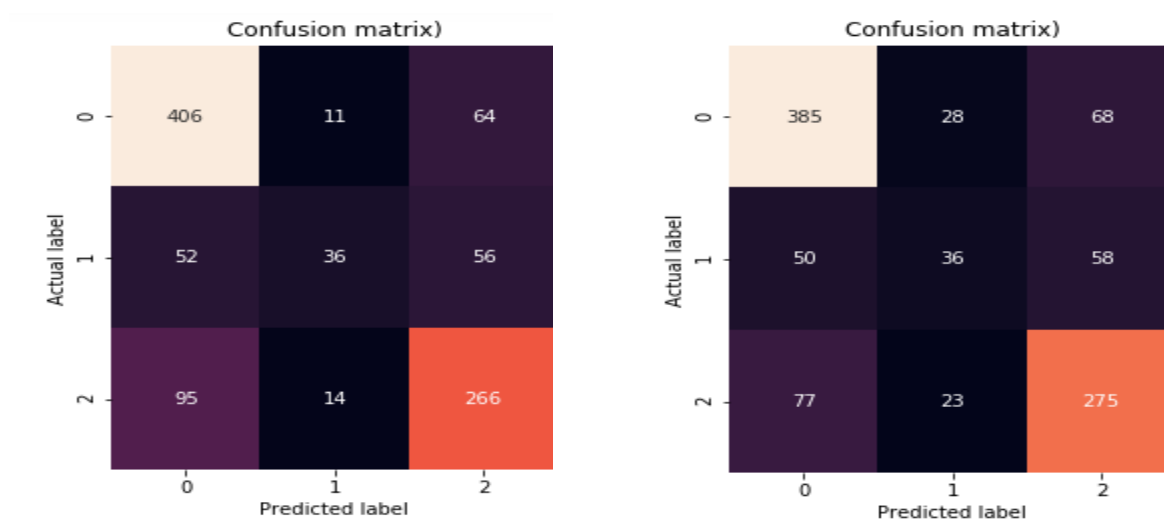
```

Figure 4.4 Sample code and result to evaluate the Model

Model	Word embedding	Accuracy
BiLSTM	Word2Vec	75.08%
BiLSTM	FastText	76.31%
BiGRU	Word2Vec	74.77%
BiGRU	FastText	75.1%

Table 4.5. Results with accuracy metrics evaluation

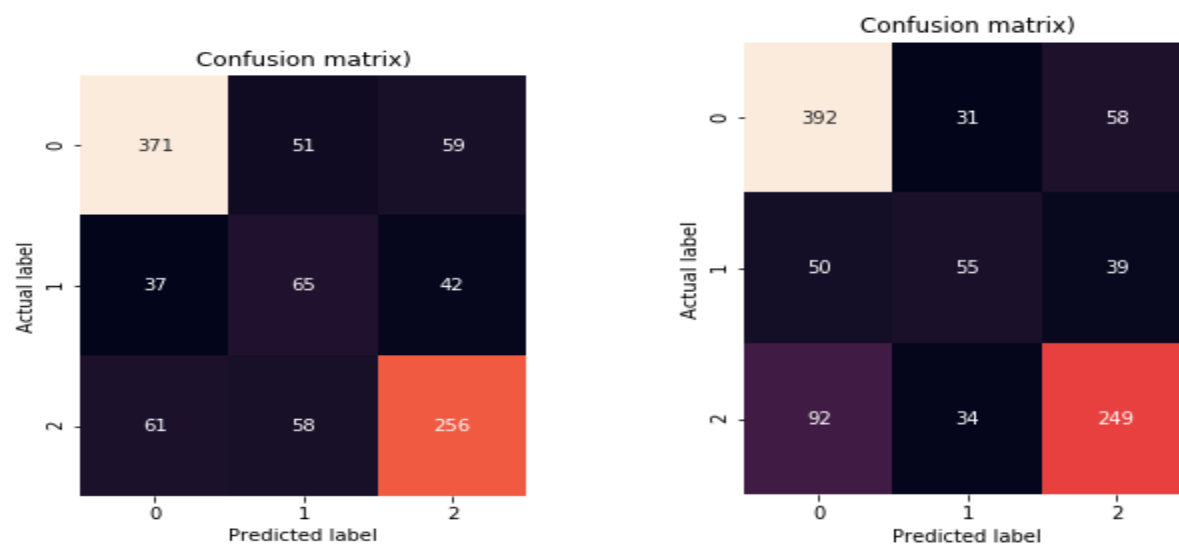
Confusion matrix for BiLSTM with FastText and word2vec word embedding:



A) BiLSTM with FastText word embedding B). BiLSTM with word2vec word embedding

Figure 4.6 Confusion matrix of BiLSTM with FastText and word2vec word embedding

Confusion matrix for BiLSTM-CNN with FastText and word2vec word embedding:



B) BiLSTM-CNN with FastText word embedding B). BiLSTM-CNN with word2vec word embedding

Figure 4.7 Confusion matrix of BiLSTM-CNN with FastText and word2vec word embedding

We evaluated classification model by using different metric, achieving an average result as presented in table 4.4, table 4.5 and Figure 4.6 and Figure 4.7 for confusion matrix result. It was evident that the BiLSTM with FastText outperform to classify non-aggressive, covertly and overtly aggressive with 0.7631, 0.71, 0.71 and 0.71 accuracy, F1-score, Precision and Recall respectively with Facebook and Twitter Amharic language aggressive posts and comments. The BiLSTM-CNN better result for FastText model with 0.751, 0.70, 0.70 and 0.70 accuracy, F1-score, Precision and Recall respectively.

The evaluation results can be observed that precision, recall, and F1-score of Non-aggressive class are higher than other two classes. Similarly, for the Covertly Aggressive class all these metrics are lower than other classes. This is because, CAG is hidden aggression in CAG class posts and it is very difficult to identify that type of posts. For CAG class our proposed method is not doing well because the dataset is imbalanced, it is very difficult to balance using synthetic data in oversampling method and aggression is hidden in CAG class comments.

4.8 Error Analysis of the Experiment

There is the problem of misclassified posts and comments in our model, since most covertly violent texts (Neutral & overtly aggressive) are both misclassified. This occurs due to the following reasons. The complex nature of covertly aggressive and lack of a clean rule to separate this content and imbalance data. The other reasons are slang words which has different meanings other than the common one and sentences without spaces which makes the models to fail to incorporate essential features, and also the occurrence of the same word for each class text could cause incorrect features to be used by the models to classify the comments incorrectly. Also, their optimized value could not be the network parameters we used in our experiments. The above-listed issues influence the efficiency of the established model and misclassification takes place.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

In this study, we designed and developed the Amharic aggression detection model. It uses a prediction model trained on a network model build using deep learning network architectures, BiLSTM with fastText word embedding.

5.1 Conclusions

Nowadays, there is a huge amount of data exchanged between users every day in social media and has a lot of positive or negative effects on user life, among those huge amounts of data aggressive texts on Facebook and Twitter play a major role in negatively affecting user life by imitating them for conflict based on their ethnicity, faith, ethnic group or any other attributes that could distinguish indigenous life which could affect individuals or groups with open or hidden expression.

A model is constructed in this study that can detect aggressive content using BiLSTM, BiGRU and BiLSTM-CNN. We begin to define the aggressive text to effectively enforce the model, which is the text consisting of both covertly and overtly content based on their description, and we examine current aggression detection techniques. In this work, we have used our trained FastText word embedding model to generate word vectors that can able to capture syntactic and semantic relations of words and randomly generated vectors using the embedding layer. The FastText word embedding take into account the internal structure of words while learning word representations; this is especially useful for morphologically rich languages like Amharic. The semantic and syntactic information of words that is contained in these vectors make them powerful features for NLP tasks like aggression detection. We experimented using state of the art deep learning algorithms such as BiLSTM, BiGRU and BiLSTM-CNN with FastText and Word2Vec embedding. BiLSTM achieved better performance with an accuracy of 76.31% with FastText embedding. This research showed positive results, but this finding is enhanced by more detailed future work.

5.2 Contribution of the study

The contribution of this study are listed as follows.

- ❖ Develop a labeled aggression dataset built from two public social media.
- ❖ Prepared our pre-trained FastText and word2vec word embedding, which is important for future works in this area.
- ❖ We design a model for Amharic text aggression and also, we compare deep learning algorithms.

5.3 Recommendation

In order to design and improve Amharic aggression detection, we did a lot of work. We propose the following as a way forward based on the concluding and finding remarks of the report.

- ❖ We have used a manually annotated dataset with three persons annotate the dataset with more people to improve the dataset quality to avoid the biased problem.
- ❖ For future researchers will contain Amharic informal text, like slang word.
- ❖ Building a model which includes other forms of aggression content on social media such as image, audio or video.

REFERENCES

- Abdullah-Al-Mamun, & Akhter, S. (2019). Social media bullying detection using machine learning on Bangla text. *ICECE 2018 - 10th International Conference on Electrical and Computer Engineering*, 385–388. <https://doi.org/10.1109/ICECE.2018.8636797>
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5, 135–146. https://doi.org/10.1162/tacl_a_00051
- Chen, J., Yan, S., & Wong, K. C. (2017). Aggressivity detection on social network comments. *ACM International Conference Proceeding Series, Part F1278*, 103–107. <https://doi.org/10.1145/3059336.3059348>
- Devishree, T., & Marseline, K. S. J. (2018). Comparative Study on Classification Algorithms for Sentiment Analysis. In *Nternational Journal of Engineering Development and Research* (Vol. 6, Issue 3).
- Elnagar, A., Al-Debsi, R., & Einea, O. (2020). Arabic text classification using deep learning models. *Information Processing and Management*, 57(1), 102121. <https://doi.org/10.1016/j.ipm.2019.102121>
- Emuye, B., & Tesfa, T. (2020). AMHARIC TEXT HATE SPEECH DETECTION IN SOCIAL MEDIA USING. *AMHARIC TEXT HATE SPEECH DETECTION IN SOCIAL MEDIA USING DEEP LEARNING APPROACH BAWOKE.*
- Enrique Muñiz Cuza, C., Liz De la Peña Sarracén, G., & Rosso, P. (2018). Attention mechanism for aggressive detection. *CEUR Workshop Proceedings*, 2150, 114–118.
- Fauzi, M. A., & Yuniarti, A. (2018). Ensemble method for indonesian twitter hate speech detection. *Indonesian Journal of Electrical Engineering and Computer Science*, 11(1), 294–299. <https://doi.org/10.11591/ijeecs.v11.i1.pp294-299>
- Galery, T., Charitos, E., & Tian, Y. (2018). Aggression Identification and Multi Lingual Word Embeddings. *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying ({TRAC}-2018)*, 74–79. <https://www.aclweb.org/anthology/W18-4409>
- Gordeev, D. (2016). Detecting state of aggression in sentences using cnn. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9811 LNCS(May), 240–245. https://doi.org/10.1007/978-3-319-43958-7_28
- Gutiérrez-Esparza, G. O., Vallejo-Allende, M., & Hernández-Torruco, J. (2019). Classification of cyber-aggression cases applying machine learning. *Applied Sciences (Switzerland)*, 9(9). <https://doi.org/10.3390/app9091828>
- Haddad, B., Orabe, Z., Al-abood, A., & Ghneim, N. (2020). *Arabic Offensive Language Detection with Attention-based Deep Neural Networks* (Issue May).
- Jang, B., Kim, M., Harerimana, G., Kang, S., & Kim, J. W. (2020). *applied sciences Bi-LSTM Model to Increase Accuracy in Text Classification : Combining Word2vec CNN and*

Attention Mechanism.

- Julian, R., & Ralf, K. (2018). Aggression Identification Using Deep Learning and Data Augmentation. *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, 25(Aggression Identification Using Deep Learning and Data Augmentation), 103–109. <https://www.aclweb.org/anthology/W18-4418.pdf>
- Kenenisa, Y., & Melak, T. (2019). Adama , Ethiopia, September 2019. *Hate Speech Detection for Amharic Language on Social Media Using Machine Learning Techniques By:*, September.
- Kovács, B., Tinya, F., Németh, C., & Ódor, P. (2020). Unfolding the effects of different forestry treatments on microclimate in oak forests: results of a 4-yr experiment. *Ecological Applications*, 30(2), 321–357. <https://doi.org/10.1002/eap.2043>
- Kulshrestha, A., Krishnaswamy, V., & Sharma, M. (2020). Annals of Tourism Research Bayesian BILSTM approach for tourism demand forecasting. *Annals of Tourism Research*, 83(April), 102925. <https://doi.org/10.1016/j.annals.2020.102925>
- Kumar, R., Ojha, A. K., Malmasi, S., & Zampieri, M. (2018). Benchmarking Aggression Identification in Social Media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying* (Issue 1).
- Manaswi, N. K. (2018). Deep Learning with Applications Using Python. In *Deep Learning with Applications Using Python*. <https://doi.org/10.1007/978-1-4842-3516-4>
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, 1–12.
- Mouheb, D., Ismail, R., Al Qaraghuli, S., Al Aghbari, Z., & Kamel, I. (2019). Detection of Offensive Messages in Arabic Social Media Communications. *Proceedings of the 2018 13th International Conference on Innovations in Information Technology, IIT 2018*, 24–29. <https://doi.org/10.1109/INNOVATIONS.2018.8606030>
- Nikhil, N., Pahwa, R., Nirala, M. K., & Khilnani, R. (2018). LSTMs with attention for aggression detection. *ArXiv*.
- Sahay, K., Khaira, H. S., Kukreja, P., & Shukla, N. (2018). Detecting Cyberbullying and Aggression in Social Commentary using NLP and Machine Learning. *International Journal of Engineering Technology Science and Research*, 5(1), 1428–1435. www.ijetsr.com
- Salawu, S., He, Y., & Lumsden, J. (2020). Approaches to Automated Detection of Cyberbullying: A Survey. *IEEE Transactions on Affective Computing*, 11(1), 3–24. <https://doi.org/10.1109/TAFFC.2017.2761757>
- Sazany, E., & Budi, I. (2018). Deep Learning-Based Implementation of Hate Speech Identification on Texts in Indonesian: Preliminary Study. *Proceedings of ICAITI 2018 - 1st International Conference on Applied Information Technology and Innovation: Toward A New Paradigm for the Design of Assistive Technology in Smart Home Care*, 114–117. <https://doi.org/10.1109/ICAITI.2018.8686725>

- Sharma, R., & Patel, M. (2018). Toxic Comment Classification Using Neural Networks and Machine Learning. *Iarjset*, 5(9), 47–52. <https://doi.org/10.17148/iarjset.2018.597>
- Si, S., Datta, A., Banerjee, S., & Naskar, S. K. (2019). Aggression Detection on Multilingual Social Media Text. *2019 10th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2019*, 1–5. <https://doi.org/10.1109/ICCCNT45670.2019.8944868>
- Sibhat, H. N. (2018). Spreading Hatred A study of Facebook in Ethiopia. *Global Media Review (GMR)*, 1(2), 1–10. issn: 2456-5652, <http://www.amitymediajournal.com>
- Singh, V., Varshney, A., Akhtar, S. S., Vijay, D., & Shrivastava, M. (2019a). *Aggression Detection on Social Media Text Using Deep Neural Networks*. 43–50. <https://doi.org/10.18653/v1/w18-5106>
- Singh, V., Varshney, A., Akhtar, S. S., Vijay, D., & Shrivastava, M. (2019b). *Aggression Detection on Social Media Text Using Deep Neural Networks*. January 2019, 43–50. <https://doi.org/10.18653/v1/w18-5106>
- Srinandhini, B., & Sheeba, J. I. (2015). Online social network bullying detection using intelligence techniques. *Procedia Computer Science*, 45(C), 485–492. <https://doi.org/10.1016/j.procs.2015.03.085>
- Tommasel, A., Rodriguez, J. M., & Godoy, D. (2018). Textual Aggression Detection through Deep Learning. *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying ({TRAC}-2018)*, 177–187. <https://www.aclweb.org/anthology/W18-4421>
- Verma, P., & Khandelwal, B. (2019). *Word Embeddings and Its Application in Deep Learning*. 11, 337–341. <https://doi.org/10.35940/ijitee.K1343.0981119>
- Viera, A. J., & Garrett, J. M. (2005). Anthony J. Viera, MD; Joanne M. Garrett, PhD (2005). Understanding interobserver agreement: the kappa statistic. *Fam Med* 2005;37(5):360-63. *Family Medicine*, 37(5), 360–363. http://www1.cs.columbia.edu/~julia/courses/CS6998/Interrater_agreement.Kappa_statistic.pdf
- Worku, K. (2013). Automatic Amharic text news classification: Aneural networks approach. *Ethiopian Journal of Science and Technology*, 6(2), 127–137.
- Yimam, S. M., Ayele, A. A., & Biemann, C. (2019). Analysis of the ethiopic twitter dataset for abusive speech in amharic. *ArXiv*, 1–5.
- Zewdie, M., & Wang, J.-H. (2018). Social Network Hate Speech Detection for Amharic Language. *4th International Conference on Natural Language Computing*, 41–55. <https://doi.org/10.5121/csit.2018.80604>

APPENDIX

Appendix A: Sample Source Codes

Appendix A.1 Sample Code for Word2Vec word embedding

```
In [9]: > import gensim
        from gensim import models
        from gensim.models import Word2Vec
```

```
In [10]: > model = gensim.models.Word2Vec(review_lines,size=400, window=5, min_count=2, workers=4,sg=1,iter=10)
```

Appendix A.2 Sample Code for TSNE to visualize FastText word embedding

```
In [21]: > import bokeh.plotting as bp
        from bokeh.models import HoverTool,BoxSelectTool
        from bokeh.plotting import figure,show,output_notebook
        #defining the chart
        output_notebook()
        plot_chart=bp.figure(plot_width=700,plot_height=600,title="A map of 50000 word vectors",tools="pan,
        #extracting the list of word vectors,limiting to 5000,each is of 200 dimensions
        word_vectors=[model[w] for w in list(model.wv.vocab.keys())[:5000]]
        #Reducing dimensionality by converting the vectors to 2d vectors
        from sklearn.manifold import TSNE
        tsne_model=TSNE(n_components=2,verbose=1,random_state=0)
        tsne_w2v=tsne_model.fit_transform(word_vectors)
        #storing data in a data frame
        tsne_df=pd.DataFrame(tsne_w2v,columns=['x','y'])
        tsne_df['words']=list(model.wv.vocab.keys())[:5000]
        #corresponnding word appears when you hover on the data point
        plot_chart.scatter(x='x',y='y',source=tsne_df)
        hover=plot_chart.select(dict(type=HoverTool))
        hover.tooltips={"word": "@words"}
        show(plot_chart)
```

Appendix A.3 Sample Code for Building and Evaluating Models

```
In [32]: > import numpy as np
        from tensorflow import keras
        from tensorflow.keras import layers
```

```
In [54]: > # Input for variable-length sequences of integers
        inputs = keras.Input(shape=(None,), dtype="int32")
        # Embed each integer in a 128-dimensional vector
        x = layers.Embedding(MAX_NB_WORDS, 128)(inputs)
        # Add 2 bidirectional LSTMs
        x = layers.Bidirectional(layers.LSTM(128, return_sequences=True))(x)
        x = layers.Bidirectional(layers.LSTM(128))(x)
        # Add a classifier
        outputs = layers.Dense(3, activation="softmax")(x)
        model = keras.Model(inputs, outputs)
        model.summary()
```

Appendix A.4 Sample Code for preprocessing of Amharic text

```
text = re.sub(r"١", "U", text)
text = re.sub(r"٢", "U", text)
text = re.sub(r"٣", "Z", text)
text = re.sub(r"٤", "U", text)
text = re.sub(r"٥", "Z", text)
text = re.sub(r"٦", "U", text)
text = re.sub(r"٧", "U", text)
text = re.sub(r"٨", "U", text)
```

```
text = re.sub(r"١", "0", text)
text = re.sub(r"٢", "0", text)
text = re.sub(r"٣", "4", text)
text = re.sub(r"٥", "0", text)
text = re.sub(r"٦", "4", text)
text = re.sub(r"٧", "0", text)
text = re.sub(r"٨", "0", text)
```

```
text = re.sub(r"ω", "Ω", text)
text = re.sub(r"ω", "Ω", text)
text = re.sub(r"υ", "Υ", text)
text = re.sub(r"υ", "Υ", text)
text = re.sub(r"υ", "Υ", text)
text = re.sub(r"υ", "Υ", text)
text = re.sub(r"υ", "Υ", text)
```

```

text = re.sub(r"ጠ/ፍርድ", r"ጠቅላይ ፍርድ ", text)
text = re.sub(r"ጽ/ቤ", r"ጽህፈት ቤት", text)
text = re.sub(r"ጽ/ቤት", r"ጽህፈት ቤት", text)
text = re.sub(r"ፌ/ዴ/ሪ", r"ፌዴራላዊ ዲሞክራሲ ሪፐብሊክ", text)
text = re.sub(r"ፌ/ፖ", r"ፌዴራል ፖሊስ", text)
text = re.sub(r"ፍ/ቤት", r"ፍርድ ቤት", text)
text = re.sub(r"መኢሰን", r"መላ ኢትዮጵያ ሶሻሊስት ንቅናቄ ", text)
text = re.sub(r"ኢዴፓ", r"የኢትዮጵያዊያን ዲሞክራሲያዊ ፓርቲ", text)
text = re.sub(r"ፕ/ር", r"ፕሮፌሰር", text)
text = re.sub(r"ፕ/ት", r"ፕሬዚዳንት", text)

```

```
text = re.sub(r'[0-9]+', r' ', text)
text = re.sub(r'[ð ð ě ō ē ē ē ē ē ī ſ ů ū ŷ æ œ þ ē ē ]+', r' ', text)
```

```

text = re.sub(r'[: :: ! i ÷ :- * *]+', r' ', text)
#text = re.sub(r', ! @$%^&*?]+', r' ', text)
#text = re.sub(r'<><>« »=:~_]+', r' ', text)
text = re.sub(r'^\w+', ' ', text) # avoid special char (!) (@) (,) (?) (#) (.)
text = re.sub(r'\s\s+', ' ', text) #avoid multi whitespace

```

```
text = re.sub(r'[a-z A-Z]+', r' ', text)
```

Appendix A.5 Sample Code for imbalance learning

```
In [18]: from imblearn.over_sampling import SMOTE
from collections import Counter
#split the training data into a training set and a validation set
indices = np.arange(trainvalid_data.shape[0])
np.random.shuffle(indices)
trainvalid_data = trainvalid_data[indices]
trainvalid_labels = trainvalid_labels[indices]
num_validation_samples = int(VALIDATION_SPLIT * trainvalid_data.shape[0])

x_train = trainvalid_data[:-num_validation_samples]
y_train = trainvalid_labels[:-num_validation_samples]
x_val = trainvalid_data[-num_validation_samples:]
y_val = trainvalid_labels[-num_validation_samples:]

sm = SMOTE()
x_smote, y_smote = sm.fit_sample(x_train, y_train)

print('Splitting the training data into train and valid is done')
```

Appendix A.6 Sample Code for model evaluation using confusion matrix

```
In [63]: import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix

def plot_cm(labels, predictions, p=0.5):
    cm = confusion_matrix(labels, predictions)
    plt.figure(figsize=(5, 5))
    sns.heatmap(cm, annot=True, fmt="d")
    plt.title("Confusion matrix")
    plt.ylabel("Actual label")
    plt.xlabel("Predicted label")

plot_cm(test_labels.argmax(axis=1), pred.argmax(axis=1))
```

Appendix B. Sample preprocessing data removing stop word and emoji

```
In [7]: df['Processed_Text'] = df.text.apply(lambda x: clean_text(x))
```

```
In [8]: df.head(4)
```

Out[8]:

	text	labels	Processed_Text
0	እንተ አሁን ዝም ብለህ ተጫን 🤔🤔🤔🤔	1	አሁን ዝም ብለህ ተጫን
1	ዘረቢህ	2	ዘረቢህ
2	ችጋርም	2	ችጋርም
3	ከአሁን ጋር እንዴት እንኑር	1	ከአሁን እንዴት እንኑር

Appendix C. List of stop words

```
[ "ይጠበቃል", "አስታውቀዋል", "ግን", "ነው", "ነህ", "ናችሁ", "ነበር", "ሆኖም", "ነገርግን", "እና", "ገለፁ", "አስታወቀ",  
 "ዘግበዋል", "ተጠናቀቀ", "ሆነ", "ሁሉ", "ይካሄዳል", "ሁለም", "ጎላ", "ሁኔታ", "ሆነ", "ምን", "ምንም", "ሆኖም", "ሁሌ",  
 "ሁሉንም", "ላይ", "ሌላ", "ሌሎች", "ለዩ", "መሆኑ", "ማለት", "ማለቱ", "መካከል", "የሚገኙ", "የሚገኝ", "ማድረግ",  
 "ማን", "ማንም", "ሰሞኑን", "ሲሆን", "ሲል", "ሲሉ", "ስለ", "ቢሆን", "ብለዋል", "ብቻ", "ብዛት", "ብዙ", "ቦታ", "በርካታ",  
 "በሰሞኑ", "በታች", "በጎላ", "በኩል", "በውስጥ", "በጣም", "ብቻ", "በተለይ", "በተመለከተ", "በተመሳሳይ", "የተለያዩ", "የተለያየ",  
 "ይሁን", "ተባለ", "ተገለፁ", "ተገልጿል", "ተጨማሪ", "ተከናውኗል", "ታዲያ", "ችግር", "ታች", "ትናንት", "ነበረች", "ነበሩ", "ነበረ",  
 "ነይ", "ነገር", "ነገሮች", "ናት", "ናቸው", "አንተ", "አንቺ", "እናንተ", "እነሱ", "እነዚህ", "እነዚያ", "አሁን", "አለ", "እስካሁን", "አሳሰበ",  
 "አሳሰበዋል", "አስገንዝበዋል", "አብራርተዋል", "አስረድተዋል", "እስከ", "እባክሽ", "እባክህ", "እባክም", "አንፃር", "እስኪደርስ", "እንኳ",  
 "እስከ", "እዚሁ", "እና", "እንደገና", "እንዳስረዱት", "እንዲሁም", "እንጂ", "እዚህ", "እዚያ", "እያንዳንዱ", "እያንዳንዳችሁ",  
 "እያንዳንዱ", "ከጎላ", "ከላይ", "ከመካከል", "ከሰሞኑ", "ከታች", "ከውስጥ", "ከጋራ", "ከፊት", "ወዘተ", "ወይም", "ወደ", "ዋና",  
 "ወደፊት", "ውስጥ", "ውጪ", "ያለ", "ያሉ", "ይገባል", "የጎላ", "የሰሞኑ", "የውስጥ", "የጋራ", "ይታወሳል", "ይህ", "ደግሞ", "ድረስ",  
 "ገልጿል", "ገልጸዋል", "ጊዜ", "ጥቂት", "ፊት", "ዛሬ", "ጋር", "ተናግረዋል", "የገለጹት", "ይገልጻሉ", "ብለዋል", "ስለሆነ", "አቶ",  
 "ሆኖም", "አመልክተዋል", "ይናገራሉ"]
```

Appendix D. Annotation Guidelines

Aggressive language varies greatly, ranging from simple profanity to much more severe types of high toxic language. Overt aggression is any speech or text in which aggression is overtly expressed, either through the use of specific kind of lexical items, lexical features or certain syntactic structures, considered aggressive text. On the other hand, covert aggression is any text in which aggression is not overtly expressed. It is an indirect attack against the victim and is often packaged as insincere polite expressions, through the use of conventionalized polite structures. For instance, cases of satire or rhetorical questions may be classified as covert aggression. To operationalize the task of aggression detection, we developed guidelines and briefly present the annotation steps below.

Overtly Aggressive (OAG): Texts which express aggression in a direct, straightforward, and explicit way. Toxic, insult, obscene and threat, offensive and hate speech with direct expression are considered as overly aggressive (OAG).

E.g., “አንተ ደደብ ስምህን ቄስ የጥራው”, “ደነዝ”, “አንተ ድርቅ በል ለማኝ ባንዳ”.

Covertly Aggressive (CAG): Texts that express aggression in an indirect, sarcastic manner. Sarcasm, irony, and rhetorical questions have in common that the meaning of the comment is different from its literal meaning which uses the property of objects or animals to cover the hate or insult. E.g., “ይህች ባቄላ ካደረች አትቆረጠምም እንዘፍዝፋት” , “ጉልቻ ግዜህን ጠብቅ”, “ተስፋ የቆረጠ ጅብ ነክሶ ይሞታል”, አንዴ እሳቱን ለኩሳሃል.

Not Aggressive (NAG): Texts which are not aggressive comments or other than covertly and overtly Amharic comments in the above.

E.g., “ክብር ለእናቶች ይሁን”, “እናመሰግናለን በርቱልን እኛም ከጎናችሁ ነን”, “ጥሩ ነው በርቱ”.