Addis Ababa University

Department of Computer Science

Hate Speech Detection Framework from Social Media Content: The Case of Afaan Oromoo Language

Lata Guta kanessaa

A Thesis Submitted to the Department of Computer Science in Partial Fulfilment for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia

December 2, 2021

G.C

Addis Ababa University

Department of Computer Science

Lata Guta Kanessa

Advisor: Solomon Gizaw (PhD)

This is to certify that the thesis prepared by Lata Guta Kanessa.

Title: *Hate Speech Detection Framework from Social Media Content: The Case of Afaan Oromoo Language* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

	Name	Signature	Date
Advisor:			
Examiner:			
Examiner:			

Abstraction

Hate Speech on social media has unfortunately become a common occurrence in the Ethiopia online community largely due to advances in mobile computing and the Internet. The connectivity and availability of social media platforms in the world allow people to Interact and interchange experiences easily. However, the anonymity and flexibility afforded by the Internet have made it easy for users to communicate aggressively. Hate Speech affects the society in many aspects, such as affecting the mental health of targeted audiences, affects social interaction, leads to violence and distraction of properties.

Identifying a text that containing Hate Speech regularly is difficult task for humans, it is tedious and time consuming.

To solve the newly emerged Hate Speech propagation in social media sites, recent studies employed different Machine learning algorithms and feature engineering techniques to detect Hate Speech messages automatically. In case of Afaan Oromoo language there is a work on Sentiment Analysis of Afaan Oromo using Machine learning Approach.but it is not in case of Hate and neutral classification rather oponions.

In this research, a new Afaan Oromoo Hate Speech dataset from Facebook social media that are labeled into binary classes. TF-IDF, N-gram and word2ve feature are used as a feature for the Machine learning models. We evaluate the models using 80% for training and 20% for testing purpose by using train-test split with accuracy, precession, recall, and f1-score performance metrics were used to compare the models. The model based on LSVM with TF-IDF combination with N-gram achieves slightly better performance than the other models. Support Vector Machine(SVM) algorithm achieve the highest accuracy of 96% which is promised result.

Acknowledgments

I would like to acknowledge God for His grace, strength and for his countless blessings throughout my life and protect with good health as I undertook this research.

My sincere gratitude to Dilla university School of Research and Postgraduate Studies for covering the fund of this research. My sincere gratitude to my advisor Dr. Solomon Gizaw for his friendly approach,

readiness and willingness to advise on the research, his critical comments, commitment to guide, and support greatly improved this thesis work. I would like to thanks all my classmates who are working on their master's thesis for their timely help, idea, and support until the completion of my thesis.

My special thanks also go to Mr. Yesuf Kasahun for providing expertise and insights into Hate Speech laws and helping me to draft Hate Speech annotation Guidelines. Also, I would like to thanks Mr. Taye Aga, and Ife Bodasa for their major role in the dataset annotation process as annotator and for anyone who participated or contribution to this work.

Dedication

To my beloved Mother:Agamti Awajii , my Father:Guutaa Kanessaa, my wife Ife Bodasa and also my sister and brothers. Thank you for your continued support and prayers always for me.

Table of Contents

List of Tables

List of Figures

List of Equations

List of Algorithms

Acronyms and Abbreviation

AOHS	Afaan Oromoo Hate Speech
AOHSD	Afaan Oromoo Hate Speech detection
GUI	Graphical user interface
BOW	Bag of words
BPTT	Back-propagation through time
CNN	Convolution neural network
DNN	Deep neural network
DT	Decision trees
FB	Facebook
FNN	Feedforward networks
GDBT	Gradient boosted decion trees
HS	Hate Speech
KNN	K-nearest Neighbor Classifiers
LR	Logistic Regression
LSTM	Long short-term memory
ML	Machine Learning
MLP	Multilayer perceptron
NB	Nave Bayes
NLP	Natural Language Processing
OONI	Open observatory of network interference
POS	Part-of-Speech tagging
RF	Random Forests Machines
RNN	Recurrent neural networks
SVM	Support vector Machine

Chapter:1 Introduction.

1.1 Background

Over the last decades, people are being more engaged with the widespread of social media. social media have also become the place for Hate Speech proliferation by which most people's social lives are disturbed because of Hate Speech posts and convicts triggered by those posts [1]. Social media is defined as web-based and mobile-based Internet applications that allow the creation, access, and exchange of user-generated content that is ubiquitously accessible [2]. The improvements in mobile computing and the Internet have in increased the usage of social media [3]. Social media such as Facebook, Twitter, and LinkedIn has been increasingly exploited for the propagation of Hate Speech and the organization of Hate-based activity [4]. Social media has a positive and negative impact on the Social, Economic and Political of one country. The positive impacts are it helps people to exchange opinions digitally which information is disseminated quickly and widely. The negative impact is Hate Speech dissemination, which is attacking people based on common characteristics such as color, religion, ethnicity and gender [5]. The Increase of social media use in all modern societies has dramatically changed the way people interact with each other [6]. Moreover, anonymity and mobility features enable individuals to hide behind a screen and spread Hate Speech against individuals or groups of people [7]. According to Biere and bhulai [8] Hate Speech has the goal of achieving at most two dreams, which is striving to inform bigots and to intimidate the targeted minority, leading them to accuse whether their dignity and social fame are at ease. Drawing upon these definitions, we adopt the definition of Hate Speech that is used to express hatred towards a targeted group or is intended to be derogatory of the members of the group [9]. Likewise, most companies like Facebook and Twitter are criticized for not doing enough to prevent Hate Speech on their platform and have come under pressure to take action against Hate Speech [10]. In the first quarter of 2018 Facebook removed two million pieces of bad content from its platform much of which was done through new mechanisms of algorithmic filtering [11]. Most developed countries are passing Hate Speech regulations and pressuring social media companies to implement policies to stop the spread of online Hate Speech [12].

In Ethiopia, Hate Speech and fake news in recent times are being blamed as the catalyst especially for ethnic-related violence in various parts of the country [13]. Facebook is a widely used social media site by most of the community members, according to stat counter global status data from April 2019 until April 2020 from the overall 74.64% of social media usage is held by Facebook [14]. Facebook users in Ethiopia use different languages to propagate Hate Speech which triggers deadly ethnic clashes among people through ugly Facebook content, which is a language supported by Facebook [15]. The Ethiopian government also believes that social media encounters aggravating the pace of Hate Speech and affecting the country’s development path [16]. The government of Ethiopia monitors contents of social media specifically to prevent harmful rhetoric messages through many time interruptions of the internet service and by blocking these sites from being accessible in the country [17]. In addition, introduces a law to encompass the online Hate Speech through the expansion of anti-terrorism laws, and the law prohibits the dissemination of terrorizing message subjecting violators to a prison

sentence of up to eight years [18]. Currently, Hate Speech detection has been an increasing trending subject over the past few years. To overcome social media problems many types of research are done for the English language also, there was some research for Italian Hate Speech detection [19], Hate Speech detection for the Amharic Language done by Mossie and Wang [20] and chine Hate Speech detection Tan and Zhang [21].

The popular Machine learning algorithms used for Hate Speech detections are Supportive Vector Machine [22], Random Forest [23], Naïve Bayes [24], and Logistic Regression [25]. The most common techniques used in Hate Speech Detection in Social Media are the Lexicon approach, dictionary- based approach [26], Bag of words [27], Word Embedding [28]. TF-IDF also another approach that measures the importance of a word in a document within a corpus and increases in proportion to the number of times that a word appears in the document. Social media currently provide localization, which allows the user to use different world languages on their sites. One of these languages is Afaan Oromoo; Afaan Oromoo languages are one of the widely spoken languages and working languages of the Oromiya regional state [29]. Afaan Oromoo language is still under-resourced that has few computing tools. Nowadays, in Oromiyaa, it is an open secret that the recent widespread Hate Speech and call for violence and attacks on particular targets of individuals or groups based on their political view, ethnic origin, and religious affiliation [30]. Therefore, it is important to monitor or automatically detect Hate Speech on this platform to prevent their spread, and possibly reduce acts of violence and Hate crimes that destroy the lives of individuals, families, communities, and the country systematically.

This research work focuses on addressing the problem of Hate Speech using a new dataset that is annotated with two labels: Hate, Neutral. Finally, since there is no work done for Afaan Oromoo Hate Speech detection we proposed a Hate Speech detection framework for the Afaan Oromoo language by using a new dataset of posts and comments from the Facebook public page and using multiple feature extraction methods and SVM Machine learning classifier algorithms.

1.2 Motivation

The motivation of this research comes firstly from the growing influence and popularity of Hate Speech on a social media platform. Social media helps people to come under the same umbrella regardless of their national border, socio-economical background, and religion. As with that, Hate Speech can also spread rapidly which may be a reason for conflicts between different groups in our society. Now day Hate Speech is a serious and growing problem in Ethiopia mostly hatred is propagated both online and offline. To design a safe environment for society it is necessary to monitor Hate Speech which may lead to violence, and distraction of property.

Secondly, the scientific study of Hate Speech from a computer science point of view is recent, also Hate Speech has become a popular topic due to it has increased media coverage and also the growing political attention in recent time. Many developed countries such as the European Commission as founding several programs towards the fighting of Hate Speech [31] and pressured to remove any appearance of Hate Speech in their platforms in less than 24 hours [32]. Social media companies like Facebook are straggling to prevent Hate Speech from social media to make a safe environment. But there are no AI tools to detect automatically, for now, maybe we will have an AI to take the primary role in automatically detecting Hate Speech on Facebook

in 5 to 10 years [33].

The development of every country is dependent on the technology implemented as per their language. But a majority of the world's languages are still under-resourced this is particularly true for the most African country. Afaan Oromoo is one of the sub-Sahara country languages Where very few computational linguistic resources have been developed and very little has been done. So we propose to develop an Afaan Oromoo automatic Hate Speech detection system which makes Afaan Oromoo part of technology.

Generally, The motivation of this research is the increasing of Hate Speech propagation and manual moderation was challenged which leading to error. By doing this safe environment where society use without any verbal attack is created. Furthermore, to contributes to the development of a Hate Speech detection system and reduces the lack of Hate Speech datasets for future research.

1.3 Statement of problems

Monitoring Hate Speech content in traditional mainstream media such as radio and television is easier than monitoring online Hate Speech social media [34]. Because social media consists of large amount of user-generated content within a second that would need to be monitored. Several studies and applications have been done for different languages, most of the works have been done in English and other languages. Social media Hate Speech detection for the Amharic Language done by Mossie and Wang [13] use a binary class dataset for Hate Speech detection. Italian Hate Speech detection [22], code mix Hindi Hate Speech detection [35].

Most of the world's languages are still under-resourced in that they have very few or no language handling devices and resources which especially substantial for sub-Saharan African countries.

So it is important to automatically detect Hate Speech on social media platforms. The popular methods to detect Hate Speeches are lexically based, dictionary, Support Vector Machine, Naive Bayes, Decision Trees and Deep Neural Networks.

The Hate Speech detection techniques differ from language to language due to word formation, grammatical arrangement and type of Hate and neutral terms. Therefore, each term needs special consideration to return the correct results. The Afaan Oromoo language is one of under-resourced language which doesn't have more language processing tool and techniques like another language before. For Afaan Oromoo Language there is only one work done for Afaan Oromoo sentiment Analysis but this work is not for Hate Speech detection rather opinion classification. so, we propose to develop a Hate Speech detection framework that applies Machine learning techniques to automatically classify comments and posts as Hate Speech or neutral. Also, this study addresses these problems by constructing a new Afaan Oromoo dataset since there is no published dataset for this language, utilizing multiple Machine learning and feature extraction method.

This automatic classification will improve the process of detecting Hate Speech on social media by reducing the amount of time to filter out manually and the human effort required.

1.4 Objective

1.4.1 General objective

The general objective of this research is to develop Social media Hate Speech Detection Framework for the Afaan Oromoo language by using an SVM Machine learning algorithm.

1.4.2 Specific objective

To achieve the aforementioned general objectives, the following specific tasks have been performed:

- Review relevant works.
- Develop a Dataset for Afaan Oromoo Hate Speech.
- Develop annotation guidelines for labeling posts and comments.
- Design Afaan Oromoo Hate Speech detection model.
- Develop a prototype to demonstrate the model.
- Evaluate the performance of the Hate Speech detection system.

1.5 Scope and Limitation

1.5.1 Scope

This study is limited to detecting Hate Speech on the social media platform Facebook and only considered textual Facebook comments and posts expressed in Afaan Oromoo language. A new dataset is built by collects Afaan Oromoo text posts and comments from Facebook popular public pages for February 2020 to March 2021 and annotating the posts or comments into binary classes, which are Hate and Neutral Speech. Since it is the first time the task of Automatic Afaan Oromoo Hate Speech detection is done, we do not consider emoji gesture, image, and videos only consider text.

1.5.2 Limitation

This study is limited to classify the text-only into two categories not considered other Hate Speech classes. In addition, it does not cover different data contents of social media like image, audio, video content, and other emotional symbols. Since there is a lack of other studies for comparison Hate Speech detection for the Afaan Oromoo language, also a lack a share public dataset and model for Hate Speech 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 Hate Speech related law experts to consult.
- Due to the tight schedule, the study only implemented the proposed Machine learning classifier and limited it to develop and evaluate algorithms.

1.6 Methods

Machine learning approach, Hate Speech classifications are three phases that can be roughly distinguished in the life cycle of automatic classification systems. In the first phase, the comments and posts are crawled and annotate dataset, preprocessing of texts are held such as stopwords removal, removal of unnecessary characters,

short word expansion, stemming. The second phase is a building of a model which classifies Hate Speech and neutral text from predefined classes. Which is referred to as the classifier learning phase. In addition, the third is the classifier evaluation phase. Depending on each phase, the following methods are employed in conducting this research.

1.6.1 Literature review

For finding up-to-date methodologies in the Machine learning Hate Speech detection domain, a comprehensive literature review will be conducted. For this study, secondary data sources, like books, articles, publications, and other resources related to the topic will be reviewed. This helps to have a better understanding of the subject of the study.

- Classifier algorithms and their applications.
- Developed tools and techniques for Afaan Oromoo data and information processing.
- Current and past practices of posts and comments classification systems for social media platforms.

1.6.2 Selection of tools and library

In this study, some tools are used to capture comments and posts.

To capture comments and posts from Facebook pages we propose Facepager and ScrapeStorm software.

1.6.3 Data Sources

Data will be collected from Facebook pages, different websites, journals, educational books, and so on for understanding the characteristics of question types and their respective answers. where the sample was determined based on the judgment of the researcher with prior knowledge of characteristics of Facebook that constitutes Hate Speech.

1.6.4 Dataset preparation

To build an automatic Hate Speech detection dataset from comments and posts retrieved from Facebook public pages, a standard and representative document corpus ought to be selected. The dataset must be selected to perform an exhaustive evaluation of performance of automatic Hate Speech detection. Accordingly, the data is collected from Facebook public pages, from OBN(Oromia Broadcasting service), FIB(Finfinnee Integrate broadcast), and different Afaan Oromoo activists. These pages typically post discussions spanning across a variety of political and religious topics. It posts several videos per day on Facebook and typically receiving thousands of comments per the post. By using versatile Facebook crawler, which exploits the Graph API to retrieve the content of the comments from Facebook posts using Face pager. To this end, experts having comprehensive domain knowledge of the language are consulted.

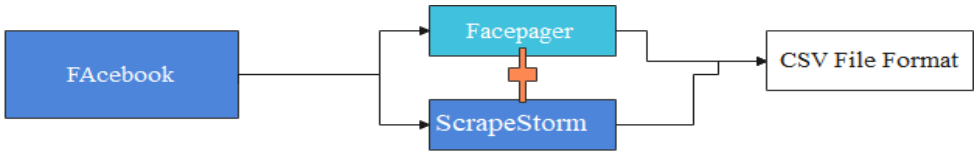

Figure 1.1 data collection techniques

1.6.5 Design Approach

In this study, we used Machine learning approaches. Since Machine learning make use of different techniques and feature extractions to automatically learn from the given data. So, we used the Supportive Vector Machine(SVM), Logistic Regression(LR), Naïve Bayes(NV), and Random Forest(RF). Since they are the most popular and higher potential Machine learning algoritjms.

Experiment: experiments are performed to evaluate the performance of the developed system.

1.7 Application of the Result

The result of this study is believed to be used as input for other researchers who need to advance the Hate Speech detection techniques. Detecting Hate Speech is important for online communities for maintaining safe environments for their users. A Hate Speech detection system will help for the reduction of time and human effort to identify a verbal attack on social media. The system will help to filter any hatred comment and post that makes people of the local population indirectly or directly participate in violent activities across the different regions of the country.

Simply purpose of this research is listed below, to easily understand.

- Input for other researchers.
- Create a safe environment or maintain a Healthy Environment in social media for all users.
- Minimize effort to monitor social media.
- Get rid of conflicts b/w groups in society by minimizing HS disseminated through social media.
- Reduce misuse of Social Media.
- Avoid unpleasant and embarrassing situations.

1.8 Summery

Several studies and applications have been done for Hate Speech detection by using different approaches. However, these methods generally belong to discriminative learning, which aims to distinguish one class from others with a clear-cut outcome, under the presence of ground truth. Most Machine learning algorithms are considered to belong to discriminative learning since they all aim to distinguish between one and other classes. However, these assumptions do not always hold the accurate result. For example “I love my country but I Hate immigrants” involves both positive and negative Speech [36]. So this paper proposes a Support vector Machine and other Machine learning algorithms to compare the performance of Hate Speech detection on social media.

1.9 Organization of the Thesis

The rest of the chapters in this report are organized as follows.

Chapter two deals with the literature reviewed so far. It includes social media definition, Hate Speech in Ethiopia, a concept related to automatic Hate Speech detection, different Machine learning approach, feature extraction, and algorithm performance matrix. In addition, the Afaan Oromoo language Writing system and common algorithm for Hate Speech detection are defined.

Chapter three discusses the existing papers related to Hate Speech detection published is reviewed. Generally few resource language and resource riched language is discussed at the last challenges of Hate Speech detection is discussed.

Chapter four describes the Design and Implementation of the proposed system.

Overview of architecture, description of components such as dataset, preprocessing, feature extraction, and classification are discussed.

Chapter five is the system implementation and experiments.

Development tools and techniques, deployment environment, dataset description, implementation of preprocessing, implementation of feature extraction, classifier, and a prototype is discussed.

Chapter six Result and Discussions of experiments finally, conclusions drawn from the thesis result, the contributions of this research work, and recommendations on possible future works related to this research.

Chapter:2 Literature Review.

2.1 Introduction

This chapter reviews relevant literature to discuss theoretical foundations used for solving the problem of Afaan Oromoo text social media Hate Speech detection. The first section contains a basic definition of social media, Hate Speech detection, the nature of Hate Speech in Ethiopia, and current processes to monitor Hate Speech on social media is reviewed. In addition, we review Machine learning techniques in Hate Speech detection and common algorithm. Generally**,** panoptic kinds of literature have been reviewed not only to understand the problem associated with the area of this thesis but also to bring the appropriate solution.

2.2 Hate Speech on Social Media

Social media companies and governments are under growing pressure to find new solutions to the toxification of online social media conversation. Hate Speech is a complex phenomenon, intrinsically associated with relationships between groups, and relying on language nuances [37]. Hate Speech on social media is adverse not just to the wellness of users also, has an impact on an open egalitarian society [38]. A social media platform makes a perfect place for anonymity to enable individuals to hide their identity behind a screen and spread Hate Speech against individuals or groups of people [8]. These online platforms are often abused and misused to spread content that can attack, Hate against groups or individuals based on protected characters such as race, color, religion. So the word "Hate" will be understood as "extreme negative feelings and beliefs held about a group of individuals or a specific representative of that group because of their race, ethnicity, religion, gender or sexual orientation.

We summarize leading definitions of Hate Speech from varying sources; these sources are social network platforms, governmental organizations, the UN, Scientific communities. The following are a list of definitions and sources of definitions:

1. Encyclopedia of the American Constitution**.** "*Hate Speech is Speech that attacks a person or Group based on attributes such as race, religion, ethnic origin, national origin, sex, disability, Sexual orientation or gender identity*" [39].
2. UN's International Committee on the Elimination of Racial Discrimination, "*Hate Speech as a form of other-directed Speech which rejects the core human rights principles of human dignity and equality and seeks to degrade the standing of individuals and groups in the estimation of society"* [40].
3. De Gilbert et al. "*Hate Speech is a deliberate attack directed towards a specific group of people motivated by aspects of the group's identity"* [41].
4. Fortuna et al. *"Hate Speech is a language that attacks or diminishes, that incites violence or Hate against groups, based on specific characteristics such as physical appearance, religion, descent, national or ethnic origin, sexual orientation, gender identity or other, and it can occur with different linguistic styles, even in subtle forms or when humour is used"* [42].

5. Legal and academic literature generally defines "*Hate Speech as Speech or any form of expression that expresses hatred against a person or group of people because of a characteristic they share, or a group to which they belong"* [43]. Hate Speech can occur in different linguistic styles and several acts like insulting, provocation, abuse, and aggression.

2.2.1 Definition of Hate Speech in social media Tools

Table 2.1 Comparison of Hate Speech definitions across social media tools

Source	Definition
Facebook:	"Content that attacks people based on their actual or perceived race, ethnicity, national origin, and religion, and sex, gender in a form of post or comment mostly in textual form. We define attack as violent or dehumanizing Speech, statements of inferiority, or calls for exclusion or segregation."
YouTube	"Hate Speech refers to content that promotes violence or hatred against individuals or groups of people but there is a fine line between what is and what is not considered to be Hate Speech. For instance, it is generally okay to criticize a nation-state, but not okay to post malicious hateful comments about a group of people solely based on their ethnicity."
Twitter:	"Hateful conduct: You may not promote violence against or directly attack or threaten other people Based on race, ethnicity, national origin, sexual orientation, gender, gender identity, religious affiliation, age, disability, or disease."

Table 2.2 Comparison of Hate Speech definitions across time and institutions

Scientific paper:	Language, which attacks or demeans a group based on race, ethnic origin, religious disability, gender, age, disability, or sexual orientation/gender identity. All conduct publicly inciting to violence or hatred directed against a group of persons or a member of such a group defined by reference to race, color, religion, descent, or national or ethnic
ILGA:	Hate Speech is public expressions, which spread, incite, promote or justify hatred, discrimination, or hostility towards a specific group. They contribute to a general climate of intolerance, which in turn makes attacks more probable against those given groups.
EC 2016:	Code of conduct on countering illegal Hate Speech online" to help users to flag illegal Hate Speech in these social platforms (Facebook, Microsoft, Twitter, and YouTube), improve the civil discourse, and increase coordination with national authorities.

Table 2.3 Hate Speech and related terms

Source	Definition	Comparison with Hate Speech
Cyberbullying	The electronic posting of mean-spirited messages about a person. Often done anonymously.	Hate Speech does not include verbal attacks toward specific individuals. It is typically addressed towards a group of people or a member of a community. Personal attacks are not included in the definition.
Discrimination	Prejudiced or prejudicial outlook, action, or unfair treatment	Hate Speech takes place only through verbal means.
Toxic language	Toxic use of the language is a synonym of aggressive language, used to hurt. It is disrespectful and leads the interlocutors to leave the conversation.	Hate Speech can be toxic; however, it is also able to trigger more discussion over a topic.
Hate	The feeling of aversion for or extreme hostility toward a target without a stated explanation for it.	Hate is a general expression of hatred, while Hate Speech has specific targets towards whom one addresses offensive content.

2.2.2 Hate Speech detection in Ethiopia

Hate Speech in Ethiopia is growing with the increase of social media users in the country. The widespread Hate Speech on social media platforms is an open secret. However, there is no system developed to detect Hate Speech automatically. Since there is no tool to detect automatically government of Ethiopia mostly block social media and file-sharing platforms such as Facebook, Twitter, WhatsApp, and Dropbox [44]. But this does not give a solution therefore, critical importance to monitor and identify instances of Hate Speech, as soon as possible to prevent their spread and possibly unfolding into acts of violence or Hate crimes.

In the case of Ethiopia, the use of Hateful words to bring about hatred against a group of people based on their ethnicity, political attitude, religion and socio-economic are prevailing. The anonymity of social networks makes it attractive for Hate Speech to mask their criminal activities online posing a challenge to the world and in particular Ethiopia.

Ethiopian Hate Speech and Disinformation Prevention and Suppression Proclamation Page 12339 under Proclamation No. 1185 /2020 “Hate Speech” *is the Speech that purposefully promotes discrimination, hatred, or attack against a discernable group of identity or person, based on race, ethnicity, gender, religion or disability”*. Based on the above proclamation, we outline Hate Speech as any kind of text format which incites violence or against individuals and groups, based on common characteristics.

- It makes use of disparaging terms with the intent to harm or incite harm between societies.
- It is targeted against a person or group of people of the same or different language.

The new Ethiopian Government regulation defined Hate Speech as an expression that incites, demeans, threatens, discriminates, or harasses another person, race, religion, color, sex, disability, nationality, immigration, language, appearance, discrimination, or harassment based on, in a manner that provokes, demeans, threatens, discriminates, or causes violence. Translated to Afaan Oromoo as *haasaa jibbinsaa jechuun gartuu ykn Saba tokko Afaan isaan, bifa isaan, aadaa isan, amantii isaan, saala isan dhiibbaa jechaa irraan ga'uu jechuudha.*

2.3 Overview of Afaan Oromoo Language

Afaan Oromoo is the language of the Oromoo people who comprises the largest ethnic group in Ethiopia.

The Afaan Oromoo language is one of under-resourced language which doesn't have more language processing tool and techniques like other languages have which could help computational linguistic researchers to go more detail and develop useful higher-level Internet or computer-based applications and models. Afaan Oromo is an official language of the Oromia regional state and is the medium of instruction for the primary schools in the region. Currently, there are a growing number of publications in hard copies and a vast amount of information in electronic formats for Afaan Oromoo. Afaan Oromoo belongs to the Cushitic branch of the Afro-Asiatic language family along with Somali, Afar, and some other languages. Oromoo people use Latin scripts for the writing system officially. Afaan Oromo is a language that is used in a wide area of the country as a result there is a dialectical variation. According to [45], four major categories can be identified. These are Western (Wellega, Iluababor, Kaffa, and parts of Gojjam), Eastern (Harar, Eastern showa, and parts of Arsi and Bale), Central (Central Showa, Western Showa and possibly Wollo), and Southern (Parts of Arsi, Sidamo, and Borena).

2.3.1 The Oromo Alphabet

The alphabets of Afaan Oromo are often called "Qubee Afaan Oromoo", alphabets of the Oromo language. The major representatives of sources of sound in a language are vowels and consonants. Afaan Oromo has 36 basic sounds (10 vowels and 26 consonants). Afaan Oromoo language has its language formulation, which is the front/back position of the tongue that gives sound in the mouth. While high/low in the place of the tongue with concerning the palate during articulation. Most vowels can appear in initial, medial, and final positions in a word in the Afaan Oromoo language. The following examples show some long vowels at word-initial, medial, and final positions.

Initial positions: uumaa to mean 'nature', eelee to mean 'pan',

Medial position: keennaa to mean 'gift', leexaa to mean 'single'

Final position: garaa to mean 'belly', daaraa to mean 'ash'

The Afaan Oromoo vowels represented by letters (a, e, o, u, and i) are called "Dubbachiistuu" in Afaan Oromo, and the consonants known as "dubbifamaa" in Afaan Oromo. As shown in Table 1 below.

Table 2.4 Dubbachiistoota (vowels)

	Front	Central	Back
High	i, ii		u, uu
Mid	e, ee		o, oo
Low		a, aa	

Table 2.5 Major places of articulations. [44]

Place	Active Articulator	Passive Articulator
Bilabial	Lower lip	Upper lip
Labio-dental	Lower lip	Upper teeth
Dental	Tip of tongue	Upper teeth
Alveolar	Blade of tongue	Alveolar ridge
Retroflex	Tip of tongue	Hard palate
Palatal	Front of tongue	Hard palate
Velar	Middle of tongue	Velum (Soft Palate)
Uvular	Back of tongue	Uvula

2.3.2 The Afaan Oromo Consonants

In Afaan Oromoo language, consonants cannot give meaning alone but it gives meaning in combination with a vowel. All Afaan Oromo consonants except the combination consonants ny, dh, ph, and sh have double consonant combinations if the syllable is stressed.

Table 2.6 Dubbifamtoota (consonants)

		Bilabial Labiodental	Alveolar Retroflex	Palato-Alveolar Palatal	Velar	Glotal
Stops and affricates	Voiceless	(p)	T	ch	k	?
	Voiced	(b)	D	j	g	
	Ejective	ph/p'/	x/t'/	c	q	
	Implosive		dh/d/			
Fricatives	Voiceless	F	S	sh		h
	Voiced	(v)	(z)			
Nasals		M	N	ny		
Approximants			L	y		
Rhotic		W	R			

2.3.3 Morphology

Natural Language Processing is the application of computational models to tasks involving human language text. In every language, whether it is spoken or written, every meaningful pattern has its structure, and the elements of the language of language are understandably related to each other. Words are the basic elements of a language and are formed from morphemes, which constitute the smallest meaningful unit of speech in a language. This is also true for Afaan Oromoo, which has its own rules of words, and/or sentence structure [46]. Morphology is a way of studying the language word's structure. It is about the way words are put together, their internal structure. The smallest meaningful units of forms are called "morphemes" which are either "free" or "bound". A free morpheme can occur on its own whereas bound morphemes do not occur alone. Bound morphemes are of three types, these are, prefix attached to the initial positions, infix inserted in the middle, and suffix attached to the final position of the word. All three types of bound morphemes are called "affixes". There are two main branches of morphology are inflectional and derivational morphology Inflectional morphology studies the inflectional changes in words that generally do not result in changing the classes of words. Derivational morphology results in changing classes of words.

2.3.4 The Afaan Oromo Morphology

Every language has its morphological structure that defines rules used for combining the different components the language may have. The English language for instance is different in its morphological structure from French, Arabic, or Afaan Oromoo [47]. There are some word-formation processes in Afaan Oromoo. Affixation and compounding are among these word-formation processes.

Affixation is generally described as the addition of affixes at the beginning, in between, and/or at the end of a root/stem depending on whether the affix is a prefix, infix, or suffix.

Compounding is the joining of two linguistic forms, which functions independently. Examples of compound nouns include; abbaa-buddenaa 'step father' from abba- 'father' and buddena 'food'. Like some other African and Ethiopian languages, Afaan Oromo has a very complex and rich morphology. It has the basic features of agglutinative languages involving very extensive inflectional and derivational morphological processes.

In Afaan Oromo language, most of the grammatical information is conveyed through affixes (i.e. prefixes and suffixes) attached to the root or stem of words. These high inflectional forms and extensive derivational features of the language are presenting various challenges for text processing and information retrieval experiments in Afaan Oromo. Although, almost all Oromo nouns in a given text have persons, number, gender, and possession makers, which are concatenated and affixed to a stem or singular noun, form. In addition, Afaan Oromo noun plural markers/forms can have several alternatives. Very common plural markers in Afaan Oromo including: -oota, -wwan, -lee, -an, -een, -eeyyii, -oo, etc...). As an example, the Afaan Oromo singular noun "mana" (house) can take the following different plural forms: Manoota (mana + oota), manneen (mana + een), manawwan (mana + wwan. The construction and usages of such alternative affixes and Attachments are governed by the morphological and syntactic rules of the language [48].

2.4 Challenges of Afaan Oromoo Hate Speech Detection

One of the main challenges of Hate Speech detection is the data source and data annotation, since there is no defined rule for labeling Afaan Oromoo Hate Speech texts and how many classes it has. Another challenge is ambiguity. In Afaan Oromoo the same word may refer to different categories of Hate Speech, for example, Qu*cii, G*la, Cinoo*tuu can be considered as insulting or refer specific target group. In addition, Afaan Oromoo language has different from one place to another that may cause conflict during communication. As an example in Arse the word bukkee (the lower part of the female reproductive tract) is Hate Speech but neutral in wellaga means (Neighboring).

2.5 Existing Hate Speech Detection approaches

The study of social media Hate Speech detection has been growing only in recent time. However, some studies have already been done in few languages. Researches focusing on algorithms for social media Hate Speech detection, and other studies on related concepts can give us insight into which method to use in this work.

2.6 Machine Learning

Machine learning is a field of computer science that enables programs to learn from experience. It is an application of Artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning allows a computer to access data and use it to learn for itself. The progress of learning begins with data such as an instruction to look for patterns in data and make decisions in the future based on the examples that we provide. The main objective of Machine learning is to allow computers to learn automatically without human intervention or assistance. Machine learning approaches can be categorized into supervised, semi-supervised, and Reinforcement approaches.

Supervised learning;-This approach is domain-dependent since it relies on manual labeling of a large volume of text [49]. In which learning can be used when the given dataset is pre-labeled. Labeling tasks is time and effort consuming but it is more efficient for domain-dependent events. During the training phase, the algorithm makes predictions about each data point's label by looking at the data and correct itself by looking at the label. The training phase ends when the algorithm achieves an acceptable level of performance. Most of the approaches used for Hate Speech detection tasks are supervised methods. Several supervised classifiers have been used to detect Hate Speech on Twitter, their results showed that all classifiers have performed the same but the different settings of features changed the accuracy of the model. Consequently, the choice depends on the features that can be extracted from the corpus. Supervised learning problems can be categorized as classification and regression problems.A classification problem is when the resulting variable is a category. A regression problem is when the output variable is a real value.

Unsupervised learning; - It is a domain-independent approach and is capable to handle a diversity of content while maintaining scalability [32]. The goal is to find regularities and patterns in the input data. Unsupervised learning problems can be categorized as association and clustering problems. An association problem is where you want to discover rules that describe large portions of data, such as people that buy X also tend to buy Y.

A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior. It does not rely on human labor to label a large volume training set; instead, it dynamically extracts domain-related key terms. The best results from their model were obtained when they incorporated semantic Hate and theme-based features.ML algorithms and pre-processing methods can be used for the classification of social media content [50].

Reinforcement Learning: -This category is based on the principle of learning by positive and/or negative feedback. It aims to train a model to find the best chain of decisions (policy), solving a specific problem. Beneficial decisions are rewarded whereas disadvantageous decisions are punished. Besides those three categories, many popular algorithms use semi-supervised learning, which as the name suggests is a mix of supervised and unsupervised learning. Here both labeled and unlabeled instances are used to train an ML model. In the case of the Machine approach, the most effective surface features in Hate Speech classification are the bag of words, words, and character n-grams features. However, in the case of classifiers, we found that the most common algorithm used is SVM, Random Forests, Decision Trees, Logistic regression, and Naïve Bayes.

2.7 Feature extraction techniques to Detecting Hate Speech

The study of Hate Speech detection has been growing only in the few last years. However, some studies have already been conducted in a few languages. Many researchers are trying to come up with different techniques to monitor Hate Speech propagation on social media. Some researcher is focused on specific Hate Speech detection techniques others propose general Hate Speech detection. Therefore, the authors allocate this general section to describe the features already employed in previous works dividing them into two categories. Those are general features used in text mining and the user or specific Hate Speech detection features.

2.7.1 General features

Textual features are the ones extracted from the text itself. They can be divided into different groups, according to the area they belong to (e.g. sentimental, semantic). On the other hand, user features represent the ones that target the user directly and the characteristics associated. Below are described the main feature extraction approaches used in text classification and Hate Speech detection.

2.7.1.1 Dictionaries based

The work categorizes the features as the features commonly used in text mining, which are dictionaries. This approach consists of making a list of words that are searched and counted in the text. In the case of Hate Speech detection, this has been conducted using content words such as insult and swear words, reaction words, and personal pronouns, several disrespectful words in the text [51]. With a dictionary that consists of words for the English language including acronyms and abbreviations [52], label specific features, which consisted in using frequently, used forms of verbal abuse as well as widely used stereotypical words.

2.7.1.2 Bag-of-words

Another model similar to dictionaries is the use of bag-of-words [53]. It can be considered as a word co-occurrence feature. The first feature included in this work is Bag of Words (vector space model) which is already mentioned as a representation model where each text is decomposed to its words, without keeping any information on the text's grammar or syntax. A vectorization process is performed on tokenized words in the corpus by assigning weight for each word according to its frequency in the document. The vectorization process is done using some statistical models (e.g. TF-IDF weight). In this case, a corpus is created based on the words that are in the training data, instead of a pre-defined set of words, as in the dictionaries. The disadvantage of this kind of approach is that the word sequence is ignored, and it is syntactic and semantic content. Therefore, it can lead to misclassification if the words are used in different contexts since it works in one straightforward way to create a bag of words features. Below figure show the Bow technique with two document example, the listing also the relevant stop words found in the two texts.

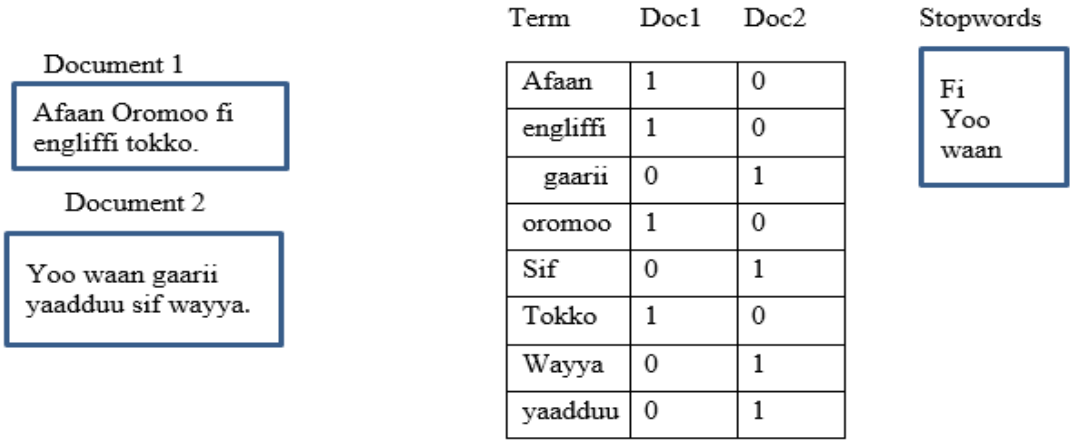

Figure 2.1 Bag of Words example

2.7.1.3 N-gram

To overcome bag-of-word limitation n-grams were implemented. N-grams are one of the most used techniques in Hate Speech automatic detection and related tasks [54], [55]. N-gram is a word prediction model using probabilistic methods to predict the next word after observing N-1 words. The most common N-grams approach consists of combining sequential words into lists with size N, where N is number of word used during probability sequences. This study uses a word N-gram method to create N-gram of post and comment features. When the number of N increases, the model performance remains the same.

2.7.1.4 TF-IDF

TF-IDF was also used in this kind of classification problem. It is a measure of the importance of a word in a document within a corpus and increases in proportion to the number of times that word appears in the document. However, it is distinct from a bag of words, or n-gram, because the frequency of the term is

off-settled by the frequency of the word in the corpus, which compensates the fact, that some words appear more frequently in general [56].

2.7.1.5 Part-of-Speech

POS approaches also make it possible to improve the importance of the context and detect the role of the word in the context of a sentence. These approaches consist in detecting the category of the word, for instance, personal pronoun (PRP), Verb non-third person singular present form (VBP), Adjectives (JJ), Determiners (DT), Verb base forms (VB). Part of Speech has also been used in Hate Speech detection problems even though proved to confuse the class identification [13].

2.7.1.6 Word Embedding

The emergence of word embedding mitigated the data sparsity problem by bringing up an extra semantic feature by generating distributed representations that introduce dependence between words. Word2Vec is one of the techniques to construct word embedding. According to Lille berg et al. [57], Word2vec has given a lot of interest by researchers in the text-mining field and it is compatible with both supervised and unsupervised Machine learning models. There is a set of different techniques to embed words as described below.

2.7.1.7 Word2vec

The granularity of the embedding is word wise, generating a vector for each word of the corpus.
Word2vec is a network of artificial neurons of two layers that can learn how to represent each word with a real number vector with its semantic features, thus it allows grouping similar words into a single vector. It is a type of mapping that allows words with similar meanings to have similar vector representations. The idea behind Word2vec is rather simple: use the surrounding words to represent the target words with a neural network whose hidden layer encodes the word representation. Word2vec, like doc2vec, belongs to the text preprocessing phase. The Word2Vec approach uses continuous bag-of-words (CBOW), and the Skip-gram model to create a high dimension vector for each word this is part of a wider concept in Machine learning: the feature vectors.

2.7.1.8 Glove

This embedding model is quite frequent in the literature of Hate Speech detection in text. It is an unsupervised learning algorithm for obtaining vector representation of words. Several corpora are provided to pre-train the models

2.7.1.9 Fast Text

This embedding is essentially an extension of Word2Vec, except each word is composed of character n-grams. The vector for each word is the sum of its character n-grams. Fast Text is one of the most important tools of word embedding that is used to represent words and sub-words of a document in the numerical form of the vector by considering the morphology of words. Facebook artificial intelligence research lab released a novel technique to solve this issue by introducing a new word embedding method called Fast Text. Each word is represented as a bag of character n-gram. In contrast, Fast Text creates a vector for each character n-gram.

Therefore, even an out-of-vocabulary word is assigned a vector based on its sub-word units. This is even more important for inflected languages since some inflected forms of words are rare and may not even appear in the training data.

2.7.1.10 Bert

BERT is a very recent word embedding that presented quite appealing results [58]. The innovation behind BERT is that, unlike Word2Vec, Glove, each word is not restricted to a single vector, depending on their context, and meaning, the same word may have different meanings. Introducing the first deeply bidirectional unsupervised language representation; this embedding might be a breakthrough in natural language processing tasks. The usage of word embeddings might be prejudicial at some point. Although most existent word embedding is trained on huge data collections.

2.7.2 Specific Features for Hate Speech Detection

As mentioned before, user-related features are an under-explored area when it comes to text classification. Most of the approaches focus on text mining and processing, ignoring user representation. User history More often than not, little information, especially due to their small length. This often results in classification errors. User profile can be generated for each author, which might help classify single messages. Three parameters are created which evaluate users' tendency towards certain behaviors: racism, sexism, and none. Such features are computed taking into account the history of each user and the class, they belong.

2.7.2.1 Objectivity and Subjectivity of the Language

In a study on [59], the authors argue that Hate Speech is related to communication that is more subjective. According to the author's explanation, a rule-based approach is used to separate the objective from the subjective sentence and remove the objective sentence from the analysis.

2.7.2.2 Account characteristics

Facebook or Twitter provides a set of characteristics related to the accounts themselves, such as the presence of a profile image, location, and time zone. Although these are not directly related to the user, they may unveil certain behaviors.

2.7.2.3 Augmentation

To suppress noise in the target comments, a set of similar comments, posted by other users is gathered in these were selected from a large unlabeled corpus using Locality Sensitive Hashing, a hasher able to reduce the dimensionality of the data and, consequently, decrease the search space.

2.8 Feature Selection in Text Classification

Feature Selection is a process of reducing the number of input variables when developing a predictive model. It is desirable to reduce the number of input variables to both reduce the computation cost of modeling and to improve the performance of the model. Feature selection is primarily focused on removing non-informative or redundant predictors from the model to improve the time needed to train a model. This involved selecting a

subset of relevant features that would help in identifying Hate and no Hate posts. Feature selection can be defined as the process of selecting a subclass of the terms occurring in a corpus and using only this subset as features in text classification. The most common feature selection techniques are chi-square, mutual information, and frequency based feature selection.

2.8.1 Frequency-based Feature Selection

This approach is to choose the terms that occur most frequently in a class. Frequency can be defined as document or collection frequency. Collection frequency refers to the number of tokens of a term t that occur in documents in class c whereas document frequency is the number of documents in class c that contain term t.

2.8.2 Chi-square Feature Selection

In feature selection for text classification, the two events are occurrence of a term and occurrence of a class. The terms are then ranked according to Equation (2) below

$$X^2(\mathbb{D}, t, c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}} \qquad \textit{Equation(1)}$$

Where: et expresses whether the document contains term t or not

ec expresses whether the document belongs to class c or not

N represents the observed frequency in document D

E represents the expected frequency

$\chi 2$ is a measure of how much the expected counts E and observed counts N deviate from each other.

2.8.3 Mutual Information

This feature selection approach computes a (t, c) as the expected mutual information of term t and class c. Mutual Information (MI) measures how much information the presence or absence of a term contributes to making the correct classification decision on c.

$$I(U;C) = \sum_{e_t \in \{1,0\}} \sum_{e_c \in \{1,0\}} P(U = e_t, C = e_c) \log_2 \frac{P(U = e_t, C = e_c)}{P(U = e_t) P(C = e_c)}, \qquad \text{Equation(2)}$$

Where: U is a random variable that takes values et=1 (the document contains term t) and et=0 (the document does not contain t), C is a random variable that takes values ec=1 (the document is in class c) and ec=0 (the document is not in class c).

2.9 Feature Weighting

Various weighting schemes could be then used for a document-term matrix. The simplest weighting scheme being a simple Boolean 1 if the term appears in the document or 0 if it does not. It could also be based on the

frequency of the term in the corpus or term frequency (number of times the term appears in the specific document). If C is the set of all classes, then TF (t, c) can be defined as the frequency of term t in class c, calculated as in Equation 4.)

$$TF(t,c) = \frac{|occurrences\ of\ t\ in\ c|}{|terms\ in\ c|}$$ Equation (3)

2.10 Algorithms Commonly Used for Hate Speech Detection Machine learning

Consulting different sources on algorithms of Hate Speech detection is the focus of this section because authors aim to work on this specific topic. The most common approach found in a Machine learning model for Hate Speech classifications is, SVM, Random Forests, Decision Trees, logistic regression, Naïve Bayes, and Deep learning respectively on the use of frequency by authors. The data classification is based on general Hate Speech, racism, sexism, religion, anti-Semitism, nationality, politics, and socio-economic status respectively on the categorization use of frequency.

2.10.1 Logistic Regression

Logistic regression (LR), or binary logistic regression, is a simple classification algorithm that uses statistics to make predictions where the outcome is binary. The goal of LR is to find the best fitting model that describes the relationship between the outcome and a set of independent variables. The LR model estimates the confidence of an outcome based on the independent variables by using a non-linear function called the logistic function. The logistic function, also known as the sigmoid function, can take any real-valued input and return output in the interval [0,1], which can then be interpreted as a probability. The outcome represents the model's confidence in the classification, where values close to 1 indicate the first class and values closer to 0 indicate the other class.

$S(x) = \frac{1}{1+e^{-x}}$ Equation(4)

2.10.2 Support Vector Machines

The standard definition of Support Vector Machines (SVM) was proposed in 1993 and published in 1995 by Vapnik and Corinna Cortes [60]. A supervised Machine-learning algorithm can be used for both regression and classification purposes [61]. The main idea of the algorithm is to find a hyperplane that divides the dataset into two distinct classes. Often the hyperplane is not easy to find, data points rarely line up perfectly, and they are often shuffled together in a linearly non-separable order. Training data instances are represented as coordinates in an n-dimensional space, where n equals the number of features. The hyperplane is a subspace with n-1 dimension, i.e. one dimension less than the number of features. Viewing input data as two sets of vectors in an n-dimensional space, an SVM will construct separating hyperplanes in that space, one that maximizes the margin between the two data sets. To calculate the margin, two parallel hyperplanes are constructed, one on each side of the separating hyperplane, which is "pushed up against" the two data sets. Intuitively, the

hyperplane has the largest distance to the neighboring data points of both classes, since in general the larger the margin the better the generalization error of the classifier achieves a good separation. SVMs are universal learners. In their basic form, SVMs learn linear threshold functions. Nevertheless, by a simple "plug-in" of an appropriate kernel function, they can be used to learn polynomial classifiers, radial basic function (RBF) networks, and three-layer sigmoid neural nets. One remarkable property of SVMs is that their ability to learn can be independent of the dimensionality of the feature space. SVMs measure the complexity of hypotheses based on the margin with which they separate the data, not the number of features. This means that we can generalize even in the tile presence of very many features if our data is separable with a wide margin using functions from the hypothesis space. In the case of support vector Machines, a data point is viewed as a p-dimensional vector, where p is the number of attributes.

In this example, as a 2-dimensional vector, and we want to know whether we can separate such points with a $(p-1)$ -dimensional hyperplane (in this example since p is taken to be 2, a 1-dimensional hyperplane is a straight line). This is called a linear classifier. There are many hyperplanes (lines in this case for 2 dimensions) that might classify the data. However, we are additionally interested in finding out if we can achieve maximum separation (margin) between the two classes. By this, we mean that we pick the hyperplane so that the distance from the hyperplane to the nearest data point is maximized. That is to say that the nearest distance between a point in one separated hyperplane and a point in the other separated hyperplane is maximized. Now, if such a hyperplane exists, it is clearly of interest and is known as the maximum-margin hyperplane and such a linear classifier is known as a maximum margin classifier.

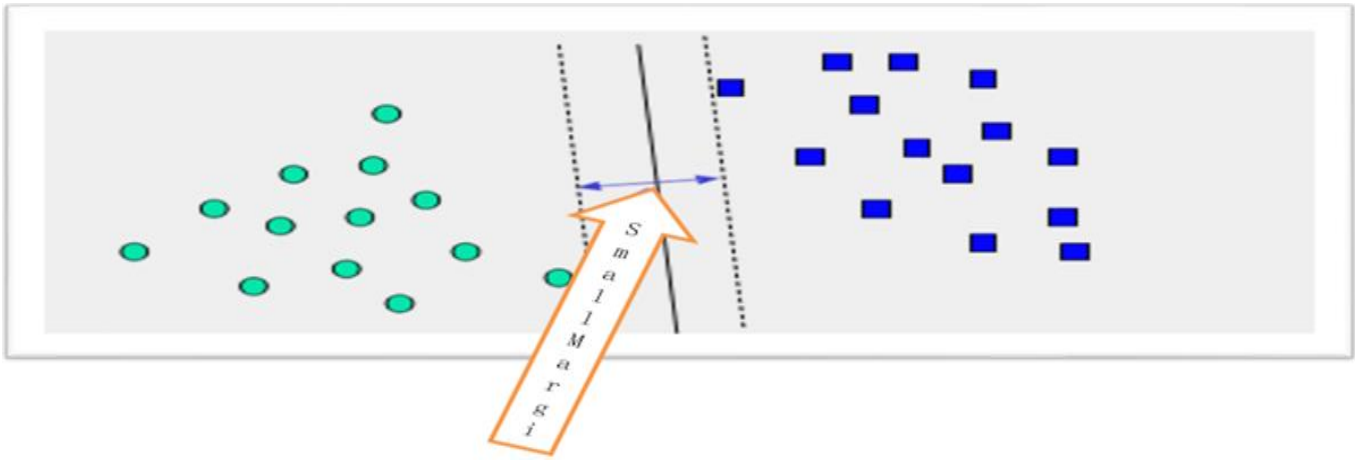

Figure 2.2 SVM Small Margin

Dashed lines are drawn parallel to the separating line mark the distance between the dividing line and the closest vectors to the line. The distance between the dashed lines is called the margin. The vectors (points) that constrain the width of the margin are the support vectors.

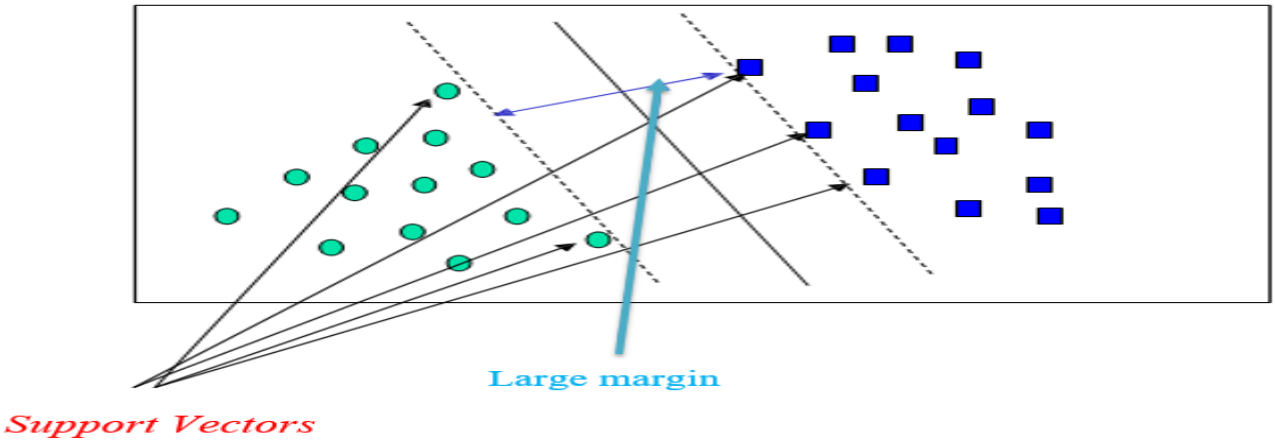

Figure 2.3 SVM Large Margin

In the above figure, the cases with one category are in the lower left corner (oval) and the cases with the other category are in the upper right corner (squares); the cases are completely separated. The SVM analysis attempts to find a 1-dimensional hyperplane (i.e. a line) that separates the cases based on their target categories. If all analyses consisted of two-category target variables with two predictor variables, and a straight line could divide the cluster of points, life would be easy. Unfortunately, this is not generally the case, so SVM must deal with More than two predictor variables.

Separating the points with non-linear curves.

- Handling the cases where clusters cannot be completely separated.
- Handling classifications with more than two categories.

If we add a third predictor variable, then we can use its value for a third dimension and plot the points in a 3-dimensional cube. A 1-dimensional line as seen above can separate points on a 2-dimensional plane. Similarly, a 2-dimensional plane can separate points in a 3-dimensional cube. As we add additional predictor variables (attributes), the data points can be represented in N-dimensional space, and an (N-1)-dimensional hyperplane can separate them.

2.10.3 Naive-Bayes classifier

Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given sample belongs to a particular class [62]. The Naive Bayes algorithm (NB) is a routine based on Bayes Theorem. Bayes Theorem states: For a given data sample X whose class label unknown, let H be some hypothesis such that the data sample X belongs to a specified class C, for the classification problem, we want to determine P (H | X), the probability that the hypothesis H holds given the observed data sample X.

$$P(H/X) = \frac{P(X/H)*P(H)}{P(X)}$$

Equation(5)

Naïve Bayesian (NB) classifier assumes that the effect of an attribute value of a given class is independent of the values of the other attributes. Naïve Bayesian algorithm has been widely used for Hate Speech detection and shown to produce a very good performance. An algorithm ignores all possible dependencies and

correlations among inputs and considers every classified feature independent of any other feature. This model gives class labels to problem instances, represented as vectors of feature values. The value of a particular feature is autonomous of the value of other features. Prior probability is known and posterior probability is checked in the naive Bayesian technique.

The naive part of the naïve Bayesian algorithm is the assumption of word independence that the conditional probability of a word given a category is assumed to be independent of the conditional probabilities of other words given that category. There are two generative models of the Naïve Bayesian algorithm. One is the multi-variate Bernoulli event model that only takes into account the presence or absence of a particular term, so it does not capture the number of occurrences of each word. Bayesian belief networks are graphical models, which unlike naïve Bayesian classifiers allow the representation of dependencies among a subset of attributes. Among the advantages of the Bayesian classifiers is that simple technique results in high accuracy, especially when combined with other methods.

The limitations when using naïve Bayes is the independent and equally important assumption which may cause skewed results, especially if any of the variables are interrelated, as that relationship will have a greater effect on the decision, for better or for worse. Naïve Bayes classification does not allow for categorical output attributes.

2.10.4 Decision Tree

A decision tree is a predictive model that is a mapping from observations about an item to conclusions about its target value. More descriptive names for such tree models are classification tree, which provides discrete outcome, or regression tree, which provides continuous outcome. Decision tree shapes classification simulations in the usage of a tree structure. The aim is to construct a model that predicts the value of a target variable based on input variables. In these tree structures, leaves characterize class labels and branches denote features of class labels. It disturbances a dataset into smaller subsets while at the identical time a decision tree is incrementally established. The ultimate result is a tree with decision nodes and leaf nodes. Decision trees can knob both categorical and numerical data. In order to classify an unknown sample, the attribute values of the sample are tested against the decision tree. Testing begins from the root, decisions are made at the branches and class prediction arrives at the leaf node. Decision tree classifiers can handle high dimensional data and have good accuracy. A path is traced from the root to a leaf node that holds the class prediction for that sample. Consequently, decision trees can easily be converted to classification rules. The tree starts as a single node representing the training sample. If the sample is all of the same class, then the node becomes a leaf and is labeled with that class, otherwise, the algorithm uses an entropy-based measure known as information gain as a heuristic for selecting the attribute that will best separate the samples into individual classes. This attribute becomes the “test” or “decision” attribute at the node. A branch is created for each known value of the test attribute, and the samples are partitioned accordingly. The algorithm uses the same process recursively to form a decision tree for the samples at each partition. Once an attribute has occurred at a node, it need not be considered in any of the node’s descendants.

The recursive partitioning stops only when any one of the following conditions is true.

- All samples for a given node belong to the same class.
- There are no remaining attributes on which the samples may be further partitioned.
- There are no more samples for the branch.

2.10.5 K-nearest Neighbor Classifiers

The k-nearest-neighbor method was first described in the early 1950s. The method is labor intensive when given large training sets, and did not gain popularity until the 1960s; increased computing power becomes available [Han and kamber, 2006]. KNN is an Instance-based learner. Instance-based learners are also referred to as lazy learners; because they wait until the test, the set is supplied, unlike eager learners.

Eager learners, when given a set of training tuples, will construct a generalization (i.e. classification) model before receiving a new (e.g. test) tuples to classify. We can think of the learned model as being ready and eager to classify previously unseen tuples.

Lazy learners wait until the last minute before doing any model construction in order to classify a given test tuple. That is, when given a training tuple, a lazy learner simply stores it (or does only a little minor processing) and waits until it is given a test tuple. Only when it sees the test tuple does it perform generalization in order to classify the tuples based on its similarity to the stored training tuples [63]. KNN classifier is a learning algorithm that is based on a distance function for pairs of observations, such as the Euclidean distance. In KNN, "Closeness" is defined in terms of a distance metric used.

The Euclidean distance between two points or tuples, say, In other words, for each numeric attribute, we take the difference between the corresponding values of that attribute in tuple X1 and in tuple X2, square this difference, and accumulate it. The square root is taken of the total accumulated distance count. For K-nearest classification, the unknown tuple is assigned the most common class among its k-nearest neighbors. When k = 1, the unknown tuple is assigned the class of the training tuple that is close to it in pattern space. In this classification paradigm, k nearest neighbors of training data are computed first. Then the similarities of one sample from testing data to the k nearest neighbors are aggregated according to the class of the neighbors, and the testing sample is assigned to the most similar class. One of the advantages of the KNN is that it is well suited for multi-modal classes as its classification decision is based on a small neighborhood of similar objects (i.e., the major class). So, even if the target class is multi-modal (i.e., consists of objects whose independent variables have different characteristics for different subsets), it can still lead to good accuracy. A major drawback of the similarity measure used in KNN is that it uses all features equally in computing similarities.

2.11 Deep learning approach

Deep learning is a subset of Machine learning approaches that attempts to learn the layered model of inputs. It enables computational models composed of different processing layers to learn representations of data with multiple abstraction levels. Deep learning methods use neural networks to automatically learn the multi-layers of features from the given data. In the early 2000s, improvements in computer hardware and advances in

optimization and training techniques made it possible to train even larger and deeper networks, which leads to the modern deep learning methods [64]. Make use of neural networks to automatically learn multi-layers of features from the given data. By the early 2000s, improvements in computer hardware and advances in optimization and training techniques made it possible to train even larger and deeper networks, leading to the modern term deep learning [65].Machine-learning approaches that use linear models and are trained over high dimensional yet very sparse feature vectors. However, lately, non-linear neural-network over dense inputs has been showing success. The most popular used networks are Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), typically Long Short-Term Memory network (LSTM). In the literature, CNN is well known as an effective network to act as 'feature extractors', whereas RNN is good for modeling orderly sequence learning problems [66].

2.12 Performance metrics

A wide range of performance metrics is used to evaluate Machine learning algorithms and models. These measures are originally built from a confusion matrix that, despite not being a performance measure by itself, serves as the basis for several other methods.

Accuracy

Accuracy is a generic performance measure that assesses the overall effectiveness of the algorithm, by computing the number of correct predictions over all the predictions made. Although it is commonly used, accuracy does not distinguish between different classes. Consequently, this performance metric may be misleading, especially when the classes of the data are unbalanced [67].

Accuracy = tp+tn/tp+fp+tn the balance between precision and recall Equation(6)

Recall

Recall also known as Sensitivity or True Positive Rate, is defined as the proportion of real positives that are correctly predicted as positive

Recall = $\frac{tp}{tp+fn}$ $High\ FN\ cost$ Equation(7)

Precision

Precision denotes the proportion of predicted positive cases that are positive.

It is a measure that tells us what proportion of data that we classify as being in a specific class, actually had in that class. The predicted positives (data is predicted, as specific classes are TP and FP) and the data's being in a class is TP. Thus, precision is the ratio between the TP with the sum of TP and FP.

Precision = tp/tp+fp high fp cost Equation(8)

F1 Score

The most used performance metric in Hate Speech detection in the text is the F1 score. It is defined as the harmonic mean of Precision and Recall, and considers class imbalance, unlike accuracy [68]. Hence, it is wide usage in Hate Speech detection. Therefore, it is best if we can get a single score that rather represents both Precision (P) and Recall(R).it is also determined simply by taking the arithmetic mean of precision and recall.

F1 score = $\frac{2*precision*recall}{Precision + recall}$ Equation(9)

The above equation shows the balance between precision and recall for unbalanced class

	Predicted No	**Predicted Yes**
Actual No	TN	FP
Actual Yes	FN	TP

Figure 2.4 Confusion Matrix

Chapter:3 Related Work.

This chapter presents a review of the existing research in the field of Hate Speech detection, as well as the challenges faced in the field. Whereas our work was focused on Afaan Oromoo Hate Speech detection and related, work considers the broader field of Hate Speech, offensive and abusive language identification in different languages. For a clear understanding, we have seen reviews of works of literature on Ethiopian languages and foreign languages social media Hate Speech detection researches.

3.1 Previous work on Hate Speech in local language

Zewdie Mossieand Jenq-Haur Wang [33] To develop an apache spark-based model to classify social network Hate Speech detection for the Amharic language by using Naïve Bayes and Random forest algorithm. The objective of this study was developing Amharic Hate Speech detection. The author proposes employing Random forest and Naïve Bayes for Machine learning algorithm, Word2Vec, and TF-IDF for feature selection. Since the Amharic language, have not public available dataset authors have built a corpus of comments retrieved from Facebook public pages of Ethiopian newspapers, individual politicians, activist, TV and radio broadcast and groups. By doing so, authors could capture both casual conversations and politically Hated posts and comments. Then preprocess the posts and comments.

Table 3.1 Dataset for this paper

Training Dataset		
No Hate	Hate	Total
2,629	2253	4882
Test Dataset		
667	571	1238
3296	2824	6120

The above table was a dataset of six thousand one hundred and Amharic posts and comments out of this four thousand eight hundred eight two training comments as training dataset and one thousand two hundred thirty-eight, comments as testing dataset using the spark Data Frame random split function with the seed of 100. Tested by 10-fold cross validation, the model based on word2vec embedding performed best with 79.83% accuracy of 79.83 % and 65.34% for Naïve Bayes with word2vec feature vector and Random Forest with TF-IDF feature modeling approach respectively. The study achieves a promising result with Naïve Bayes with word2vec.

In a study by **Tesfaye* and Tune** [69] Researchers aim to prepare a labeled huge Amharic dataset by collecting posts and comments from selected Facebook pages of activists that participated actively. The authors propose to use recurrent neural network models for automated Hate Speech posts detection from Amharic posts on Facebook is developed by using Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) with word n-grams for feature extraction and word2vec to represent each unique word by vector representation. The experiment was conducted on those two models by using 80% data for training and 10% for validation. The

remaining 10% of the dataset was used for test purposes. At the last LSTM based RNN of batch size 128 and learning rate 0.001 with RMSProp optimizer and 0.5 dropout archives an accuracy of 97.9% to detect post as Hate and free by training with 100 epochs.

A study by **Zemedkun** [70] stated that in the Ethiopian case, in recent years the dissemination of online Hate Speech has either becomes destructive or shows its potential to be so by fomenting among other things, intolerance, mistrust, and discrimination between the members of a different society. As a result, many people have lost their lives and many others are displaced from their houses. In the meantime, the government's measure so far is mainly limited to using illegitimate and uneconomical means of restricting the right by shutting down the accessibility of social media in the country. Even though, recently set out prosecution of the perpetrator of the Hate Speech in the Ethiopian criminal justice system. However, as they stated the Absence of a legal framework criminalizing Hate Speech, the anonymity of the actors is the main challenge to prosecute individuals for Hate Speech.

A study by **Mossie & Wang** [71] proposes a Hate Speech detection approach that identifies hatred against defenseless minority groups on social media. They used Gated Recurrent Unit (GRU), and LSTM (long short-term memory) through n-gram TF-IDF and word Embedding feature extraction techniques for GBT, RF, and GRU and LSTM respectively. They used 12,839 (90%) posts for training and 1426 (10%) posts for testing. They get the best performance through RNN-GRU with an accuracy of 0.9256 and AUC of 0.9785. Using word embedding as a feature to classify Amharic texts into Hate and neutral. Finally, they used the Hate lexicon to predict the potential target ethnic group for hatred, and they verified the Tigre ethnic group as the highly vulnerable community. However, Mossie (2019) [14] has published while this work was in progress but for this research, by examining different datasets with four categories, different deep learning algorithms and data augmentation techniques performed.

A study by **Megersa Oljira** *[72]* focuses on sentiment analysis of Afaan Oromo social media content because automatically identifying and classifying opinions from social media posts. By employing Multinomial Naïve Bayes Machine learning algorithm and different n-grams such as unigram, bigram, trigram and their combinations as features. The Authors propose to use the Naive Bayes algorithm for the classification of Afaan Oromoo sentiment using n-gram techniques and use precision, recall, f-measure and accuracy to evaluate performance. The proposed MNB approaches achieved According to the experiment, the result shows that accuracy of 90.7%, 71.1%, 54.6%, 92.7%, 92.4%, and 75% for unigram, bigram, trigram, unigram-bigram, unigram-trigram, and bigram-trigram respectively.

Table 3.2 summary of Hate Speech detection for local language

Title and Authors	Feature extraction method	Algorithms Used	Selected model	datasets	Results
Social Network Hate Speech Detection for Amharic Language, Zewdie Mossie, et al. 2018	Word2Vec and TF-IDF	NB, RF	Naive Bayesian	6,120	79.83
Vulnerable community identification using Hate Speech detection on social media, Zewdie Mossie, et al. 2019.	n-gram TF-IDF and word2vec embedding	GBT, RF, GRU and LSTM	RNN-GRU	14,319	92.5
Automated Amharic Hate Speech Posts and Comments Detection Model,Surafel Getachew Tesfaye and Kula Kekeba Tune	word n-grams and word2vec	LSTM and GRU	RNN	5000	97.9%
Sentiment Analysis of Afaan Oromo using Machine learning Approach , Megersa Oljira	n-gram	Naïve Bayes	Multinomial Naïve Bayes	1452	92.7%

3.2 Previous work on Hate Speech for foreign language

A study by **Marian, Tianyu W, Gabriela F, Hanna** [73] researchers used a Logistic Regression model with character level features to classify comments short messages from Twitter, a major social media platform, by applying standard lexical features and a linear SVM classifier to establish a baseline for Hate Speech detection and system use character n-grams, word n-grams and word skip-grams. The system is Models based on Machine learning and natural language processing provide a way to detect Hate Speech in web text. By using, the idea of utilizing features weights or otherwise defined knowledge acquired for one task to solve another related problem. The authors use Waseem data set is publicly available and it consists of 15,216 instances from Twitter that were annotated as Racist, Sexist, or Harmless, and also Davidson dataset is also publicly available. It consists of 22,304 instances from Twitter annotated as Hate, Offensive, and Harmless. Out of this dataset 90% used for training and 10% for tests. The Authors obtain results of 78% accuracy in identifying posts across three classes. This method enables generating an interpretable 2D text visualization called the Map of Hate that is capable of separating different types of Hate Speech and explaining what makes text harmful. These methods and insights hold the potential for not only safer social media but also reduced need to expose human moderators and annotators to distressing online messaging. The study shows that methods are the keys for analyzing social media contents at scale to make these web platforms safer and understand the genre of Hate Speech and its sub-genres better.

A study by **Zeerak Waseem and Dirk Hovyanalyze** researchers focus on the influence of several extra-linguistic features in conjunction with character *n*-grams for Hate Speech Detection. In addition, use a dictionary to detect Hate Speech on the social median, which is in form of racist and sexist remarks. Developing a taxonomy and Machine learning models for identifying and classifying Hate in online news media have been made with Machine learning, including Logistic Regression, Decision Tree, Random Forest, Adboost, and Linear SVM, to generate a multiclass, multilevel classification model that automatically detects and categorizes Hateful comments in the context of online news media. Various Machine learning approaches have been made in order to tackle the problem of toxic language. The majority of the approaches deal with feature extraction from the text. Lexical features such as dictionaries [74] and bag-of-words [75] were used in some studies.

A study by **Marco Polignano and Pierpaolo Basile** researchers proposes Ensemble Learning and Deep Neural Networks by using three classification strategies vote algorithm including Support Vector Machine Random Forest, Deep Multilayer Preceptor. The author applies a 5-fold subdivision of the training Sets and formalizes messages as a concatenation of word2vec sentence vectors and a TF-IDF bag of words.

A study by **Han Liu and Pete Burnap** [76] researchers propose fuzzy multi-task learning for Hate Speech identification by using a different dataset. Religion - 1,901 comments, with 222 instances of offensive or antagonistic content (11.68% of the annotated sample). Race - 1,876 comments, with 70 instances of offensive or antagonistic content (3.73% of the annotated sample). Disability - 1,914 comments, with 51 instances of offensive or antagonistic content (2.66%of the annotated sample) and Sexual Orientation - 1,803 comments, with 183 instances of offensive or antagonistic content (10.15% of the annotated sample). The study results show that the proposed fuzzy approach out performs the state-of-the-art probabilistic approaches such as SVM and DNNs on embedding features.

In the case of Ethiopia, the use of Hateful words to bring about hatred against a group of people based on their ethnicity, political attitude, religion and socio-economic are prevailing [77]. The work categorizes the features as the features commonly used in text mining which are dictionaries and lexicons. This approach consists of making a list of words that are searched and counted in the text. In the case of Hate Speech detection, this has been conducted using content words such as insult and swear words, reaction words, and personal pronouns [78]. Label specific features which consisted in using frequently used forms of verbal abuse as well as widely used stereotypical words [79].

Sreelakshmi k, Premjith B,Soman K.P. [80]. This study was done by using the SVM and RBF classifier with word2vec and doc2vec features. Researchers collected around 10000 data samples from Twitter and HASOC with 5000 Hate Speech and 5000 Neutral Speech. Then data is labeled into two classes Non-Offensive and Hate after label data was preprocessed and features like punctuation count, emoticon count, word n-gram, character n-gram, word2vec of lexico words were extracted. As the researcher states detection of Hind English code-mixed is very complex because of linguistic complexity and non-standard variations in language structure. This paper contributes a Machine learning model which does a binary Classification of Hindi-English comments to Hate and no-Hate. Fast Text has pertained model and bilingual embedding were used as features for the

classification. One of the purposes of fast text in this paper is text representation and text classification. The whole dataset was taken which formed a feature matrix of size 10000x300. Then evaluated by 10-fold Cross-validation and evaluation parameters for this experiment. Experiment one conduct by SVM-linear, SVM-RBF, Random Forest but Random Forest by using doc2ve feature for classification by Appling two types of training method CBOW and skip-gram have high Accuracy=0.6415, Precision=0.6415, Recall=0.6415, F1-Score=0.6414.

In experiment two use, word2vec feature with SVM_RBF has high score than the other which is Accuracy=0.7511, Precision=0.7517, Recall=0.7511, F1-score=0. 7509. For this experiment, hyperparameters are fixed such as the vector length 300 and the window length as 5.

Experiment three Use Fast Text considers character n-grams as the smallest unit achieve Accuracy=0.8581, Precision=0.8586 Recall=0.8581 F1-Score=0. 8580. These features created a fat matrix. So they used a dimensionality reduction technique such as chi square to reduce the size of the feature extracted SVM-Random forest was used for classification which gave accuracies of 85.81%.

In a study by **T. Davidson, D. Warmsley** [81], a similar application was explored. They created a new dataset using the Twitter API for twitter data classification as Hate Speech, offensive language, or neither. In their dataset, they collected a set of 85.4 million twitter samples from about 33 thousand Twitter users. From there, they built a set of 24k labeled twitter samples. Features, such as bigram, unigram, and trigram were weighted by their TF-IDF and were used for the classification task. Other features included binary and count indicators for hashtags, mentions, recommends, and URLs. They tested a large number of classifiers: logistic regression, Naive Bayes, decision trees, random forests, and linear SVMs. Through their experiments, it was found that Logistic Regression and Linear SVM tended to perform better results. The best model obtained an overall precision of 0.91, recall of 0.90, and F1-score of 0.9. However, the classifier did not present good results to detect Hate Speech, for this class, the precision and recall were 0.44 and 0.61, respectively.

In a **Research conducted by NDjuric et al.**, [82] concentrates on high-dimensionality and Sparsity issues that influence the current state-of-the-art detection systems. They proposed a two step Method for detecting Hate Speech. First, they used the paragraph2vec for joint modeling of comments and words, along with the continuous BOW (CBOW) neural language model. Thus, a low-dimensional text embedding is obtained, which is then used to train a binary classifier on Yahoo! Finance website comments dataset.

In a Study by **F.D. Vigna et al**., [83] proposed the first Hate Speech classifier focusing on Italian texts.

They implemented a model to classify comments of public Italian Facebook pages into strong Hate, weak Hate and no Hate categories. They used two different classifiers namely, SVM and LSTM with word embedding lexicons and sentiment polarity as the features obtaining effective results. They obtained an accuracy of 64.6 and 6o.5 and Fl-score of 75.7 and 74.7 for SVM and LSTM respectively for three classes and 72.9 and 75.2 for SVM and LSTM. Authors using 3,575 documents and sentiment polarity and word embedding lexicons taken from pre-developed Italian corpora for sentiment polarity and web and comments detections.

In a study by **P. Badjatiya et al.,** [84] research conducted by researchers focused on using deep learning models

to classify comments as racist, sexist or neither. They used various comments semantic embedding's such as char n-grams, word Term Frequency-Inverse Document Frequency(TF-IDF) values, Bag of Words and task-specific embeddings learned by the Fast Text, CNNs and LSTM models to train classifiers such as Gradient Boosted Decision Trees (GBDTs), Logistic Regression, Random Forest, SVMs and DNNs. They obtained a F1 score 18 points higher than the state-of-the-art methods.

In a study by **L. Gao et al.**, [85]proposed an approach wherein they detected Hate Speech in Fox News user comments by considering the context in which the comments were made. They trained two types of models namely Logistic Regression and a Neural Network consisting of Bi-directional LSTMs to obtain results that showed an increase of 3% – 4% in F1 score as compared to the existing baseline models.

In a study by **Tan and Zhang** [21] conducted an empirical study of sentiment categorization on Chinese documents. They tested four features – mutual information, information gain, chi-square, and document frequency; and five learning algorithms: centroid classifier, nearest Neighbor, Winnow classifier, Naıve Bayes (NB) and Support Vector Machine (SVM). Their results showed that the information gain and SVM features provided the best performances for sentiment classification coupled with a domain or topic-dependent classifiers.

Table 3.3 some of Hate Speech detection summary for foreign language

Title and Authors	Feature extraction method	Algorithms Used	Selected model	datasets	Results
Developing a Taxonomy and Machine Learning Models for Identifying and Classifying Hate in Online News Media, Marian ,Tianyu ,Gabriela, Hanna	character n-grams, word n-grams and word skip-grams	linear SVM	Logistic Regression	22,304	78%
Hate me, Hate me not : Hate Speech detection on Facebook, F. Del Vigna, et al.	Word embedding lexicons	SVM and LSTM	LSTM	3575	75.2%
Detection of Hate Speech Text in Hindi-English Code-mixed Data, Sreelakshmi k, Premjith B,Soman K.P	word2vec and doc2vec	SVM and RBF	RBF	10000	85.81 %
Automated Hate Speech detection and the problem of offensive language,T. Davidson, D. Warmsley	bigram, unigram, and trigram	LR, RDF Nb, DT & linear SVMs	Linear SVM	24k	90%

3.3 Challenges of Studies on Hate Speech Detection

Study on automated Hate Speech detection, especially in social media, has become increasingly relevant during the last couple of years. With the rise of social media, it has become a subject of public attention. Companies like Facebook, Google, or Yahoo are actively participating research to provide solutions for this problem. However, just a few publicly available datasets are large enough to train ML models with. To achieve suitable

results, datasets have to be sufficiently big. Besides that, a human workforce is needed in order to provide labels for the samples. In the case of Hate Speech, it can be useful to employ trained annotators to label a dataset. The identification of Hate Speech requires a certain amount of skills, especially when it has to be distinguished from language that is just offensive. The societal impact of the Internet and social media has increased over the past years, and perhaps this is why there has also been growth and interest in research covering Hate Speech detection.

Automatic detection and classification of Hate Speech language in user-generated content are far from trivial due to many factors such as the high variety and noisiness of the data. Some of the common challenges that arise when developing automatic classification systems for these types of tasks are discussed below.

First, note the common practice in online communities to obfuscate offensive words and phrases to prevent manual and automatic intervention. This marks it difficult for simple keyword noticing algorithms to be effective since obfuscations are common on these social media platforms, making the effectiveness of simply looking for certain keywords very limited [86]

Secondly, even if a simplistic blacklist approach would be effective, it is impossible to keep track of all racial, sexist and generally Hate Speech terms as these tend to evolve quickly over time and this happens exponentially fast in online communities [87].

Thirdly, the offensiveness of a conversation often crosses sentence boundaries so the full context is required in order to correctly classify the offensiveness. A thorough understanding of the context is also required to be able to accurately identify phenomenon's such as sarcasm, which is listed in [71] as one of the major challenges when it comes to these kinds of tasks. While the amount of research increases, the field still faces several challenges, both in the actual task of detecting Hate Speech and the research area in general.

Nobata et al [88], have summarized the task of detecting Hate Speech is interpreted as offensive or Hate Speech is subjective which a results a problem for annotation to training Hate Speech detection system.

Ross et al [89], aimed to estimate the reliability of annotations and found that there is a low agreement among users when it comes to identifying hateful messages that suggest annotators are unreliable.

Chapter:4 The Proposed System Architecture.

This chapter present a methodology to design and develop a Hate Speech Detection framework. In addition, we will discusses the proposed solution of Afaan Oromoo language Hate Speech detection using Machine learning techniques suitable to solve the problem of under study. In addition, it discuss the architecture of the proposed HSD framework is presented and individual components are described. The primary goal of this research work is to design a framework that detects HSD using SVM.

4.1 Overview of the Architecture

The proposed solution based on the architecture shown in figure 4.1. It takes Afaan Oromoo datasets as input and then the preprocessed based on the language nature, which removing unnecessary characters, normalization, tokenization, and another basic preprocess is done. After all the preprocessing, then feature extraction takes place to extract features using TF-IDF, n-gram, and word2vec. The output of this task is an important feature vector of the dataset for training the model. After feature extraction, a Classifier was developed using SVM Machine learning algorithms and a training set with feature vectors data frame of the whole dataset. A Classifier is liable for creating a framework using a support vector Machine.

Finally, the framework will classify the data into Hate Speech and neutral. The models were evaluated using accuracy, precision, recall metrics. The outcome of these tasks is a model used for Hate Speech and normal detection. Finally, the detection model is evaluated and selected based on the results obtained using the model evaluation method discussed in chapter three. The final selected detection model was used to develop a prototype that can take new Afaan Oromoo texts as input and it classifies the input whether it contains Hate or normal Speech.

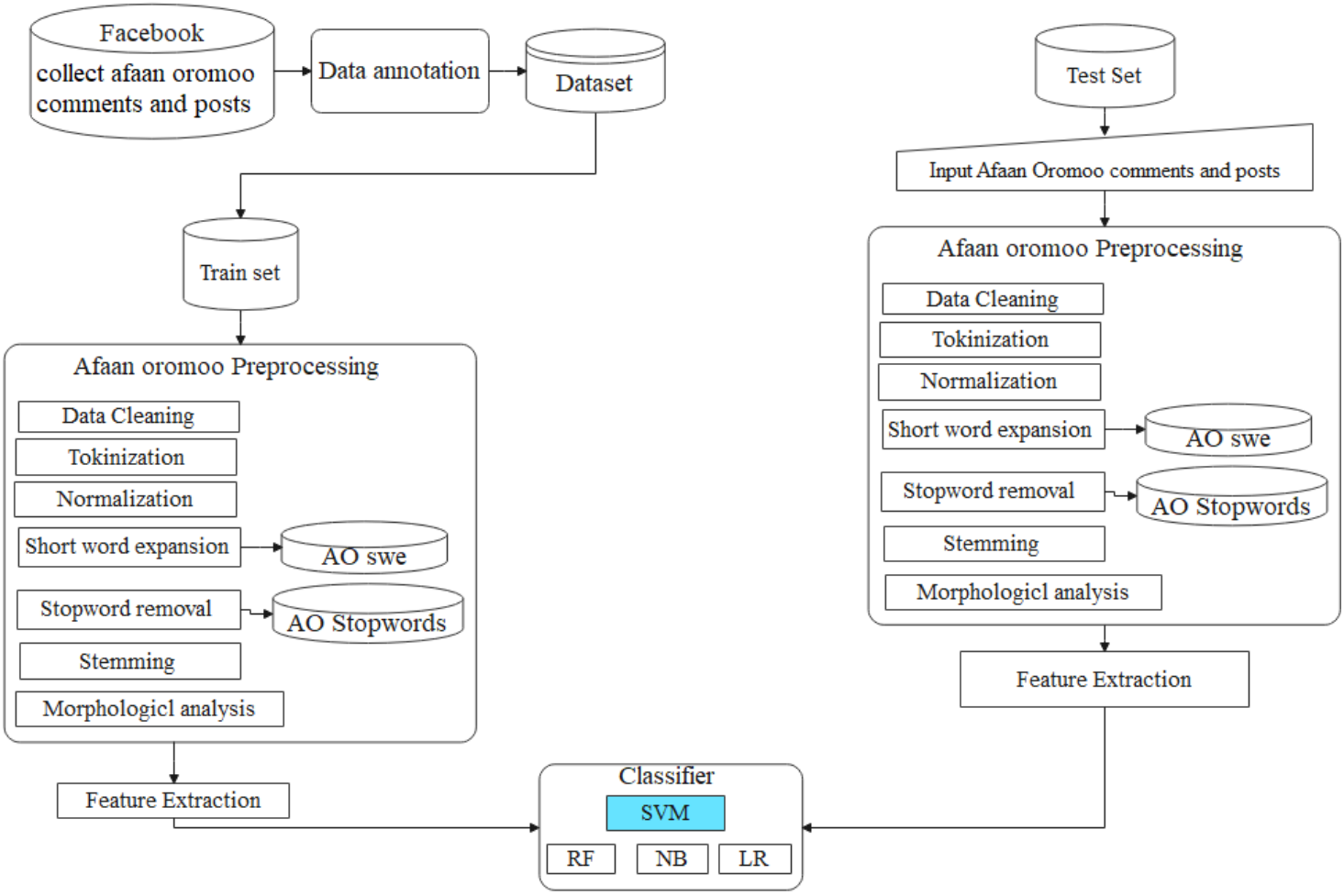

Table 4.1 The Proposed HS Detection Architecture

4.2 Dataset Construction

The objectives of this study are to detect Afaan Oromoo Hate Speech. Therefore, it needs to build a new Afaan Oromoo Hate Speech dataset. This new dataset a needed because there is no published or annotated dataset for this purpose. The process of building the dataset for Afaan Oromoo Hate Speech consists of three main steps,

1. Crawling Afaan Oromoo post and comment textual data from public Facebook pages
2. Preparing, filtering, or merging gathered data into one file dataset. And
3. Annotating the dataset.

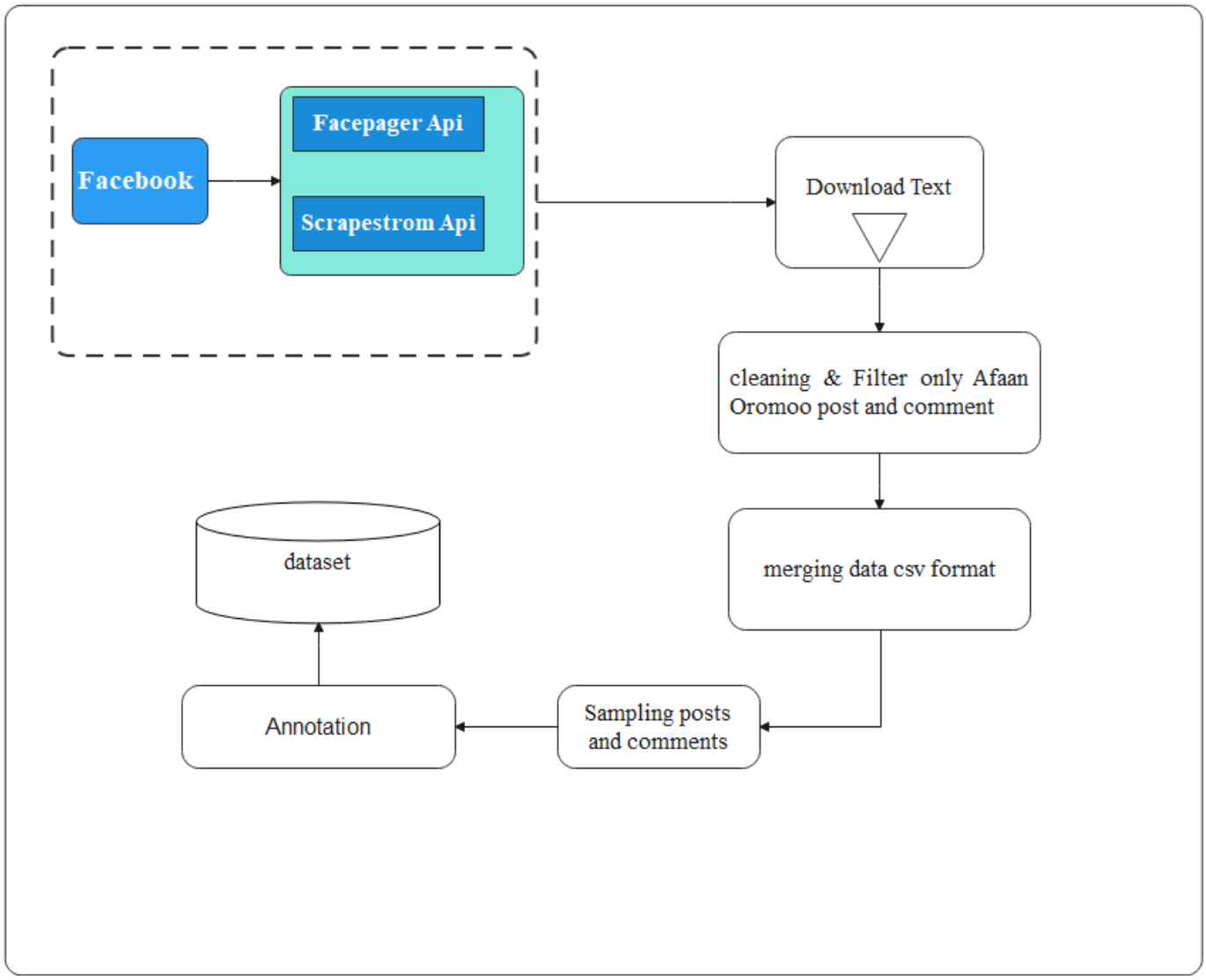

Figure 4.1 Method for Building Afaan Oromoo Hate Speech Dataset

4.2.1 Procedure to collect data

Creating the dataset consists of few steps like selecting the data source, collecting and labeling the data. Each process is discussed below. Even though there is a publicly, available dataset those are for the English language so that to achieve the goal of building the dataset the following tasks are performed.

1 Mostly followed pages of activists and news pages are selected because, over the previous years, Speeches by activists in Ethiopia have spread rapidly across social media.

2 The posts and comments were collected from those pages by using Facepager and scrape storm software, which extracts comments and posts via page Id.

3 Label those posts and comments collected from those pages as Hate Speech and Free Speech.

In this work first, we consider a definition of Hateful Speech from different toolkits, annotation rules, journals, and our work definitions.

4.2.2 Data collection

Many researchers are using the publically available dataset, which, is annotated and labeled to detect Hate Speech from social media. Some of the researchers use partial datasets from the publicly available and add

some data by own contribution. However, in the case of Afaan Oromoo Language, there is no publically available dataset published for Afaan Oromoo Hate Speech detection. The process of data collection varies, not only according to which social media is chosen to investigate but also to the modalities of data extraction. A recurring approach when collecting data for Hate Speech detection is the use of a lexicon of words that are considered Hateful[26]. Regular expressions [90]are also used as techniques to retrieve particular data from users known to have previously shared Hate Speech. In addition, several sampling metrics or criteria can be used to select these pages or a user on a social media platform. This study chooses below metrics or criteria for selecting a public page.

- A page that posts news or comment issues on religion, ethnicity, politics, gender, call for violence daily.
- The number of likes and followers of the page must be greater than 10,000, which allows more active public pages.
- A page that uses the Afaan Oromoo language most frequently for posts and comments.

Depending on the above Facebook page selection metrics and data collection method, we collect data from different public pages. The major data source was collected from the most known Afaan Oromoo activist pages, Oromoo Democratic Party official pages, and different Afaan Oromoo TV pages. Some known Afaan Oromoo new companies are Fana, FIB, and EBC. The Company is a multi-lingual radio station. The reason for choosing this page is that there is a massive user-generated opinion and availability of the electronic items for the Afaan Oromoo. It is necessary to collect data from social media to construct a dataset from scratch. We have employed a versatile Facebook crawler, which exploits the Graph API to retrieve the content of the comments from Facebook posts using Facepager and Scrape storm software. We have built a corpus of comments retrieved from Facebook public pages. These pages typically post discussions spanning across a variety of political and religious topics. By doing so, we could capture both casual conversations and politically Hated posts and comments.

4.2.3 Dataset preparation

After the data is collected, a preparation process follows, which are collecting, cleaning, filtering, and merging data into one file or data table. This process used preparation tools such as MS Excel. The Next step is, enable us to build a dataset using the partitioned CSV data. Since the Hate Speech data records are stored in a CSV file, we cannot use them directly in our system. At a preliminary stage, the constructor will read the CSV file using Pandas library, which is data manipulation and analysis tool built on top of the Python programming language, and then passes it to NumPy arrays. NumPy is a Python library used as an array processor for numbers, strings, and records. Finally, after obtaining the array values from NumPy, the dataset constructor performs the following two tasks. Then it will enable us to construct a dataset using the partitioned CSV data. During dataset preparation below tasks are performed.

- Removing all non Afaan Oromoo and non-textual posts and comments.
- Removing all null values.

- Filtering using keywords that are an indicator of Hate and normal.
- Joining data of each page into one dataset.
- Removing duplication to ensure the uniqueness of each text in a dataset.

4.2.4 Annotation guidelines

Annotation is a procedure for adding information to the collected data at some level [91].

For data, annotation researchers consider conceptual things for data annotation provides some characteristics for identifying Hate Speech and how effectively counter it. 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 following are specific rules for labeled a given post or comments as **Hate Speech**:

1. If the post/comment uses references to the alleged inferiority or superiority of some target groups.
2. If the post/comment affects different characteristics of the person and motivates audiences to take action or making a violation.
3. If post/comment contains stereotype which means over-generalized belief about a given target.
4. If post/comment Accusing or Condemning people based on their target groups.

The following are specific rules for labeled a given post or comments as **Neutral** Speech:

1. If post/comment is neither Hate Speech nor offensive Speech
2. If post/comment is expressed by government or licensed agency of government with the intent for mass advocacy and enlightenment.

The purpose of annotations are

- To organize important material.
- Monitor your learning as you read.
- Identify key concepts.
- Systematic summary of the text that you create within the document.
- A key tool for close reading that helps people uncover patterns, notice important words and identify the main point. An active learning strategy that improves compression and retention of information.

4.2.5 Annotation Procedure

The dataset annotation development, mainly achieved by the researcher, also performs annotation with two additional annotators. The number of annotators is limited because of a lack of resources, mainly budget and time need more annotators to participate in the process. The annotators were selected based on their willingness to perform the task and Afaan Oromoo language skills. Two of the annotators are Afaan Oromoo Instructors. The annotators were instructed to annotate a random subset of an equal number of posts and comments from the dataset. Due to the thought-provoking nature of the manual annotation process of Hate Speech 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 comments into the binary classifier, as defined in section 4.2. The below table shows a sample of Hate and neutral Speech detail showed in appendix 3.

Table 4.2 sample of Hate and neutral

Language	Afaan Oromoo Comment Text and its English translation class	Class
Afaan Oromoo	Sabni A*a jedhamu addunyaa kana irra hin jiru	Hate
English	There are no people called A*a	
Afaan Oromoo	Furmaati kan dhufu ollaa keenya Somalia ajjeesuudha.	Hate
English	The solution is to kill the neighbouring Somalia.	
Afaan Oromoo	Oromoon diina amaraati.	Hate
English	Oromo is an enemy of Amhara.	
Afaan Oromoo	Oromoon tokko Oromoo hundaaf.	Neutral
English	One Oromoo to All Oromoo.	
Afaan Oromoo	Oromoo fi amharaan tokko.	Neutral
English	Oromoo and Amhara is one.	

4.3 Preprocessing

Several research studies have explained that using text preprocessing makes better classification results [33]. To perform Hate Speech detection, the dataset should first be converted to an acceptable representation that should be used by the classifier. Preprocessing is the necessary part, which improves the accuracy, efficiency, and scalability of the classification process. So we applied different preprocessing techniques to filter noisy and non-informative features from the collected data. The data collected for training was in an unstructured format unsuitable for the application of Machine learning techniques. As such it was imperative to preprocess the data before passing it to the training Machine. In preprocessing, we changed the text into lowercase, removed all the URLs, usernames, white spaces, hashtags, punctuations, and stop-words from the collected data. Besides this, we also performed tokenization and stemming from preprocessed texts. The tokenization converts every single text into tokens or words, then the porter stemmer converts words to their root forms, such as offended to offend using porter stemmer. It plays a paramount role in Machine learning since Machine learning models learn from data. The preprocessing component's function is to prepare the raw dataset for training and testing our HS detection Framework.

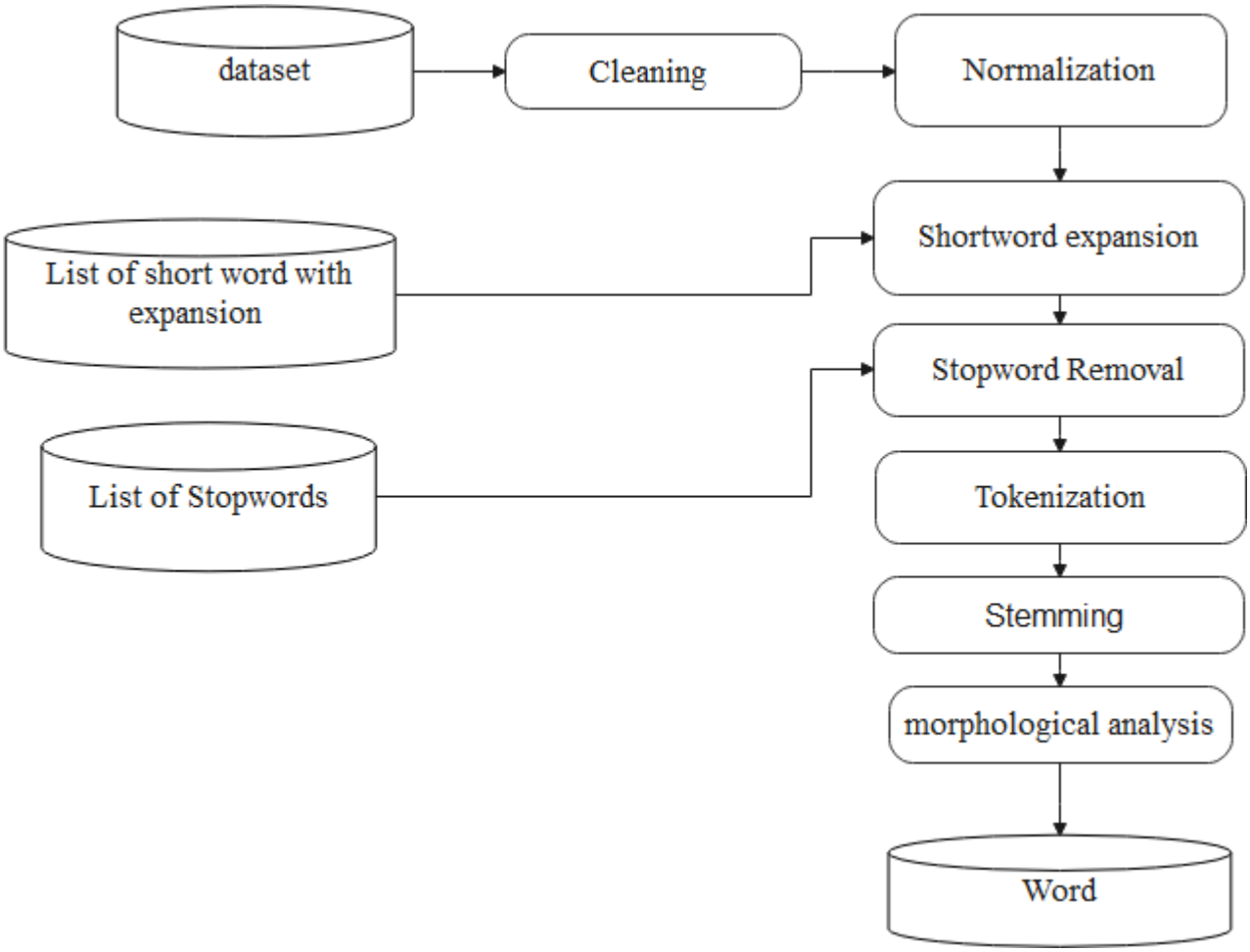

Figure 4.2 Architecture of Afaan Oromoo preprocessing Component

A. Data Cleaning

Data cleaning is the process of preparing data for analysis by detecting and removing data that is duplicated, empty or inaccurate record from a recordset. For our study the training and testing corpus is collected from online sources, which contain Afaan Oromoo text. The corpus is composed of simple sentences and complex sentences in different domains. First, we read the file and perform a text cleaning based on the languages properties. We performed removing quotes, removal of user name, removal of links, removing spaces, adding a space between the word and the punctuations, cleaning digits, and removing of special characters. We also performed Converting of text to lower case and removal of none Afaan Oromoo texts. We used the data cleaning technique to fill null values in our dataset. This removed outliers shaped our dataset to make it more consistent.

```
                Read the corpus
        For each sentence in the corpus
If the sentence is special character, empty,
  digits, non-Afaan Oromoo, HTML, URL, and
                hyperlink
              Then remove it
                 End if
              Close files
```

Algorithm 4.1 Data cleaning algorithm

B. **Stop word Removal**

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. Some Afaan Oromoo Stop words are significant for the sentiment classification and need to remain in the text. For instance, the “hin” is used to indicate the negativity of the word: for example, “dhufeera”, “hin dhufne”. In another case, some stop words constitute a phrase: “walii hin gallu”, “isin waliin jirra” etc. These Stopwords portray important information. Therefore, we filtered removed stopwords through an automatic process that is relevant for the classification process. Thus, these terms should not be considered in the indexing process. A List of some stop words is shown in Appendix 1.

```
Open corpus and stopword list
Read the corpus file
Check the word against the stopwords list
If not a word exists in the stopwords list then write it as a
candidate for Document representation
Else
remove the term from the corpus and
end
```

Algorithm 4.2 Stopword removal

C. **Short Word Expansion**

Short Word Expansion could contain words written in short forms. Short words are a short form of words or phrases, which can be formed from initial letters of important terms of a word or the combination of letters of a word or a phrase and other characters. Usually in Afaan Oromoo, '.' and'/' are used while writing words in short form. For example, “FKN” (Fakkeenya) the word “FKN” should be expanded to “Fakkeenya” (example). Some short words and their expansions are shown in Appendix 2.

D. **Data normalization**

Normalization, sometimes called feature scaling, is a method used to reduce the range of independent variables or features in the data. It includes the removal of noisy features that do not contribute to information regarding the meaning of the text. It could be a correction of misspelled words or removal of slang words [92]. Homophones like “baayee” and “baayyee” have the same meaning with different writing. The only difference is that the apostrophe “ ‘ “ is replaced by “y”.

- Normalization of elongated texts, for example, si jaallannaaaaaanna is normalized to ‘si jaallanna’
- Normalization of numbers into equivalent texts. Example: “sin jaallanna 100% “normalized to” sin jaallanna persentii dhibba tokko”.

```
Open the dataset
For each character in the dataset
If the character is jaalallaa(jaallala) or any order,
Then
Transform to (jaalala)
Else if it is fkn(fakk) or any order of it then
Changed to (fakkeenya)
Else if …. End if
End for
```

Algorithm 4.3 Normalization

Table 4.3 Sample of Normalized word

Words	Normalization
Baay'ee.	Baayyee.
Jaalalaaaaa.	Jaalala.
sin jaallanna 100%.	sin jaallanna persentii dhibba tokko.
Jaallannaaaaaannaaa	Jaallannaan

E. **Tokenization**

Tokenization is the process of splitting an input sequence into tokens. It a useful unit for semantic processing. The tokens usually consist of either a single word or what is called an N-gram, meaning that N consecutive words are split into a single token. The idea is to preserve some of the information that is stored in the order of the words. Tokenization is defined as slicing a stream of text into pieces, denoted as tokens. In Afaan Oromoo, white spaces are used to separate the boundary of a token, and punctuation marks such as commas, periods, question marks, exclamation marks, and hyphens are important to demarcate the boundaries of tokens. However, in Afaan Oromo language apostrophe mark is considered as a part of a word if it exist in single word. For example, in word baay'ee (many). Namni hunduu gaarii yaaduu qaba.the tokens will be 'Namni', 'hunduu', 'gaarii', 'yaaduu', qaba'. Thus, we can define the token as an instance of a sequence of characters. The tokenization varies from language to language but lexical characteristics such as colloquialism (e.g. "u" instead of "you"), contractions (e.g. "aren't" instead of "are not") and used by recent publications make the task harder [93].

```
Assign tokineze(text)
Tokens = re.split()
Return tokens
Dataset_name['assign name of column for output']=
dataset_name['text_column_name from
dataset'].apply9lambdaX:tokenize(X.lower()))
Dataset_name_head or tail()
```

Algorithm 4.4 Tokenization Algorithm

F. Stemming

Stemming is the technique of reducing inflection in words and an activities to find the stem of a word by removing affixes. i.e., it enables to merge of morphological variants of a word under a single index entry or its common form. Thus, for this research work, we have used Debela stemmer, which takes a word as input and removes its affixes using a rule-based algorithm [94].

G. Morphological Analysis

Morphological analysis is the segmentation of words into their component morphemes and the assignment of grammatical morphemes to grammatical categories and lexical morphemes to lexemes [95]. HornMorpho [96] is used for the morphological analysis task. HornMorpho is a Python program that analyzes Amharic, Oromo, and Tigrinya words into their constituent morphemes (meaningful parts) and generates words, given a root or stem and a representation of the words grammatical structure.

Morphological analysis is needed for morphologically rich languages like Amharic, Oromiffaa and Tigrigna. It creates many words by adding affixes to the root or stem of the word Morphological analyzer used to minimize dictionary size, the smaller the dictionary size needs less storage space, processing time and it may improve performance by reducing the dimensionality of the data since the words “Ijoollee”, ”Ijoollummaa” are treated as the same word “Ijoollee”.

4.4 Feature Extraction

The phenomenon of Hate language online involves social, cultural, linguistic, and individual factors. Hate content is therefore not homogeneous, as there is great variety between single instances of such language. The term features refer to measurable pieces of information on a post, which can be fed into a classifier to help classification. Feature extraction is quite a complex concept concerning the translation of raw data into the inputs that our particular HS detection framework requires. It provides helpful information about the stored dataset by creating a meaningful future. The most common techniques of feature extractions are Term Frequency Inverse Document Frequency (TF-IDF), Term Frequency (TF), word2vec, and Glove, N-gram. For this study TF-IDF, N-gram, and word2vec feature extraction were used because they are popular and fit my work for better results.

N-gram: N-gram is a word prediction model using probabilistic methods to predict the next word after observing N-1 words. The most common N-grams approach consists of combining sequential words into lists with size N, where N is number of words used during probability sequences. This study uses a word N-gram method to create N-gram of posts and comments features as defined in more detail discussed in chapter two subsection 2.7.1.3.

TF-IDF: TF-IDF is also one of the most feature modeling methods used in Hate Speech detection. TF-IDF is a Measure of the importance of a word in a document within a dataset and increases in proportion to the number of times that a word appears in the document. More detail discussed in chapter two subsection 2.7.1.4.

Word2vec: Word2vec takes a large corpus of text as input and produces a vector space that will be used when training a model for Afaan Oromoo Hate Speech detection. Word2vec is a collection of models capable of

capturing the semantic similarity between words based on the sentential contexts in which these words occur. It does so by projecting words into an n-dimensional space and giving words with similar contexts similar places in this space [97]. This method is used due to the absence of standard Afaan Oromoo word2vec model, and it is recommended that to build domain-related models for better results. The word2vec model containing a vectors space and a similarity of all the words in posts and comments. More detail is discussed in chapter two subsection 2.7.1.4.

4.5 Feature Selection

Feature selection can be defined as the process of selecting a subset of the terms occurring in a corpus and using only this subset as features in text classification [98]. It would help in identifying Hate and no Hate posts and can be used in the modeling of the classification problems. The feature space for text classification problems consists of all the unique terms that occur in a document. The number of features can therefore be quite big for a corpus that is average sized. This high dimensionality of the feature space is an inherent characteristic of text classification problems and poses a significant problem too many Machine learning algorithms [99]. The high dimensional feature space may result in poor accuracy results and over fitting. It is therefore important to reduce the feature space to improve the performance of the learning algorithms, reduce overfitting and improve the time needed to train a model. Feature selection is primarily focused on removing non-informative or redundant predictors from the framework. Generally, the quality of the features has a significant impact on the performance of our proposed Hate Speech detection framework. There may be some features that will not affect the classification result at all, or perhaps might make the result worse. Therefore, since some attributes might not equally contribute to the classification result, through feature selection the most relevant subset of features in the dataset is selected for analysis.

4.6 Model Training

The model will be fit using the training dataset, which contains a set of Hate and normal records to train the model. The corpus was then randomly split into two sets using the train_test_split method of the scikit-learn library; 80 percent of the corpus was used to train the model whereas the remaining 20 percent was used to test its performance. Before the training data could be passed to the SVM classifier for training, it had to be converted into a document term matrix suitable for learning. A count vectorizer was used to convert the text into a matrix of token counts. Since no dictionary had been defined beforehand and no feature selection method was used, the number of features used was equal to the vocabulary size found by analyzing the data. TF-IDF weighting was then applied to the matrix to transform the counts into TF-IDF weights. An SVM classifier could then be applied to the document term matrix. The SVMs implementation is provided by python's scikit-learn Machine learning library.scikit-learn's CountVectorizer module was used to transform the training data into a matrix of token counts and the Tfidf transformer was used to transform the counts using TF-IDF weighting. Both vectorization and transformation steps can be combined into a single scikit-learn pipeline.

4.7 Test dataset

The testing dataset is used to assess the classification performance of our model. It provides an unbiased evaluation of the final model that is fit on the training dataset. The first one is test of the model using a testing dataset, which is 20% of the dataset by which we will see how the trained model performs on the entire defined test data. In this case, the model attempts to predict the labels for the given testing datasets, uses the test labels to measure the accuracy of those given testing datasets, and uses the labels to calculate the accuracy of those predictions.

```
In [49]: #model Building
         import sklearn.model_selection as model_selection
         from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X, df_y, test_size=0.2, random_state=42)
         # where
         # each train and test have two components which is x , y.
         # train (x,y), test (x,y)
         # where x is two dimension array
         # train size parameter is a sets of valu assigned for the training out of total data available in a dataset.
         #test_size parameter is a sets of value used for testing.
         #random_state parameter is a value used to generate random numbers
```

Figure 4.3 sklearn function for dataset division

4.8 Classifier

Classification is a subset of Supervised Learning methods and constitutes an automatic process to identify which label corresponds to a new observation based on an already known labeled training dataset. Classifier algorithm SVM, RF, and NB on the dataset features and labels. Which could be selected for the detection of the Afaan Oromoo Hate Speech. This study proposed the One-vs-Rest (OVR) strategy of Support Vector Machine (SVM) Machine learning classifier. It used to separate one class from another of the classes.

SVM

Support Vector Machine is a supervised Machine learning model that can be used to solve classification and regression problems. SVM model finds a linear classifier that separates data in some high dimensional space. SVMs have extensively been used in NLP contexts, and are especially effective when there is relatively little training data. These models are discriminative in that they can distinguish between entities into their respective classes without explicit knowledge of these classes. These models make use of different kernels such as linear, Rbf, and so on, to transform the data before separating them based on the labels generated. For this study, a linear kernel is used to preform detection.

4.9 The Detection Component

The detection of Hate Speech is done after the text features are learned. The learning of the text features occurs after a careful design of appropriate components. In any ML framework, training and testing are a standard workflow. Hence, our detection component comprises the training component the testing component. In this section, we will briefly describe each element of our proposed model.

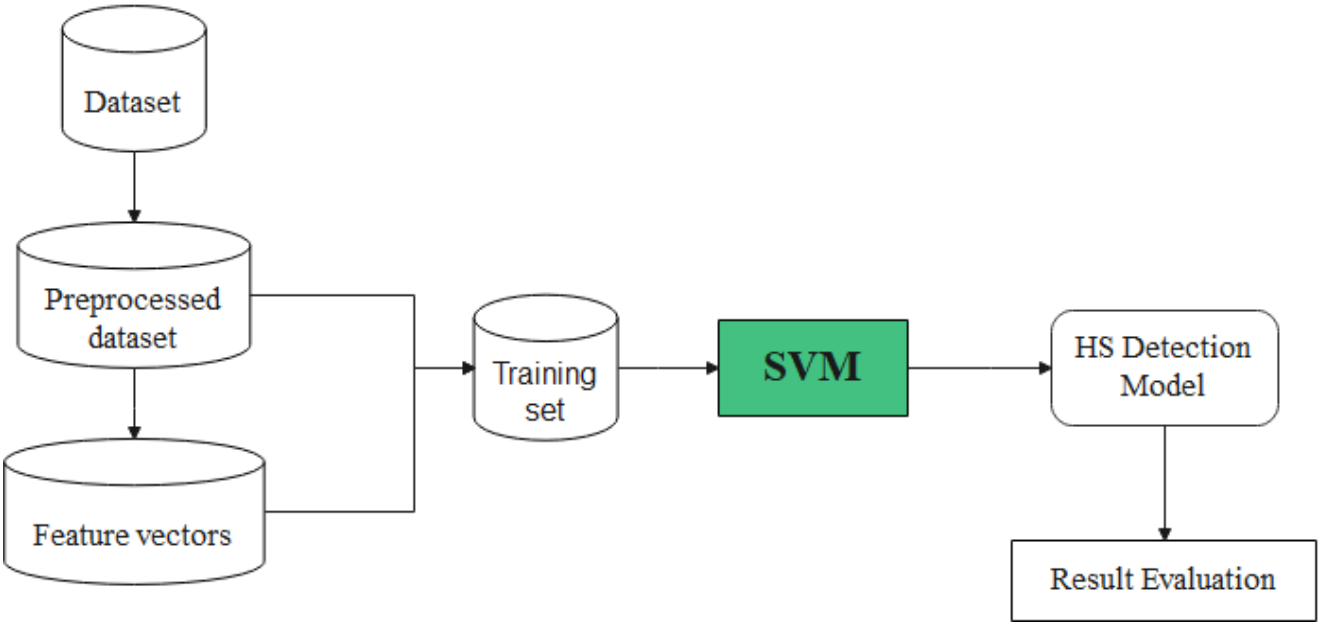

Figure 4.4 Detection framework flow Diagram

Chapter:5 System Implementation And Experimentation.

Chapter five presents the implementation of the proposed solution for Afaan Oromoo Hate Speech detection by using the methodology and proposed solution discussed in chapter four. This study follows an experimental approach to determine the best result by cross checking different Machine learning algorithms and feature extraction. During the experiment, we use the unbalanced dataset. In addition we use different feature extraction techniques. The experiment performs by using a binary class dataset. Finally, develop a prototype for Hate Speech detection by using a model.

5.1 Development Tools and Techniques

This study uses several development tools and packages to implement the proposed solution, Afaan Oromoo Hate Speech detection. This study uses python programming language for implementing and experimenting with each proposed solution from the data preprocessing to the model building steps. Python was used for the evaluation of the implemented proposed classifier model. Python is used because researchers prefer language and python programming languages have many libraries, which support natural language processing. In the end, the Flask framework is used to develop a web application since it provides tools, libraries, and technologies that are useful to build a web application. For doing whole tasks, we used Jupyter Notebook (Anaconda3) which is a web-based interactive computing notebook environment for edit and run python codes in each code separately. One of the purposes of Jupter notebook is easy to find errors since each cell displays its own output. Table 5.1 shows the list of tools and python packages with their Version and description, which is used in this study.

Table 5.1 Description of the Tools and Python Package Used During the Implementation

Tools	Version	Description
Pandas profiling	2.11.0	Pandas-profiling is used to display sample, correlation, and duplicated data easily
Microsoft Excel	2016	Used data preparation tasks during data are crawled from Facebook pages and sorting the gather data. Also, used to manage the annotation task.
Facepager and Scrape	3.6 and	Social media content retrieval tools. Used for data collection tasks to build the dataset. This tool is used to fetch posts, comment on Facebook a public page, and store the

Storm	3.5.3	data in SQLite database. Function to export the database to CSV file which is easy for manage datasets.
Anaconda Navigator	1.10.0	Allows us to launch development applications and easily manage condominium packages, environments, and channels without the need to use command-line commands.
Jupyter notebooks	6.1.4	An open-source web application that allows us to create and share documents that contain live code, equations, visualizations, and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, and Machine learning.
Python	3.7	Powerful programming language to develop a Machine learning application. It is also easy to process natural language.
Pandas	1.1.3	High-performance, easy-to-use data structures, and data analysis tools. This study uses it for data reading, manipulation, writing, and handling the data frame.
Flask	1.1.2	Microframework for making web services in python. This study uses it for implementing the prototype for selected models.
Genism	4.0.1	Python library for topic modeling document indexing and similarity retrieval with large corpora. This study uses it for the constricting word2vec model.
RegEx (Re)	___	A regular expression (or RE) specifies a set of strings that matches it; the functions in this module let you perform string matching, removal, replace, etc. package used in this study to perform preprocessing of Afaan Oromoo text.
Scikit-learn	0.23.2	A set of python modules for Machine learning and data mining. This study uses it for feature extraction, training, and testing model. The name of the package is called sclera.
Nltk	3.5.0	Build python programs to work with human language data. This study uses it for tokenization and stopword removal.
Matplotlib	3.3.2	Publication quality figures in python. This study uses it for data and results visualization.

5.2 Deployment Environment

The tools which are discussed above in section 5.2 have been deployed on a personal computer equipped with a processor Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz 2.40 GHz, 4.00 GB , 465 Gigabyte hard disk storage capacities. The operating system is Windows 10 Pro, 64 bits.

5.3 Dataset Description

In order to build the dataset for this study, we collected posts and comments from Facebook using the content retrieval tools Facepager and Scrape Storm. Based on Facebook page selection criteria.

5.3.1 Building the Corpus

Aiming at developing automatic Hate Speech detection in social media for Afaan Oromoo texts, authors have built a corpus of comments and posts retrieved from Facebook public pages of Ethiopian newspapers,

individual and group politicians, activists, TV, and Radio broadcast, different group association pages. We have employed a versatile Facebook crawler, which exploits the Graph API to retrieve the content of posts and comments from Facebook pages. Therefore, we used scrape storm and Facepager to crawl the data from Facebook and other different Media.

It extracts different attributes such as object_id, from, created time, Message.

The collected dataset contains four columns.

- Object_id is a Facebook Id for the post and comment.
- Created time is the timestamp that the post or comment created.
- The Message contains the actual post and comments.
- From is name of pages from where data is crawled.

Label is the class the annotator assigned to the comments and posts. However, we filter only the message content for our purpose.

Qeerroon Oromoo: Roorroofi Hacuuccaa Ilmaan Oromoo Warshaa Eastern Industry Zone jedhamu keessa hojjetan irratti hammachuu irraa Hojjettootni
Afaan Oromoo
[October 21, 2018] Dirree Dhawaa:Hirmaatootni sagantaa simmannaa jiila itti anaa Hayyu Duree Adda Bilisa Baasaa Oromoo Oromiyaa Bahaa (ABO) Jaal Araarsaa Biqilaatiin TV
[October 21, 2018] Wow Must Listen! Gootittii Baddaa Baalee Haadha qabsoo Nuuf dhiyeessuu keetiif ulfaadha Dejene Gutema fi OMN! Oromo
[October 21, 2018] Fayyisaa leellisaa ergaa Oromiyaa ti galee Ibsa gazixeessitootaf kenne. TV
[October 21, 2018] Qeerroo, Qarree, Abbootii Fardaa?hundi isaanii bara ukkaamsaa san keessa kan darban qabsoo hadhooftuudhaan. News
[October 21, 2018] Dirree Dhawa- Kun simannaa ABO magaalaa Dirree Dhawaatti guyyaa har?aa gaggeeffame. News

Figure 5.1 Sample data Collected

5.4 Preprocessing

The data collected for training was in an unstructured format unsuitable for the application of Machine learning techniques. As such, it was imperative to pre-process the data before passing it to train models.

no like	no commen	date	post
Like	13 Commer	[October 21, 20	Roorroofi Hacuuccaa Ilmaan Oromoo Warshaa Eastern Industry Zone jedhamu keessa hojjetan irratti hammachuu irraa Hojjettootni
Like	73 Commer	[October 21, 20	Afaan Oromoo
Like	136 Comm	[October 21, 20	Dirree Dhawaa:Hirmaatootni sagantaa simmannaa jiila itti anaa Hayyu Duree Adda Bilisa Baasaa Oromoo Oromiyaa Bahaa (ABO)
Like	11 Commer	[October 21, 20	Wow Must Listen! Gootittii Baddaa Baalee Haadha qabsoo Nuuf dhiyeessuu keetiif ulfaadha Dejene Gutema fi OMN! Oromo
Like	95 Commer	[October 21, 20	Fayyisaa leellisaa ergaa Oromiyaa ti galee Ibsa gazixeessitootaf kenne. TV
Like	87 Commer	[October 21, 20	Qeerroo, Qarree, Abbootii Fardaa?hundi isaanii bara ukkaamsaa san keessa kan darban qabsoo hadhooftuudhaan. News
lke	40 Commer	[October 21, 20	Dirree Dhawa- Kun simannaa ABO magaalaa Dirree Dhawaatti guyyaa har?aa gaggeeffame. News

Figure 5.2 Collected Comments with Separated Columns

The collected data often contain a number of different types of information including no like, comment, date, post, text, mentions, hashtags, id, and permalink. The research was only interested in the text part of the collected data to build the corpora. The collected data were therefore processed to separate the various parts as illustrated in Figure 12 to easily extract the text part of the comments that are relevant to this study. On analysis of text contained in the collected data a number of data contain avoidable for this research. Such observed terms included hashtags(#), (@username), URL-links, non Afaan Oromoo Languages, and non-utf8 characters in dataset. All listed above were unnecessary for building the corpora and were therefore removed using regular

expressions as depicted in the code below in figure 5.4.

5.4.1 PreProcessing Implementation

To pre-process the dataset we perform the following common activities, which are importing important libraries packages and load the dataset file to the disk using the code in figure

```
In [17]: import pandas as pd
         df= pd.read_csv("data22.csv")
```

Figure 5.3 Python Code for Load the Dataset Post and Comment

5.4.2 Implementation of Cleaning Irrelevant Characters

To remove unnecessary characters, symbols, special character, and Stopword from imported data. We implement a method that performs removal and replacements. The preprocessing method accepts the comments and then removes or replaces with a single space the unwanted characters. At the end, the method returns are clean comments.

```
In [24]: # 1. Removal of punctuation and capitlization  2. Tokenizing 3. Removal of stopwords
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem.porter import *
stopwords = nltk.corpus.stopwords.words("oromic")
#extending the stopwords to include other words used in twitter such as retweet(rt) etc.
other_exclusions = ["#ff", "ff", "rt"]
stopwords.extend(other_exclusions)
stemmer = PorterStemmer()
def preprocess(comments):
    # removal of extra spaces
    regex_pat = re.compile(r'\s+')
    comments_space = comments.str.replace(regex_pat, ' ')
    # removal of @name[mention]
    regex_pat = re.compile(r'@[\w\-]+')
    comments_name = comments_space.str.replace(regex_pat, '')
    # removal of links[https://abc.com]
    giant_url_regex =  re.compile('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|'
            '[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+' )
    commentss = comments_name.str.replace(giant_url_regex, '')
    # removal of punctuations and numbers
    punc_remove = commentss.str.replace("[^a-zA-Z]", " ")
    # remove whitespace with a single space
    newcomments=punc_remove.str.replace(r'\s+', ' ')
    # remove leading and trailing whitespace
    newcomments=newcomments.str.replace(r'^\s+|\s+?$','')
    # replace normal numbers with numbr
    newcomments=newcomments.str.replace(r'\d+(\.\d+)?','numbr')
    # removal of capitalization
    comments_lower = newcomments.str.lower()
    # tokenizing
    tokenized_comments = comments_lower.apply(lambda x: x.split())
    # removal of stopwords
    tokenized_comments=  tokenized_comments.apply(lambda x: [item for item in x if item not in stopwords])
    # stemming of the tweets
    tokenized_comments = tokenized_comments.apply(lambda x: [stemmer.stem(i) for i in x])
    for i in range(len(tokenized_comments)):
        tokenized_comments[i] = ' '.join(tokenized_comments[i])
        commentss_p= tokenized_comments
    return commentss_p
label_commentss = preprocess(comments)
df['precommentss'] = label_commentss
print(df[["comments","precommentss"]].head(10))
```

Figure 5.4 Code for preprocessing dataset

5.4.3 Implementation of Normalization and short word expansion

Short words are a short form of words or phrases, which can be formed from initial letters of important terms of a word. Normalization, sometimes called feature scaling, is a method used to reduce the range of independent variables or features in the data. In case of Afaan Oromoo normalization very essential, because the most sentence of Afaan Oromoo is written with an apostrophe. It means that apostrophe is count as a character but the apostrophe is used for tokenization so we design to replace apostrophe with related words. Example

“Baay’ee” is replaced with “ Baayyee” which is the same meaning in Afaan Oromoo.

For both normalization and short word expansion, we design simple code in figure 5.5 below. Lists of words are in Appendix1

```
In [33]: #Code for shortword expansion and normalization of text.
d = {'Qar':'Qarshii','Bill':'Billiyoona','Mill':'Milliyoona','A.L.A':'Akka Lakkoofsa Awuroopa','A.L.I':'Akka Lakkoofsa Itoophiyaa
              'Ykn':'Yookiin','Kkf(K.K.F) ':'Kan Kana Fakkaatan','M/B ':'Mana Baruumsaa','fkn':'Fakeenya','Pirof':'Pirofeesara',
              'Dr':'Dooktoora','I/G':'Itti Gaafatamaa',
              'M/Murtii ':'Mana Murtii','Hosp. ':'Hoospitaala','M/Ministeeraa':'Muumee Ministeeraa','Lakk':'Lakkofsa','Dh.K.D':'
              'Dh.K.B':'Dhaloota Kiriistoosin Booda','Hogg.':'Hooganaa','H/Bulaa':'Hoorsisee Bulaa','FQ/Bulaa':'FQoonaan Bulaa',
              'Dr.':'Dooktoora','I/G':'Itti Gaafatamaa',"baay'ee":'baayyee',"ta'u":'tahuu',"ba'eessaa":'baheessa',
    "ta'uun":'tahuun',"ba'uuf":'bahuuf',"du'an":'duhan',"dhaga'ame":'dhagaame'," jeâ€™e":'jedhe',"ka'e ":'kahe',"Br/j":'Birgaader
    "waa'ee":'wahee',"ka'aa":'kahaa',"qopha'iin":'qophahiin',"dhaga'ame":'dhagahaame',"ta'anii":'tahanii',"ta'aa":'tahaa',"joonja
     "taâ€™uu ":'tahuu',"ba'ee":'bahee',"baâ€™u":'bahuu',"FDG":'finchila diddaa garbummaa',"gaâ€™e":'gahe',"Kaâ€™i":'kahii',
     "mulâ€™isuu":'mulisuu',"Har'aa":'haraa',"sochoâ€™aa":'sochohaan',"taaâ€™uun":'taahuun',"taâ€™ee":'taahee',"Dukkanaaâ€™us":'
     "duâ€™ee":'duhee',"kaâ€™eera":'kaaheera',"Harâ€™a":'haraa','ONN:':'oromiyaa media network',"Milkaaâ€™innaa":'milkaahinna',"
     "duâ€™aan":'duhaa',"kaâ€™ee":'kaahee',"buâ€™anii":'buhaanii',"deebiâ€™a":'deebihaa','2"@12"ff':''

    } ## Need a huge dictionary
words = "kaâ€™ee"
words = words.split()
reformed = [d[word] if word in d else word for word in words]
reformed = " ".join(reformed)

In [157]: #reformed

In [35]: df['reform'] = df['comments'].apply(lambda x : ' '.join(d[word] if word in d else word for word in x.split()))
```

Figure 5.5 Code for short word expansion and normalization of words

5.5 Implementation of Feature Extraction

To extract Feature extraction from a dataset that is used for training the model and help the researcher to find get key feature names from the whole dataset. In order to implement Feature extraction we used the python Scikit learn module for TF-IDF and N-gram and Genism module for Word2vec. To implement feature extractor method, first, the dataset must be clean and normalize using preprocessing methods and each feature extraction use list of token created from comments as the input. Then each token is vectorized with vector transformation . Because Machine learning models operate on numbers (vectors) instead of words to train models.

5.5.1 Implementation of N-gram

To implement N-gram, the study uses the CountVectorizer class of sci-kit learn feature extraction submodule. In N-gram the implementation dataset convert to a matrix of N-gram features vector by N-gram ranges.

```
In [108]: from sklearn.feature_extraction.text import CountVectorizer
          n_gram_feature = CountVectorizer(
              ngram_range=(1,2),
              #tokenizer=tokenize_text,
              analyzer='word',
              min_df=3,
              max_df=0.75
          )
          n_gramdata = n_gram_feature.fit_transform(comments).toarray()
```

Figure 5.6 Sample Code Used to Generate n-gram Features

5.5.2 Implementation of TF-IDF

TF-IDF Vectorizer class of sci-kitlearns package is selected for this study. This TF-IDF vector class convert comments of dataset to matrix of TF-IDF feature vectors. Which contain the word frequency and their importance in dataset.

```
In [37]: # convert the comment into statistical numbers
         from sklearn.feature_extraction.text import TfidfVectorizer
         vectorizer = TfidfVectorizer(ngram_range=(1,3),
                                      use_idf=True,
                                      smooth_idf=False,
                                      norm=None,
                                      decode_error='replace',
                                      max_features=2267,
                                      min_df=5,
                                      max_df=0.75)
         tfidf = vectorizer.fit_transform(df['finalaohsd'])
         tfidf

Out[37]: <2780x2267 sparse matrix of type '<class 'numpy.float64'>'
                with 28942 stored elements in Compressed Sparse Row format>
```

Figure 5.7 Sample Code for Extracting TF-IDF

5.5.3 Implementation of Word2vec

To implement word2Vec, this study uses a python Genism module, which is straightforward.

It works after import and instantiate word2vec class with necessary parameter and builds the vocabulary, and training the Word2vec model using the comments that are gathered from the selected pages.

```
In [82]: from gensim.models import Word2Vec
         w2v_model = Word2Vec(min_count=2,
                              window=10)
         w2v_model.build_vocab((df['finalaohsd']),progress_per=10000)
         w2vmodel=w2v_model.train((df['finalaohsd']),total_examples=w2v_model.corpus_count,epochs=30,report_delay=1)
         w2v_model.save('w2vmodel21.bin')
         w2v_model

Out[82]: <gensim.models.word2vec.Word2Vec at 0x1b005386520>
```

Figure 5.8 Sample Code for Building Word2vec Feature Model

5.6 Machine Learning Models Implementations

Machine learning models need own library go give function so it needs to import all library first before start code. To build Machine learning models, we used a python sci-kit learning library package with

The important library package for modeling and metrics for model evaluation.

```
In [1]: import numpy as np
        import wordcloud
        import pandas_profiling
        import pandas as pd
        import string
        import nltk
        import seaborn
        import seaborn as sns
        import matplotlib.pyplot as plt
        import seaborn as sns
        from wordcloud import WordCloud
        from nltk.tokenize import word_tokenize
        from nltk.corpus import stopwords
        from nltk.stem.porter import *
        from sklearn.feature_extraction.text import CountVectorizer
        from sklearn.feature_extraction.text import TfidfVectorizer
        from sklearn.metrics import confusion_matrix
        from textstat.textstat import *
        from sklearn.linear_model import LogisticRegression
        from sklearn.model_selection import train_test_split
        from sklearn.metrics import f1_score
        from sklearn.feature_selection import SelectFromModel
        from sklearn.metrics import classification_report
        from sklearn.metrics import accuracy_score
        from sklearn.svm import LinearSVC
        from sklearn.ensemble import RandomForestClassifier
        from sklearn.naive_bayes import GaussianNB
        from nltk.sentiment.vader import SentimentIntensityAnalyzer as VS
        import warnings
        warnings.simplefilter(action='ignore', category=FutureWarning)
        %matplotlib inline
        import warnings
        %matplotlib inline
        warnings.filterwarnings("ignore", category=DeprecationWarning)
        from nltk.corpus import stopwords
        stop = stopwords.words('oromic')
        import pickle
        import sys
        import re
        import plotly.express as px
        from textstat.textstat import *
```

Figure 5.9 Important Package for Modeling

At initial the stage, the constructor will read the CSV dataset using panda's library, which is a data manipulation and analysis tool built on top of the python programming language.

After the dataset is constructed it passes it to Numpy arrays. Numpy is used to process numbers, strings, and records. Finally, the dataset constructor performs splitting dataset to train and test set and CSV Data Conversion to appropriate format.

```
In [2]: import pandas as pd
        df= pd.read_csv("data22.csv")
```

Figure 5.10 Code for reading dataset

The study used the LinearSVC () classifier of the sklearn package to building the SVM model.

```
#SVM
from sklearn.svm import LinearSVC
clf = LinearSVC(multi_class='ovr',
                penalty='l2',
                loss='squared_hinge',
                dual=True,
                C=1.01,
                class_weight='balanced',
                verbose=0,
                max_iter=1374,
                random_state=20)
clf.fit(X_train, y_train)
clf.score(X_test, y_test)
```

Figure 5.11 Implementation of SVM Machine learning model

To implement the RF classifier, the LogisticRegressionClassifier () method of the sklearn ensemble package is used to train the classifier. This classifier fits several decision tree classifiers as declare in estimator's parameters.

```
In [76]: #Logistic Regression
         X_train, X_test, y_train, y_test = train_test_split(X.toarray(), y, random_state=20, test_size=0.2)
         model_lr=LogisticRegression()
         model_lr.fit(X_train,y_train)
         y_preds = model_lr.predict(X_test)
         acc=accuracy_score(y_test,y_preds)
         report = classification_report( y_test, y_preds )
         print(report)
         print("Logistic Regression:",acc)
```

Figure 5.12 Implementation of LR Machine learning model

To implement the RF classifier, the RandomForestClassifier () method of the sklearn ensemble package is used to train the classifier. This classifier fits several decision tree classifiers as declare in estimator's parameters.

```
In [77]: X_train_tfidf, X_test_tfidf, y_train, y_test = train_test_split(X, y, random_state=20, test_size=0.2)
         rf=RandomForestClassifier()
         rf.fit(X_train_tfidf,y_train)
         y_preds = rf.predict(X_test_tfidf)
         acc1=accuracy_score(y_test,y_preds)
         report = classification_report( y_test, y_preds )
         print(report)
         print("Random Forest, Accuracy Score:",acc1)
```

Figure 5.13 Implementation of RF Machine learning model

To implement the NB classifier, the study again uses a sklearn GaussianNB() classifier. This classifier is suitable for classification data with discrete features vector and multi-class data.

```
In [78]: X_train_tfidf, X_test_tfidf, y_train, y_test = train_test_split(X.toarray(), y, random_state=20, test_size=0.2)
         nb=GaussianNB()
         nb.fit(X_train_tfidf,y_train)
         y_preds = nb.predict(X_test_tfidf)
         acc2=accuracy_score(y_test,y_preds)
         report = classification_report( y_test, y_preds )
         print(report)
         print("Naive Bayes, Accuracy Score:",acc2)
```

Figure 5.14 Implementation of NV Machine learning model

5.7 Using the Model in Prediction

Using the model to predict other comments and posts involves the user entering keywords, which are using the model to predict other comments and posts. To avoid retraining the model every time it was required for prediction the model had to be persisted, this involved dumping the model in pickle file done using the joblib module in sklearn using the below code.

```
In [445]: svmML = open("myFinalModel.pkl", "wb")

In [446]: pickle.dump(clf, svmML)

In [447]: svmML.close()

In [448]: #Load the model
          my_model = open("myFinalModel.pkl", "rb")
          new_model = pickle.load(my_model)
```

Figure 5.15 Persisting learning model

5.8 Prototype of the System

The system prototype is developed using python programming language using a jupyter notebook and pycharm. The developed system provides the functionalities for analyzing user-generated input texts. Users can provide the input texts into the system by providing the sentences one by one on the provided GUI application that is accessed by the user through a web browser. Then, the system generates the detected value of the sentences on the other browser page. For creating the application, we were used FLASK python micro-framework. The

trained model is deployed in our local Machine and the application utilizes the predictive capabilities through HTTP requests. When we run the code or application by using jupter notebook it opens a default web browser and navigate to http://localhost:8888/ and when we run the application or code by using pycharm run flask web to interact with http://127.0.0.1:5000/ we should see a simple website form working only for our local Machine clicking above link to check the model.

Figure 5.16 Hate Speech detection system Graphical user Interface

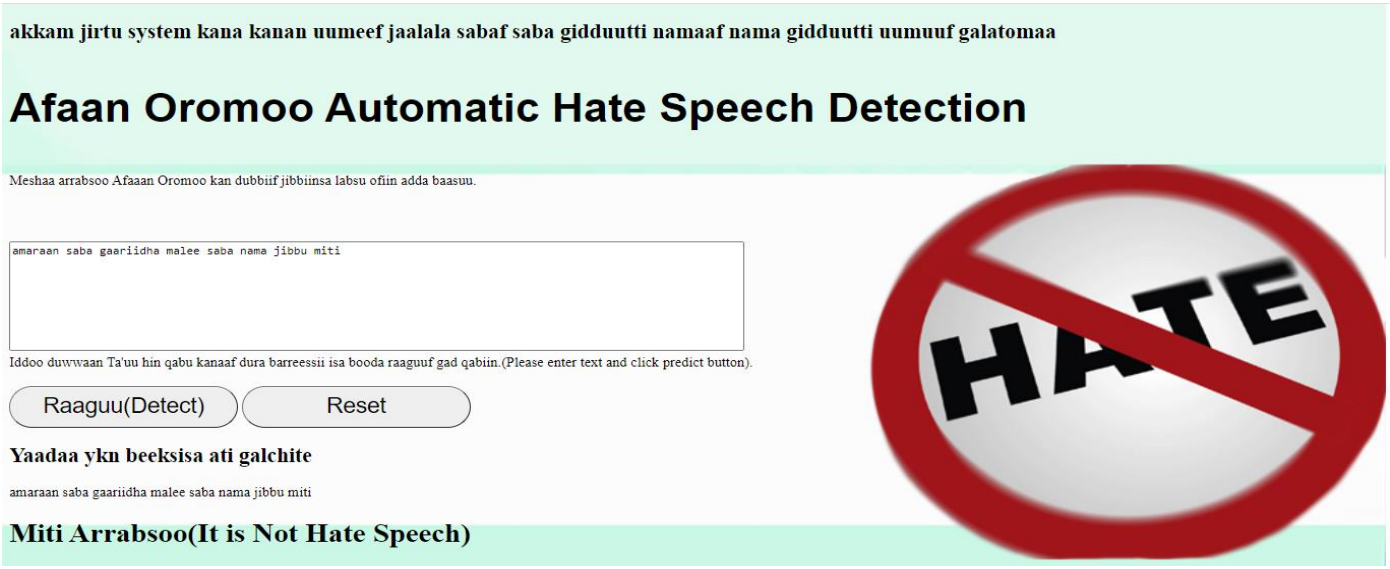

Figure 5.17 sample of normal Speech detection

Figure 5.18 sample of Hate Speech detection

Chapter:6 Result and Discussions.

We discuss the results of the experiments of the proposed solution for Hate Speech detection using linear SVM model and others Machine learning is discussed. SVM model was created using n-gram, TF-IDF, and word2vec features and performance tested against the test set. In addition it compare the performance of the linear SVM model with three Machine learning algorithms(Logistic Regression , Naïve Bayes and Random Forest). Finally, The result of the data annotation and implication of the result obtained by each experiment are discussed.

Table 6.1 The Two-Class Distribution of the Dataset

Label of class	Numbers of comments and posts	Numbers of comments and posts in percentage
Hate	1122	40.4%
Neutral	1657	59.6%
Total	2780	100%

In order to build the binary class dataset, the two-class dataset converted to two class datasets by Considering all none Hate to neutral.

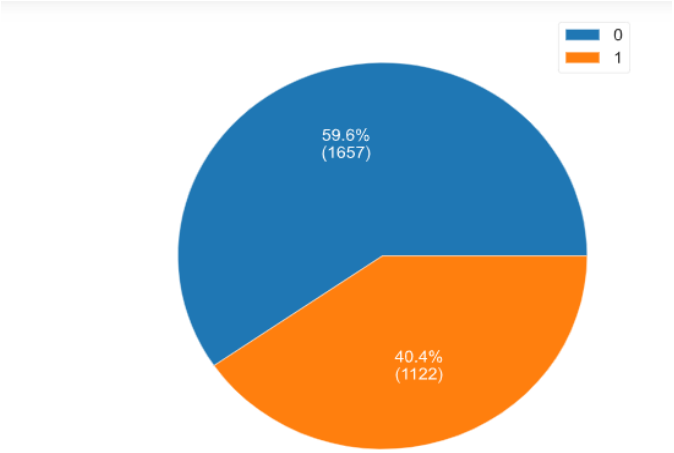

Figure 6.1 Two-Class Distribution of The Dataset

6.1 Evaluation Results

This experiment is aimed to compare the performance of the linear SVM model with other Machine learning algorithms (Logistic Regression, Naïve Bayes, Random Forest) using the same TF-IDF feature where a term is considered to be a word(Trigram). The results of the experiment are summarized in table 6.2.

Table 6.2 Performance Comparison of Different Classifiers

Classifier	Accuracy	F1-score	Precision	Recall
LinearSVM	0.96	0.96	0.96	0.96
Naïve Bayes	0.94	0.95	0.95	0.95
Logistic Regression	0.94	0.94	0.94	0.94
Random Forest	0.94	0.95	0.95	0.95

The results in table 6.2 show that the linear SVM model performs better than Naïve Bayes, Logistic Regression, and Random Forest across all the evaluation metrics (accuracy, f1-score, precision, and recall)

6.1.1 LinearSVM Classification Models Evaluation Results

Table 6.3 summary of linear svm classification results with different parameters

Feature name	Average Accuracy (%)	Evaluation metrics results		
		Precession	Recall	F1 score
Ngram+tfidf	0.96	0.96	0.96	0.96
W2vec	0.88	0.89	0.88	0.88
Tfidf+w2vec	0.915	0.94	0.94	0.94
Sentiment	0.90	0.95	0.95	0.95
Tfidf+w2vec+sentiment	0.94	0.94	0.94	0.94

The results in table 6.3 show that the linear SVM model with ngram+TF-IDF performs better than other features across all the evaluation metrics (accuracy, f1-score, precision, and recall)

6.1.2 Random Forest Classification Models Evaluation Results

Table 6.4 summary of Random Forest classification results

Feature name	Average Accuracy (%)	Evaluation metrics results		
		Precession	recall	F1 score
Ngram+tfidf	0.94	0.95	0.95	0.95
w2vec	0.95	0.95	0.95	0.95
Tfidf+w2vec	0.95	0.96	0.96	0.96
Sentiment	0.95	0.96	0.96	0.96
Tfidf+w2vec+sentiment	0.96	0.96	0.96	0.96

The results in table 6.4 show that the linear RF model with (w2vec+TF-IDF+sentiment) performs better than across all the evaluation metrics (accuracy, f1-score, precision, and recall).

6.1.3 Logistic Regression Classification Models Evaluation Results

Table 6.5 summary of Logistic Regression classification results

Feature name	Average Accuracy (%)	Evaluation metrics results		
		Precession	Recall	F1-score
Ngram+tfidf	0.94	0.94	0.94	0.94
w2vec	0.94	0.94	0.94	0.94
Tfidf+w2vec	0.52	0.48	0.51	0.49
Sentiment	0.94	0.95	0.95	0.50
Tfidf+w2vec+sentiment	0.93	0.94	0.94	0.94

The results in table 6.5 show that the linear RF model with (ngram+TF-IDF, w2vec) performs better than across all the evaluation metrics (accuracy, f1-score, precision and recall) but Sentiment perfume better in precession, and Recall)

6.1.4 Naive Bayes Classification Models Evaluation Results

Table 6.6 summary of Naive Bayes classification results

Feature name	Average Accuracy (%)	Evaluation metrics results		
		precession	Recall	precession
Ngram+tfidf	0.88	0.90	0.88	0.88
w2vec	0.94	0.95	0.95	0.95
Tfidf+w2vec	0.91	0.93	0.92	0.92
Sentiment	0.91	0.93	0.92	0.92

Tfidf+w2vec+sentiment	0.91	0.93	0.92	0.92

The results in table 6.6 show that the linear NV model with (w2vec) performs better than across all the evaluation metrics (accuracy, f1-score, precision, and recall)

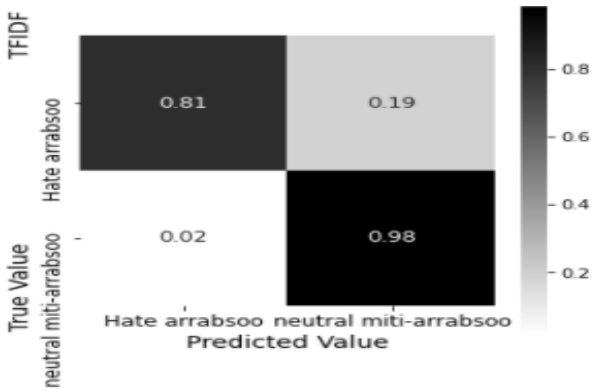

Figure 6.2 TF-IDF Confusion Matrix classification evaluation

Figure 6.2 illustrates the sampled confusion matrix for SVM models. SVM with bigram classify 0.81% of Hate Speech and 0.98% of neutral correctly, and 0.19% of Hate Speech and 0.02 % of neutral are misclassified.

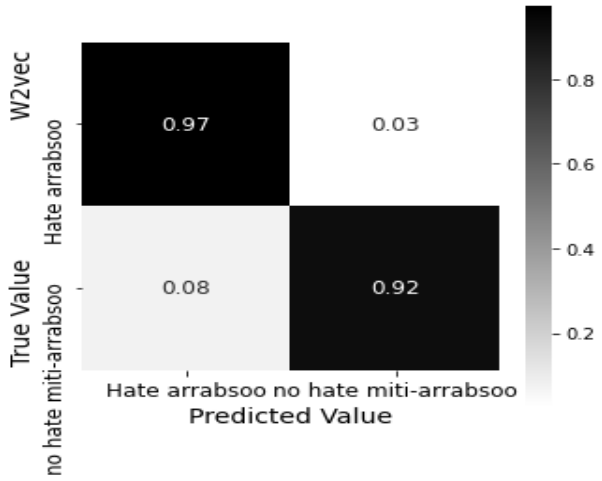

Figure 6.3 Word2vev Confusion Matrix classification evaluation

Figure 6.3 illustrates the sampled confusion matrix for SVM models. SVM with bigram classify 0.97% of Hate Speech and 0.92% of neutral correctly, and 0.03% of Hate Speech and 0.08 % of neutral are misclassified.

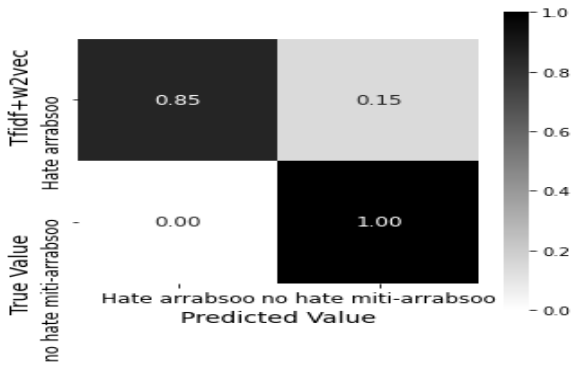

Figure 6.4 TF-IDF with w2vec Confusion Matrix classification evaluation

Figure 6.4 illustrates the sampled confusion matrix for SVM models. SVM with bigram classify 0.85% of Hate Speech and 100% of neutral correctly, and 0.15% of Hate Speech and 0.00 % of neutral are misclassified.

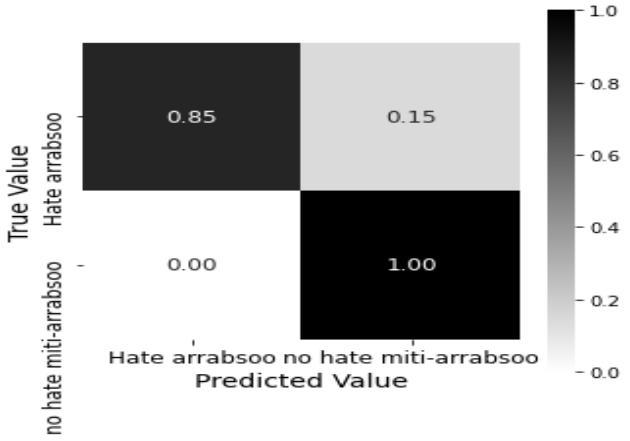

Figure 6.5 Sentiment analysis Confusion Matrix classification evaluation

Figure 6.5 illustrates the sampled confusion matrix for SVM models. SVM with bigram classify 0.85% of Hate Speech and 100% of neutral correctly, and also 0.15% of Hate Speech and 0.00 % of neutral are misclassified

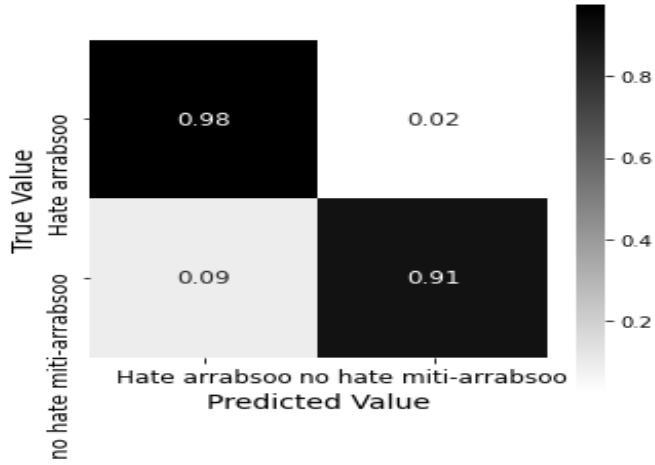

Figure 6.6 All Feature combination Confusion Matrix classification evaluation

Figure 6.6 illustrates the sampled confusion matrix for SVM models. SVM with bigram classify 0.98% of Hate Speech and 0.91% of neutral correctly, and 0.02% of Hate Speech and 0.09 % of neutral are misclassified.

6.2 Conclusion

This research intended to develop a tool to Hate Speech on social media using Machine learning techniques.

The study attempted to develop, implement, and compares Machine learning and text feature extraction methods specifically for Hate Speech detection for the Afaan Oromoo language. To enable successful execution of the research it was necessary to understand what Hate Speech is and its occurrence and manifestation on social media platforms, explore existing various techniques used to tackle the problem and understand the Afaan Oromoo language, as discussed in the literature. In addition, different methods followed to implement and design models that have the capability of detecting Hate Speech. These methods include collecting posts and comments for building the dataset, develop annotation guidelines, preprocessing, features extraction using n-gram, TF-IDF and word2vec models training using SVM models and testing. Finally, evaluation was performed for comparison. In this study, we manually annotated the posts and comments into two classes of Hate Speech (HS), and neutral Speeches. The total labeled dataset is 2784 instances of posts and comments with binary classes dataset. The experimental results showed that the trigram features used with the support vector Machine algorithm achieved 96% off overall evaluation metrics.

6.3 Contribution

The main contributions of this research work are as follow:

- Develop a labeled Hate Speech dataset for Afaan Oromoo language from social media.
- We develop standard Afaan Oromoo stopword lists.
- We develop a dictionary for short word expansion lists.
- We develop a SVM model for Afaan Oromoo text Hate Speeches.
- We tested our new model on Hate Speech detection and have achieved a competitive result.

At last, we Compare a SVM model with RF, NV, and LR Machine learning algorithms to check performance.

6.4 Future works

The proposed solution used a supervised learning algorithm with general feature or a text mining feature extraction method used to builds models. It is better to see the difference in performance results using unsupervised with specific features such as subject, and objective, augmentation and account characters.

This study is limited only on Afaan Oromoo language, but In Ethiopia, they are more than 80 languages are used in the country. For more compressive detection results, feature researchers can focus on developing multi datasets such as Amharic, Afaan Oromoo, Tigrigna, Somaligna, walayita and other language combinations. Moreover, comments and posts often contain video, images, emoji is which not a textual format is but it may contain Hate Speech. Therefore, feature researchers may also study how to include picture, emoji and video during Hate Speech detection.

Generally, researchers must consider below things for feature work.

- This work focused only on basic Hate Speech category as neutral speech and Hate speech further researchers could consider additional classes of Hate Speech category like racism, sexism, politics Hate, religious Hate, socio-economy which identify category of speech.

- To conduct the experiment, we have used a small amount of corpus to train the models. By increasing the dataset, quality and quantity improve the performance of the model.
- Detecting other forms of Hate Speech content on social media such as video, audio, emoji, and image containing Hate Speech.

References

[1] Z. N.Gitari, "A Lexicon-based approach for hate speech detection," *Int.J. Multimedia Ubiquit.eng,* vol. 4, pp. 215-230, 2015.

[2] H. M. Kaplan AM, "The challenges and opportunities of social media.," *International journal on media management,* vol. 1, p. 59–68, 2015.

[3] R. Cohen-Almagor, "Fighting Hate and Bigotry on the Internet.," *Policy & Internet,,* vol. 2, pp. 44-64, 2011.

[4] E. F. Unsvag, "Investigating the effects of user features," *Norwegian university of science and technology,* vol. 2, pp. 60-70, 2018.

[5] p. a. M. williams, "identifying cyber hate on Twitter across multiple protected characteristics," *EPJdata*

science, vol. 5, no. 1, 2016.

[6] Liu, "Sentiment analysis and subjectivity.," *In Handbook of Natural Language Processing Second Edition,* no. Chapman and Hall/CRC, pp. 627-666, 2010.

[7] M. B. a. T. O. H. Watanabe, "Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection," *IEEE Access,* vol. 6, p. 13825–13835, 2018.

[8] B. S. Biere, "Hate speech detection using natural language processing techniques," *Master Business AnalyticsDepartment of Mathematics Faculty of Science,* 2018.

[9] w. Hovy, "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection," *proceedings of NAACL_HLT,* pp. 88-93, 2016.

[10] B. G. a. Sikdar, "Using Convolutional Neural Networks to Classify Hate Speech," *Proc. ofthe First Work. Abus. Lang,Online,,* no. 7491, p. 85–90, 2017.

[11] "Facebook.," F8 2018:Using technology to remove the bad stuff before it's even reported., 2 may 2018. [Online]. Available: https://newsroom.fb.com/news/2018/05/removing-content-using-ai/. [Accessed 23 dec 2020].

[12] B. S. a. S. Bhulai, "Hate Speech Detection Using Natural Language Processing," *Vrije university Amsterdam,* 2018.

[13] z. tunu, "news:Ethiopia preparing new bill to curb hate speech," 23 nov 2018. [Online]. Available: https://www.addisstandard.com/news-ethiopia-preparing-new-bill-to-curb-hate-speech/amp/. [Accessed 4 jan 2020].

[14] "Social media Stats Ethiopia," [Online]. Available: https://gs.statcounter.com/social-media-stats/all/ethiopia. [Accessed 25 Oct 2019].

[15] M. F. a. P. Dave, "Facebook's flood of languages leave it struggling to monitor content," *Reuters,* vol. 23, 2019.

[16] "borkena.com," Ethiopia issued advisory on social media hate speech, misinformation dissemination, 19 mar 2019. [Online]. Available: https://borkena.com/2019/03/20/ethiopia-issued-advisory-on-social-media-hate-speech-misinformation-dissemination/. [Accessed 26 nov 2019].

[17] M. B. I. M. P. Z. B. A. Z. G. Aynekulu, "“MECHACHAL :Online debates and elections in Ethiopia.From hate speech to engagement in social media.,”," *Oxford & Addis Ababa university,* no. 10.2139/ssrn.2831369, 2016.

[18] A. P. a. M. P. I. Gagliardone, "Mapping and Analysing Hate Speech Online:Opportunities and Challenges for Ethiopia," *Oxford & Addis Ababa university/10.2139/ssrn.2601792,* 2014.

[19] S. M. A. S. C. T. V. Michele Corazza, "Comparing Different Supervised Approaches to Hate Speech Detection".

[20] Z. &. W. J. H. Mossie, "Social network hate speech detection for Amharic language.," . *Computer Science & Information Technology,* pp. 41-55, 2018.

[21] S. Z. J. Tan, "An empirical study of sentiment analysis for chinese documents.," *Expert Systems with applications,* p. 2622–2629, 2008.

[22] K. P. R. L. a. V. A. Soman, "Machine learning with SVM and other kernel methods," *PHI Learning Pvt. Ltd,* 2019.

[23] B. K. P. S. a. P. P. Premjith, "Amrita CEN@ FACT: Factuality Identification in Spanish Text.," *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019). CEUR Workshop Proceedings,CEUR-WS, Bilbao, Spain ,* 2019.

[24] W. W. a. J. Hirschberg, "Detecting hate speech on the world wide web," *In Proceedings of the Second Workshop on Language in Social Media,* pp. 19-26, 2012.

[25] B. F. L. W. J. H. a. C. R. Guang Xiang, "Detecting offensive tweets via topical feature discovery over a large scale twitter corpus," *In Proceedings of the 21st ACM international conference on Information and knowledge management,* pp. 1980-1984, 2012.

[26] B. V. H. D. Stephan Tulkens, "A Dictionary-based Approach to Racism Detection in Dutch Social Media," *ResearchGate,* 2016.

[27] P. B. a. M. L. Williams, "identifying cyber hate on twitter across multiple protected characteristics," *EPJ Data Science,* 2016.

[28] S. G. M. G. a. V. V. Pinkesh Badjatiya, "Deep learning for hate speech detection in tweets.," *In Proceedings of the 26th International Conference on World Wide Web Companion,* p. 759–760, 2017.

[29] W. T. a. D. TAMIRAT, "Investigating Afan Oromo Language Structure and Developing Effective File Editing Tool as Plug-in into Ms Word to Support Text Entry and Input Methods," *American Journal of Computer,* vol. 1, no. 008, p. 2, 2017.

[30] S. A. W. M., "A Survey on Hate Speech Detection using Natural Language Processing," *In Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media,Valencia, Spain, Association for Computational Linguistics:,* pp. 1-10, 2017.

[31] "Council of Europe," No hate speech movement, 16 Oct 2018. [Online]. Available: https://www.coe.int/en/web/no-hate-campaing.. [Accessed 27 Dec 2019].

[32] A. Hern., "Facbook ,youtube,twitter and microsoft sign eu hate speech code.," 31 may 2016. [Online]. Available: htts://www.theguardian.com/technology/2016/may/31/facebook-youtube-twitter-microsoft-eu-hate-speech-code. [Accessed 16 Sep 2019].

[33] "Transcript of Mark Zuckerberg's Senate hearing The Washington Post,," " Washington post, , 2018. [Online]. Available: https://www.washingtonpost.com/news/the-switch/wp/2018/04/10/transcript-of-mark-zuckerbergs-senate-hearing/?noredirect=on. . [Accessed 10 may 2012].

[34] O. D. Apuke, "Social and traditional mainstream media of communication:synergy and variance perspective," *international knowledge sharing* , vol. 53, p. 84, 2016.

[35] S. K. a. A. Joshi, "Hate Speech Detection from Code-mixed Hindi-English Tweets Using Deep Learning Models," vol. 1, 2018.

[36] H. L. a. M. C. C. Jefferson, "Fuzzy approach for sentiment analysis," in *IEEE international*, Naples, Italy, 2017.

[37] B. S. a. S. Bhulai, "Hate Speech Detection Using Natural Language Processing Techniques," *Vrije university Amsterdam,* 2018.

[38] S. &. B. Mondal, "A measurement study of hate speech in social media.," *In Proceedings of the 28th acm conference on hypertext and social media* , pp. 85-94, jul-2017.

[39] N. JT., "Hate Speech. Encyclopedia of the American Constitution.," vol. 3, p. 1277–79, 2000.

[40] "U. Nations and I. Introduction," *I.8. International Convention on the Elimination of All Forms of Racial Discrimination,Basic Doc. Int. Migr. Law,* vol. 7, no. 7, pp. 26-28, 2013.

[41] P. N. G.-P. A. M. de Gibert O, "Hate Speech Dataset from a White Supremacy Forum.," *2nd Workshop on Abusive Language Online @EMNLP,* 2018.

[42] W. D. M. M. W. I. Davidson T, "Automated Hate Speech Detection and the Problem of Offensive Language.," *ICWSM,* 2017.

[43] H. M. K. P. D. S. B. a. D. R. Saleem, "A web of hate:Tackling hateful speech in online social spaces," *ArXiv,* 2017.

[44] Z. M. a. J.-H. Wang, "social network hate speech detectionfor amharic language," *Department of International Graduate Program in Electrical Engineering and Computer Science,National Taipei University of Technology, Taipei, Taiwan* , p. 41–55, 2018.

[45] W. mekonnen., "Development of a stemming algorithm for Afaan Oromo text," *Master thesis at school of Information studies for Africa, Addis Ababa University,* 2000.

[46] A. Woldemariam., "Development of Morphological Analyzer for Afaan Oromo text,," *Master thesis at Faculty of Informatics, Addis Ababa University,* 2005.

[47] W. mekonnen., "Development of a stemming algorithm for Afaan Oromo text,," *Master thesis at school of Information studies for Africa,Addis Ababa University,* 2000.

[48] V. a. P. Kula, "Evaluation of Oromo-English Cross-Language Information Retreval.," *In IJCAI 2007 Workshop on CLIA,Hyderabad, India.* , 2007.

[49] E. Alpaydin., "Introduction to Machine Learning.," *The MIT Press,,* no. 2nd edition, 2010..

[50] Z. J. a. G. Xiaolin., "Comparison research on text pre-processing methods on twitter sentiment analysis.," *IEEE Access,,* vol. 5, p. 2870–2879, 2017,.

[51] L. H. E. L. B. V. a. W. D. S. Tulkens, "A Dictionary based Approach to Racism Detection in Dutch Social Media," *arXiv Prepr,,* no. arXiv:1608.08738, 2016.

[52] Z. Z. H. f. D. a. J. L. Njagi Dennis Gitari, "A lexicon-based approach for hate speech detection,," *International Journal of Multimedia and Ubiquitous Engineering,,* vol. 4, p. pp:215–230, 2015.

[53] L. A. M. M. D. C. F. B. a. I. W. Silva, "Analyzing the Targets of Hate in Online Social Media," *In ICWSM,,* pp. pp. 687-690,, 2016. .

[54] Z. a. D. H. Waseem, "Hateful symbols or hateful people? Predictive features for hate speech detection on twitter,," *In Proceedings of the NAACL student research workshop,,* pp. pp. 88-93,, 2016. .

[55] T. D. W. M. M. a. I. W. Davidson, "Automated hate speech detection and the problem of offensive language,," *arXiv preprint arXiv:1703.04009,,* 2017..

[56] S. a. A. S. Agarwal, "Characterizing Linguistic Attributes for Automatic Classification of Intent Based Racist/Radicalized Posts on Tumblr Micro-Blogging Website,," *arXiv preprint arXiv:1701.04931,,* 2017.

[57] Kwok and Wang, "Detecting Tweets against Blacks.," *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence,,* 2013.

[58] M.-W. C. K. L. a. K. T. Jacob Devlin, "pre-training of deep bidirectional transformers for language understanding.," *Computing Research Repository,* 2018.

[59] H. D. J. L. D. G. Z. Zuping, "A Lexicon-based Approach for Hate Speech detection," *Int. J. Multimed. Ubiquitous Eng,* vol. 10, no. 4, pp. 215-230, 2015.

[60] V. N. Vapnik., "The Nature of Statistical Learning Theory.," in *Springer-Verlag New York,* New York, NY, USA, , 1995..

[61] V.Vapnik, "The Nature of Statistical Learning Theory.," *NY: SpringerVerlag,,* 1995.

[62] H. a. Kamber, "Data mining: concepts and techniques," in *First edition, Morgan Kaufmann Publishers,* San Francisco,California., 2004.

[63] H. a. Kamber, "Data mining: concepts and techniques, Second edition,Morgan Kaufmann Publishers, San Francisco,," *California. ,* 2006.

[64] J. Martin, "Speech and Language Processing(draft)," *URL https://web. Stanford. edu/~ jurafsky/slp3.,* 2019.

[65] D. J. a. J. H. Martin., "Speech and Language Processing:," *An introduction to natural language processing, computational linguistics,and speech recognition.,* 2017..

[66] F. J. O. a. D. Roggen., "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition.," in *Sensors, ,* 2016., p. 16(1)..

[67] F. Harrell, "Damage caused by classification accuracy and other discontinuous improper accuracy

scoring rules," 2017. [Online]. Available: http://www.fharrell.com/post/class-damage/.. [Accessed 3 march 2020].

[68] Y. Sasaki, "The truth of the f-measure," *Teaching, Tutorial materials,* 2007.

[69] S. G. T. a. K. K. Tune, "researchsquare," Automated Amharic Hate speech Posts and Comments Detection Model using Recurrent., [Online]. Available: https://www.researchsquare.com/article/rs-114533/v1. [Accessed 19 01 2021].

[70] B. ZEMEDKUN, "Hate Speech and Disinformation Prevention and Suppression," *SCHOOL OF LAW (Doctoral dissertation,. University of Gondar.,* 2019.

[71] M. a. Wang, "Vulnerable community identification using hate speech detection on social media.," *Information Processing & Management,,* vol. 57, no. 3, 2020.

[72] M. Oljira, "Sentiment Analysis of Afaan Oromo using Machine learning approach," *International Journal of Research Studies in Science, Engineering and Technology,* vol. 07, no. ISSN, pp. 07-15, 2020.

[73] T. W. F. S. Marian-Andrei Rizoiu, "Transfer Learning for Hate speech detection in social Media," 15 may 2019. [Online]. Available: https://dictionary.cambridge.org/dictionary/english/hate-speech. [Accessed 28 Feb 2020].

[74] S. L. a. T. Forss, "New classification models for detecting Hate and Violence web content," *7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K),* pp. 487-495, 2015.

[75] H. A. M. M. S.-g. J. A. K. J. J. Joni Salminen, "Developing a Taxonomy and Machine Learning Models for Identifying and Classifying Hate in Online News Media," *Proceeding of the Twelfth Internonational AAAI Conference on Web and Social media* , no. anatomy of online Hate , 2018.

[76] H. L. a. P. Burnap, "Fuzzy Multi-task Learning for Hate Speech Type Identification," Cardiff, United Kingdom, May 2019.

[77] A. P. a. M. P. Iginio Gagliardone, "Mapping and Analyzing Hate Speech Online Opportunities and challenges for Ethiopian," 2014.

[78] S. a. T. F. Liu, ""New classification models for detecting Hate and Violence web In Knowledge Discovery engineering and knowledge management (Ic3K)," vol. 1, pp. 487-495, 2015.

[79] J. R. M. M. G. V. R. a. N. Nemanja, "Hate speech detection with comment embeddings," *In Proceedings of the 24th International Conference on World Wide Web,* p. 29–ACM2, 2015.

[80] P. B. K. Sreelakshmi k, "Detection of Hate Speech Text in Hindi-English Code-mixed Data," *Third International Conference on Computing and Network Communications,* vol. 1, pp. 36-41, 2018.

[81] D. W. M. M. a. I. W. T. Davidson, "Automated hate speech detection and the problem of offensive language," *in Proceedings of the 11th International AAAI Conference on Weblogs and Social Media,*

no. ser ICWSM , 2017.

[82] J. Z. R. M. M. G. V. R. N. B. N. Djuric, "Hate Speech Detection with Comment Embeddings," *Proceedings of the 24th International Conference on World Wide Web,* no. Italy doi>11145/2740908.274276 , 2015.

[83] A. C. F. D. M. P. M. T. F. D. Vigna, "Hate me, hate me not:Hate speech detection on facebook," *In Proceedings of the First Italian Conference on Cybersecurity,* p. 86 – 95, 2017.

[84] S. G. M. G. V. V. P. Badjatiya, "Deep Learning for Hate Speech Detection in Tweets," *in Proceedings of the 26th International Conference on World Wide Web Companion,* no. Australia [doi>11145/3041021.3054223], 2017.

[85] R. H. L. Gao, "Detecting online hate speech using context aware models," *In Proceedings of the International Conference Recent Advances in Natural Language Processing,RANLP 2017,* no. INCOMA Ltd, p. 260–266, 2017.

[86] J. T. A. T. Y. M. Y. C. Chikashi Nobata, "Abusive language detection in online user content.," *In Proceedings of the 25th international conference on world wide web,International World Wide Web Conferences Steering Committee,* p. pp.145–153., 2016.

[87] W. a. J. Hirschberg, "Detecting hate speech on the world wide web," *In Proceedings of the Second Workshop on Language in Social Media,Association for Computational Linguistics.,* pp. 19-26, 2012.

[88] C. D. G. S. C. Y. S. N. Sap M, "The risk of racial bias in hate speech detection," *In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics,* p. 1668–1678, 2019.

[89] R. M. C. G. C. B. K. N. W. M. Ross B, "Measuring the Reliability of Hate Speech Annotations: The Case of the European Refugee Crisis.," *3rd Workshop on Natural Language Processing for Computer-Mediated Communication @ Conference on Natural Language Processing,* 2016.

[90] R. a. J. L. Magu, " Detecting the hate code on social media.," *arXiv preprint arXiv:1703.05443.,* 2017.

[91] W. J. W. A. A. R. HAGIT SHATKAY, "Annotation Guidelines," *School of Computing, Queen's University, Kingston,* 2005.

[92] I. G. a. N. Joshi., "Tweet normalization: A knowledge based approach.," *International Conference on Infocom Technologies and Unmanned Systems* , p. pages 157–162, Dec2017..

[93] M. B. a. T. O. Hajime Watanabe, "Hate speech on twitter a pragmatic approach to collect hateful and offensive expressions and perform hate speech detection.," *IEEE Access,,* pp. 2169-3536., 2018. .

[94] T. Debela, "Design a stemmer for Afan Oromo text: Ahybrid approach," *Internation journal of computational linguistics,* vol. 1, no. 2, p. 6, 2010.

[95] M. Gasser, "a system for morphological processing of Amharic, Oromo, and Tigrinya," *Indiana University,* 2012.

[96] [Online]. Available: http://www.cs.indiana.edu/~gasser/Research/software.html. [Accessed 07 04 2021].

[97] L. H. E. L. B. V. a. W. D. S. Tulkens, "A Dictionary based Approach to Racism Detection in Dutch Social Media," *arXiv Prepr* , no. no.arXiv:1608.08738, 2016.

[98] R. &. S. Manning, "An Introduction to Information Retrieval.," *Cambridge University Press.* , 2009.

[99] P. Yang, "A Comparative Study on Feature Selection in Text Categorization.," *International Conference on Machine Learning*, pp. 412-420, 1997.

Appendices

Appendix 1: List of some of Afaan Oromoo short words and their Expansion

Qar	Qarshii
A.L.A	Akka lakkoofsa Awuroopa
A.L.I	Akka Lakkoofsa Itoophiyaa
Bill	Billiyoona
Mill	Milliyoona
Ykn	Yookiin/yookaan
KKf(K.K.F)	Kan kana Fakkaatan
Prof	Profeesara
M/B	Mana Barumsaa
Fkn	Fakkeenya

Dr	Dooktoora
I/G	Itti Gaafatamaa
M/murtii	Mana Murtii
Hosp	Hoospitaala
M/ministeeraa	Muumee Ministeeraa
Lakk	Lakkoofsa
Dh.K.D	Dhaloota Kiriistoosin Dura
Dh.k.B	Dhaloota Kiriistoosin Booda
Hogg	Hooganaa
H/Bulaa	Hoorsiisee Bulaa
Q/Bulaa	Qonnaan Bulaa

Appendix2: List of some Afaan Oromoo stopwords

Aanee	Akkam	Akkuma	Alatti	Sitti	Tanaaf	Ta'ullee	Utuu	Warra	yommuu
Gidduu	Hanga	Hogguu	Immoo	Ammo	Ani	Bira	Booddee	Duuba	Eegasii
Itti	Jala	Sana	Kana	Irraa	Isaaf	Isatti	Iseen	Ishiif	Isii
Narraa	Nu	Nuti	Siin	Kee	Keenna	Keessan	Keessatti	Koo	Malee
Akka	Akkasumas	Ala	Amma	Tanaaf	Tanaafuu	Teenya	Waan	Yeroo	Yoo
Gubbaa	Henna	Illee	Inni	An	Ati	Booda	Dura	Eega	Fi

Ittuu	Jara	Kan	Kanaaf	Isaa	Isaanirraa	Tun	Ishii	Ishiirraa	Isin
Natti	Nurra	Siin	Kanaafi	Kee	Keessa	Keenya	kiyya	kun	Na
yookaan	gama	isiin	naaf	yoom	kun				