Hate Speech Detection for Amharic Language on Social Media Using Machine Learning Techniques

By: Yonas Kenenisa Defar

A Thesis Submitted to the Department of Computing

School of Electrical Engineering and Computing.

Presented in Partial Fulfillment for the Degree of Masters of Science in

Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

Adama, Ethiopia,

September 2019

Hate Speech Detection for Amharic Language on Social Media Using Machine Learning Techniques

By: Yonas Kenenisa Defar

Advisor: Tilahun Melak (PhD.)

A Thesis Submitted to the Department of Computing

School of Electrical Engineering and Computing.

Presented in Partial Fulfillment for the Degree of Masters of Science in

Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

Adama, Ethiopia,

September 2019

Declaration

I hereby declare that this MSc thesis is my original work and has not been presented as a partial degree requirement for a degree in any other university and that all sources of materials used for the thesis have been duly acknowledged.

Name: Yonas Kenenisa

Signature:_____

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

Name: Tilahun Melak (PhD.)

Signature:_____

Date of Submission:_____

Approval of Board of Examiners

We, the undersigned, members of the Board of Examiners of the final open defense by Yonas Kenenisa have read and evaluated his/her thesis entitled "Hate Speech Detection for Amharic Language on Social Media Using Machine Learning Techniques" and examined the candidate. This is, therefore, to certify that the thesis has been accepted in partial fulfillment of the requirement of the Degree of Masters of Science in Computer Science and Engineering.

Supervisor /Advisor	Signature	Date
Chairperson	Signature	Date
Internal Examiner	Signature	Date
External Examiner	Signature	Date

Acknowledgment

I would like to acknowledge God for his grace, strength and good health as I undertook this research. My sincere gratitude to my advisor Dr. Tilahun Melak 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 also like to acknowledge the members of Smart Software SIG of the CSE department. Including the SIG supervisor Dr. Mesifn Abebe for his continued commitment to guide and support this research from its inception to completion. And to all members who sat on each phase presentation panels for their inputs which greatly shaped and improved the 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. Mohammed Awel 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. Mulualem Mussie, Menelik Sahalu, and Abdiwak Tesema for their major role in the dataset annotation process as annotator and for anyone who participated or contribution to this work.

Last but not least, my thanks go to my wonderful wife and parents for their every support and advice in my life. May God always Bless and keep them safe.

Table of Contents

Contents

List of Tables

List of Figures

List of Equations

List of Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
AUC	Area Under the Curve
Avg	Average
BLR	Bayesian Logistic Regression
BOW	Bag of Words
CBOW	Continuous Bag of Words
CEO	Chief Executive Officer
CNN	Convolutional Neural Networks
CPU	Central Processing Unit
CV	Cross-Validation
EU	European Union
F1	F1-Score
FN	False Negatives
FP	False Positives
HS	Hate Speech
KNN	K-Nearest Neighbors
LR	Logistic Regression
LSTM	Long Short-Term Memory
ML	Machine Learning
NB	Naïve Bayes
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NH	Non-Hate
OFS	Offensive Speech
P	Precision
POS	Part of Speech
R	Recall

RE	Regular Expression
RF	Random Forest
RFDT	Random Forest Decision Tree
RNN	Recurrent Neural Network
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
TN	True Negatives
TP	True Positives
UN	United Nations
URL	Uniform Resource Locator

Abstract

Hate speech on social media has unfortunately become a common occurrence in the Ethiopian online community largely due to the substantial growth of users on social media in recent years. Hate speech on social media has the potential to quickly disseminate through the online user that could escalate into an act of violence and hate crime on the ground. Determining a portion of a text containing hate speech is not simple tasks for humans it is time-consuming and introduces subjective notions of what constitutes a text to be hate or offensive speech.

As a solution to this problem, this research proposed hate speech detection using machine learning and text-mining feature extraction techniques to build a detection model. A hate speech data was collected from the Facebook public page and manually labeled into three classes and then converted into binary class to build binary and ternary datasets. The research employed an experimental approach to determine the best combination of the machine learning algorithm and features extraction for models. SVM, NB, and RF models trained using the whole dataset with the extracted feature based on word unigram, bigram, trigram, combined n-grams, TF-IDF, combined n-grams weighted by TF-IDF and word2vec for both datasets. The models evaluated using 5-fold cross-validation, and classification performance were used to compare the models.

Using the two datasets the study developed two kinds of models with each feature those are binary models and ternary models. The models based on SVM with word2vec achieve slightly better performance than the NB and RF models for both binary and ternary models. According to the classification performance result, the ternary models achieve less confusion between hate and non-hate speech than the binary models. However, the models tend to misclassify offensive speech as hate speech. Generally, hate speech detection with the machine learning and text feature extraction method based on multi-class dataset achieves a better performance than the binary class detection models.

Key Words: Amharic hate speech detection, offensive speech, Amharic posts and comments datasets, machine learning classifier

CHAPTER ONE

1 Introduction

1.1 Background

Social media is changing the face of communication and culture of society around the world. In Ethiopia, the user of social media has grown substantially in recent years, even though the low quality of the internet services and sometimes interruptions or blocking of social media sites in the country. As people in the country are using the internet for social interactions on online social media to communicate, express opinions, interact with others, and to share information. This leads to an increase in hateful activities that exploit such sites. These social media's anonymity and mobility features enable individuals to hide behind a screen and spread the hateful content without effort or consequence[1], [2].

Furthermore, social media 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 [3]. Government worldwide are passing hate speech regulation and pressuring social media companies to implement policies to stop the spread of online hate speech [4]. The government of Ethiopia monitors content on social media to prevent harmful rhetoric messages and govern online hate speech through many time interruptions of the internet service and also by blocking these sites form being access in the country[5], [6]. Also, the government introduces a law to encompass the online space through the expansion of anti-terrorism laws, and the law prohibits “the use of any telecommunication network or apparatus to disseminate any terrorizing message” or “obscene message,” subjecting violators to a prison sentence of up to eight years [6].

Deciding if a portion of text contains hate speech is not simple tasks, even for humans[7], [8]. The human moderation of hate speech is not only time consuming but also introduces subjective notions of what constitutes hate speech. Therefore, it is crucial to clearly define hate speech to make it easier to outline a rule for the annotation process of the dataset for annotator and automatic detection models evaluation. The majority of the social media and research study define, Hate Speech as a language that attacks or diminishes, that incites violence or hate against groups, based on specific characteristics such as race, ethnic origin, religious affiliation, their political view,

physical appearance, gender or other characteristics. The definition points out that hate speech is language incite violence or haters toward groups. Also, there is an acknowledgment that hates speech on social media has a high probability of relating to actual hate crime[9], [10]. However, there are other types of speech which their definition is similar to hate speech, but the level or the effect is not similar. One of these speeches is an offensive speech is a language used to offend someone. The difference between hate speech and other offensive language is often based upon indirect verbal dissimilarities[11].

Social media currently provide localization, which allows the user to use different world languages on their sites. One of these languages is Amharic, Amharic languages are one of wildly spoken language and working language of the federal government of Ethiopia. The language is written left-to-right and has its unique script, which lacks capitalization and in total 275 characters, mainly consonant-vowel pairs. It is the second-largest Semitic language after Arabic and spoken by about 40% of the population as the first or second language[1], [12]. The current estimated population is 107.53 million. The Amharic language is still under-resourced that have few computing tools, and hate or offensives speech detection tools or research study that propose a solution. As far as we know, the study of hate speech detection in the Amharic language is still very rare that we found only one previous work [1] on this subject for computer science field but there are few social sciences field research work that studies the online hate speech in an election period in the country [5], [6].

Nowadays, in Ethiopia, it is an open secret that the recent widespread hate speech and call for violence and attacks on particular targets of individual or group based on their political view, ethnic origin, and religious affiliation. Therefore, it is important to monitor or automatically detect hate speech on this platform to prevent their spread, and possible reduce acts of violence and hate crimes that destroy the lives of individuals, families, communities, and also the country.

This research work focuses on addressing the problem of hate speech using a new dataset that is annotated with three labels: Hate, Offensive, and Neither hate nor offensive (OK). Most Previous work considers binary class approach to solve hate speech problem, which leads to mix-ups of hate with offensive language and other types of speech, but they should not be mixed with each other because if someone using offensive language is not automatically hate speech as definition

stated, people tend to use term that is technically speaking highly offensive, for all sort of reasons, example in-joke, criticism, debate and also condemnation to make positive point[11]. So, the binary classification of post and comment as hate and non-hate leads to the conclusion of considering many people on social media to be hateful people.

Finally, this research study proposed hate and offensive speech detection models for the Amharic language by using a new dataset of posts and comments from the Facebook public page and also using multiple feature extraction methods and multiple machine learning classifier algorithms to compare each method and select a better a detection model.

1.2 Motivation

The researcher, as a social media user, observed that these sites are becoming easy tools for propagating hateful content that harms individuals and groups. Also, a 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 of this problem. The social media company like Facebook and twitter are straggling on these issues of automatic detection. In fact, the Facebook CEO said on his testimony to US Congress on April 10 2018, “we get wrong and right on hate speech because we don’t have adequate AI tools now for automatic detection hate speech, but we will have an AI to take the primary role in automatically detecting hate speech on Facebook in 5 to 10 year”[13]. Generally, hate speech remains a sensitive process that user needs to flag hate speech to social media platform for it to be manually reviewed or remove(deleted) from the platform based on their policies, this particular process is difficult to handle by human because users communicate with many different languages.

Recently in Ethiopia, the use of social media widely increased this because of the availability of social media platform in the country, which leads to misuse of this platform. The researcher inspired to study hate speech detection for the Amharic language on social media because detecting online hate speech is one of the important tasks to tackle actual hate crime on the ground. Also, to contribute to the development of better detection system and reduces the lack of hate speech dataset for future research.

1.3 Statement of the Problem

The hate speech detection mechanism is far from perfect to be able to detect hate on social media. Because social media consists of a large amount of user-generated content that would need to be monitored in order to detect hateful activities. Online hate speech may lead to the actual hate crime that destroys lives and communities on the ground. On the one hand, research on detecting hate speech for the Amharic language is rare we found only one research previous study by Mossie and Wang [1], which use a binary class dataset for hate speech detection model. Considering hate speech as a binary problem leads to non-hate speech that may contain offensive language to misclassified as hate speech. If we combined hate speech with other types of speech like offensive speech, then mistakenly consider many people on social media to be hateful.

On the other hand, most of the research study on detecting hate speech conducted for English language and few others language like Dutch [14], Indonesian [15], [16], and Italian[17][18] focusing on particular hate speech issue such as racism, sexism, anti-Semitism, and cyberbully, etc. This study addresses these problems by building a new Amharic dataset, utilizing multiple machine learning and featured extraction method; this study addressed the problem by answering the following questions:

- What are the effective ways to differentiate hate speech from offensive speech?
- How hate speech detection mechanisms can be improved?
- How to implement high-performance machine learning classifiers and feature modeling to accurately detect Amharic hate speech text?

1.4 Objective

1.4.1 General Objective

The main objective is to develop a machine learning model that can detect hate and offensive speech for the Amharic language on social media.

1.4.2 Specifics Objectives

- ✓ To build the datasets using Amharic posts and comments from Facebook.
- ✓ To develop annotation guidelines for labeling posts and comments.

- ✓ To identify effective features extraction and machine learning algorithms for hate and offensive speech detection.
- ✓ To build detection models using machine learning algorithms.
- ✓ To recommend the best detection model for Amharic hate speech by evaluating performance and accuracy.

1.5 Scope and Limitations

1.5.1 Scope

The study focuses only on detecting hate speech for the Amharic language by using machine learning. A new dataset that is built by collects Amharic text posts and comment from Facebook popular public pages for April 2018 to April 2019 and annotating the post or comment into three different classes, which are hate, offensive, and neither speeches. Also, the binary class that converts all the offensive to hate for comparison of classifications models results. The study implements machine learning classifiers for the detection model and evaluates the result of each classifier based on the classification accuracy of the K-fold cross-validation technique.

1.5.2 Limitations

This study comes across to limitations on a different phase of the research process. Since there is a lack of other studies for comparison hate speech detection for the Amharic language, also lack 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 lack of standard Amharic language stemmer and stopwords lists. The study did not apply these two preprocessing methods.
- Due to the tight schedule, the study only implemented the proposed machine learning classifier and limited it to develop and evaluate algorithms.

1.6 Application of Results

The main beneficiaries of this study are any stakeholders that use social media platforms for their day to day activities. On the one hand, Social media platform benefit by taking this work results as input for developing better hate speech detection or monitoring models for the Amharic language on their platform, because they are straggling for developing hate speech detection or hate content moderation system that can support most language used on the platform. Also, it helps the user to be protected from hate speech during the time they spent on this social media platform. Additionally, researchers can replicate the proposed research for other languages used in Ethiopia and can serve as a baseline for related work on this issue or use the dataset to improve the research.

1.7 Thesis Organization

The research paper organized under seven chapters in the following ways:

Chapter one discussed above, and It includes the introduction of the study, the motivation, and the statement of problems, the research questions that would be answered in the proposed solutions, the scope and limitation, the objective of the study, the methodology, and the application results of the study.

Chapter two, presents the literature review and related works on automation hate speech detection, definition of hate and offensive speech, hate speech on social media and Ethiopia, Methodologies use in hate speech detection: feature extraction and machine learning classifier, it discussed overview of the Amharic language and finally, it discussed the related work.

Chapter three, discusses the research methodology used in this research work, methods used to build the dataset, methods used to develop Amharic hate and offensives speech detection models which are Amharic text preprocessing, feature extraction methods, and machine learning classifier algorithm's, and finally, it presents the evaluation methods.

Chapter four discusses the proposed solution for automatic Amharic hate speech detection, which is the proposed Amharic text preprocessing, feature extraction, machine learning training, and testing.

Chapter five discusses the implementation and experimentation of the proposed solution. Including working environment, dataset description, also the implementation of preprocessing, feature extraction implementation, models implementations, and evaluation models.

Chapter six discusses the result of the dataset annotation process, feature extraction, binary, and ternary models and also discusses the major results obtained by comparing both models based on features.

Chapter seven presents conclusion, recommendation and identifies future works for further research direction on this research topic

CHAPTER TWO

2 Literature Review and Related Works

This chapter reviews relevant literature to further comprehend the concept and investigate the research problem. Which provides a technical review on hate and offensive speech and existing techniques used to detect. Starting by defining the term hate and offensive speech, and followed by a review of hate speech on social media, hate speech in Ethiopia, characteristics of Amharic language, and techniques that have been used in hate speech detection and related work previously studied on the problem.

2.1 Hate Speech

There is no single agreed-upon definition of the term 'hate speech' for online or offline. The topic has hotly debated by academics, legal experts, and policymakers. There are many examples of the term 'hate' being used to describe speech that constitutes a slur, toxic language or an instance of verbal abuse against a wide range of targets, in a way that does not distinguish 'hate speech' from other types of speech such as offense or insult [19]. The definitions of hate speech form Different sources list below, these sources are social network platforms, governmental organizations, non-governmental organizations, UN, and scientific communities. The following are a list of definitions and sources of definitions:

- **Definition 1: 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."[20].
- **Definition 2: The European Court of Human Rights**, "hate speech shall be understood as covering all forms of expression which spread, incite, promote or justify racial hatred, xenophobia, anti-Semitism or other forms of hatred based on intolerance, including intolerance expressed through aggressive nationalism and ethnocentrism, discrimination and hostility against minorities, migrants and people of immigrant origin"[21].
- **Definition 3: Ring, Caitlin Elizabeth 2013**, "hate speech is all forms of expression that spread, incite, promote or justify racial hatred, xenophobia, anti-Semitism or other forms of

hatred based on intolerance, including intolerance expressed by aggressive nationalism and ethnocentrism, discrimination and hostility against minorities, migrants and people of immigrant origin."[22]

- **Definition 4: Facebook**, hate speech is "Objectionable content that direct attack on people based on what we call protected characteristics such as race, ethnicity, national origin, religious affiliation, sexual orientation, caste, sex, gender, gender identity, and serious disease or disability. We also provide some protections for immigration status. We define attack as violent or dehumanizing speech, statements of inferiority, or calls for exclusion or segregation." [23]
- **Definition 5: 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 serious disease. We also do not allow accounts whose primary purpose is inciting harm towards others based on these categories." [24]
- **Definition 6: YouTube,** "Hate speech is not allowed on YouTube. We remove content promoting violence or hatred against individuals or groups based on any of the following attributes: Age, Disability, Ethnicity, Gender, Nationality, Race, Immigration Status, Religion, Sex, Sexual Orientation, Veteran Status. We refer to Gender, Sex, and Sexual Orientation definitions mindful that society's views on these definitions are evolving." [25]

The majority of the sources and other entities have similarities between their definitions of hate speech. All the above definitions of hate speech can be summarized in the following three points. This summarization of content analysis of the definitions is adopted from Fortuna et al. Survey work on hate speech detection [19].

- **Hate speech has specific targets**: all the definitions point out that hate speech has specific targets and it is based on specific characteristics of groups, like ethnic origin, religion, or other.
- **Hate speech is to incite violence or hatred:** the majority of the definitions point out that hate speech is to incite violence or hate towards a minority
- **Hate speech is to attack or diminish**: some of the definitions state that hate speech is to use language that attacks or diminishes these groups

The definitions can be summarizing the content to define hate speech for this research work.

***Hate speech**:* is language and expression or kind of writing that attacks or diminishes, that incites violence or hate against groups based on specific characteristics such as race, ethnic origin, religious affiliation, political view, physical appearance, and gender.

Additionally, this definition has been used in most of the previous related research work which also points that, hate speech is a complex phenomenon, intrinsically associated to relationships between groups, and also relying on language nuances [1], [4], [8], [11], [16], [26], [27].

2.2 Offensive Speech

Offensive speech can be defined as a language or expression that offends and negatively characterizes an individual or group of people. This kind of speech causes someone to feel upset, annoyed, insulating, angry, hurt, and disgusting.

The difference between hate speech and offensive language often based upon understated linguistic distinctions [11]. Offensive speech occurs when there is a low degree of hate speech characterizations occur. The speech may contain a target but do not direct incite violence, attack or diminish based on the target group characters because people often use offensive language to make a point in the debate, on heated conversation, and condemn another violent act performed by others people. Any speech that contains a sarcastic, mocking, and joke that offend can be considered offensive if it conducts using language contains high negative, abusive, dirty words or phrases in both oral or text.

2.3 Hate Speech on Social Media

The Internet is one of the greatest innovations of mankind, which has brought together people from every race, religion, and nationality. Social media sites such as Twitter and Facebook have connected billions of people and allowed them to share their ideas and opinions instantly. That being said, there are several hostile consequences as well such as online harassment, trolling, cyber-bullying, and hate speech[10]. A social media nature makes a perfect place for peoples to create and share content or to participate online. These online platforms are often exploited and misused to spread content that can attacks or diminishes, that incites violence or hate against

groups or individual. Because of the anonymity and mobility of social media platform features for security and privacy, enable their users to hide their true identity behind the screen and express or spread hateful content than they might otherwise.

2.4 Hate Speech in Ethiopian

Hate speech in Ethiopia is growing with the increase of social media users in the country. The widespread hate speech using social media is an open secret. However, there is no Legal law or recommendation that indirectly define or tackle hate speech. However, there is a law that indirectly used for hate-related issues, which is anti-terrorism law, the law prohibits "the use of any telecommunication network or apparatus to disseminate any terrorizing message" or "obscene message", this includes a use of any social media platform and other forms of commutation platform to disseminate terrorizing message. The subjecting violators to a prison sentence of up to eight years[6]. However, the law has been used to restrict messages or speech criticizes government policy and officials. This law has got backlash from the national and international organization and academic community because of the use of the law contradict freedom of speech of human right law[5]. Currently, law-maker, government officials, and politicians in Ethiopia have been aware of hate speech on social media, and they are looking for a solution to tackle this problem. The country lawmakers are drafting new hate speech and fake news law which will be law soon.

2.5 Hate Speech Detection Techniques

Hate speech detection problem has been studied by a different researcher, and they used different techniques to detect hate speech propagation on social media and other online web platforms. Just as there is no clear agreement on the definition of hate speech, there is no consensus concerning the most effective methods to detect on social media platforms.

The majority of automated approaches to identifying hate speech begin with a binary classification task in which researchers concerned with labeling a post, tweet, or comment as 'hate speech or not,'[1], [15], [16], [27]. Also, then, multiclass approaches have been used to detect hate speech with labeling of post, tweet or comment as 'Hate' and 'Offensive' and 'Clean', and most multiclass classification of hate speech based on general hate speech, racism, sexism, religion, anti-Semitism, nationality, politics [2], [11], [14], [26], [28], [29]. The vast majority of studies examine English

language content, though some researchers have developed methods to detect hate speech in other languages. These include pragmatic examinations of hate speech in Amharic[1], Indonesian[16], [30], Arabic[31],[9], Italian[18], Dutch[14], and German[32]. Almost all the previous research reviewed for this research work uses text mining feature extraction and machine learning approach to detect hate speech.

2.5.1 Feature Extraction Used in Hate Speech Detection

Features extraction used in hate speech detection study can be dividing into two categories a text mining feature extraction used for hate speech detection and specific features for hate speech detection. The majority of the research papers adopt strategies already known in text mining to the specific problem of automatic detection of hate speech[9], [19], [33]. Also, few research papers use specific hate speech detection features used to tackle the problem of hate speech automatic detection.

2.5.1.1 Text Mining Features Used for Hate Speech Detection

This section presents and discusses the features extraction commonly used in text mining approach that is applied for hate speech detection on social media studies.

Bag-of-words (BOW): It is a way to extract features from the text for use in algorithms for machine learning. Text in a corpus is depicted as the bag (multiset) of its phrases in this technique of extraction of features, disregarding grammar, and even word sequence but maintaining multiplicity. The function is developed based on the term in the corpus. The BOW technique indicates the number of occurrences in the specified text for each word. Also, used to train a classifier. The drawbacks of this type of approach are that the word sequence and its syntactic and semantic content are ignored. Therefore, it can lead to misclassification if the words are used in different contexts [16], [30], [34].

Dictionaries-Based: Using dictionaries is one of the easiest methods in text mining. dictionary-based approaches involve developing a list of words that are searched and counted in a text. These methods can also involve normalizing or taking the total number of words in each text into consideration. The approaches generally used to identify hate speech by creating a dictionary of insults and slurs word as a feature [14], [29].

N-grams: is a word prediction model using probabilistic methods to predict the next word after observing N-1 words in a text. This feature extraction approach consists in combining sequential words into lists with size N. Simply, N-grams are all combinations of adjacent words or characters of size N that is found in the text in a document. This method allows improving classifiers' performance than BOW because it incorporates, to some degree, the context of each word. Instead of using words it is also possible to use N-grams with characters or syllables. Character N-gram features proved to be more predictive than token N-gram features for the specific problem of abusive language detection [35]. N-grams are one of the most used techniques in hate speech automatic detection and related tasks [3], [7], [11], [30], [36], [37],[38].

TF-IDF (Term Frequency-Inverse Document Frequency) 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. The TF-IDF is composed of two terms: The first computes the normalized Term Frequency (TF), which is. The number of times a word appears in a document, divided by the total number of words in that document;

$$TF(t) = \frac{Number\ of\ times\ term\ t\ appears\ in\ a\ document}{Total\ number\ of\ terms\ in\ the\ document} \text{ ---------- (2.1)}$$

The second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears[39].

$$IDF(t) = \frac{\log\_e\ (\text{Total number of documents})}{\text{Number of documents with term t in it}} \text{ ---------- (2.2)}$$

It is distinct from a bag of words, or N-grams 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. TF-IDF is also one of the feature modeling methods used in hate speech detection.[1],[11], [26],[37].

Word2Vec: is a type of mapping that allows words with similar meaning to have similar vector representation. It utilizes either of two model architectures to produce a distributed representation

of words of the corpus: CBOW or skip-gram. In the continuous Bag of Word (CBOW) architecture, the model predicts the current word from a window of surrounding context words. The order of context words does not influence prediction. In continuous skip-gram architecture, the model uses the current word to predict the surrounding window of context words. The skip-gram architecture weighs nearby context words more heavily than more distant context words. Word2vec input is a text corpus, and its output is a set of vectors: feature vectors for words in that corpus. Word2vec is not a deep neural network, but it turns text into a numerical vector for training both supervised machine learning or deep learning algorithm. It is one of word embedding techniques that most previous related research used on hate speech detection work[1], [3], [4], [40], [41].

Part-of-speech (POS): approaches 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-3rd person present singular form (VBP), Adjectives (JJ), Determiners (DT), Verb base forms (VB). Part-of-speech has also been used in hate speech detection problems [11].

2.5.1.2 Specific Features for Hate Speech Detection

Complementary to the approaches commonly used in text mining feature extraction, several specific features are being used to tackle the problem of hate speech automatic detection.

Objectivity and Subjectivity of the Language: a study on [29], the authors argue that hate speech is related to more subjective communication. A rule-based approach is used to separate objective sentences from subjective ones and, after this step, remove the objective sentences from the analysis.

Othering Language: "Othering" is a term that not only encompasses the many expressions of prejudice based on group identities, but it provides a clarifying frame that reveals a set of common processes and conditions that propagate group-based inequality and marginality. Othering has been used as a construct surrounding hate speech and consists in analyzing the contrast between different groups by looking at "Us versus Them." It describes "Our" characteristics as superior to "Theirs," which are inferior, undeserving, and incompatible. Expressions like "send them home"

show this cognitive process[42]. Typed dependencies provide a representation of syntactic grammatical relationships in a sentence.

Focus on Particular Stereotypes: In the studies by Warner et al. [36], the authors hypothesize that hate speech often employs well-known stereotypes. A stereotype is an over-generalized belief in a specific category of people. Therefore, subdivide such speech according to the stereotypes is useful because each stereotype has specific language words, phrases, metaphors, and concepts related to that specific stereotype.

2.5.2 Machine Learning for Hate Speech Detection

Most state-of-the-art hate speech detection techniques involve machine learning algorithms for classification. The machine learning approaches often rely on feature extraction techniques presented in section 2.3.2 above. After preparing the dataset to work with machine classification algorithms can take a place to perform the detection task.

2.5.2.1 Machine Learning

Machine learning 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 focuses on the development of computer programs that can access data and use it to learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers to learn automatically without human intervention or assistance and adjust actions accordingly. In terms of classifiers, machine learning approaches can be categorized into supervised, unsupervised, and semi-supervised approaches.

Supervised machine learning: algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system can provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

Unsupervised machine learning: algorithms used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system does not figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

Semi-supervised machine learning: is a combination of supervised and unsupervised machine learning methods. This machine learning used when labeled data is not enough to produce an accurate model and when there is a lack of the ability or resources to get more labeled data, then use semi-supervised techniques to increase the size of training data with unlabeled data.

Machine learning used in hate speech detection; the most common approach used in hate speech detection is a supervised method. This approach is domain-dependent since it relies on manual labeling of a large volume of text for instant most of the research work reviewed for this research uses a classifier algorithm such as support vector machine (SVM), Random Forest (RF), Naïve Bayes (NB), Logistic regression (LR) and Ensemble method. Almost all of the research studies for automatic hate speech detection uses more than two classification algorithms for computational, comparison reason, and used to suggest which algorithm has high performance and accuracy for their proposed detection model.[11], [31], [38], [43].

Table 2. 1 Frequently Used Machine Learning Algorithms in Hate speech detection

Algorithm	Pros	Cons
SVM	High accuracy, deal with high dimensional, handle imbalance class, and non-linear problems.	A large dataset required higher training time, does not perform very well when the dataset has more noise or overlapping classes.
RF	Avoid the overfitting, wok with high dimensional data, provide a reliable feature importance estimate	time-consuming to modeling and prediction, require more computational resources.
NB	perform well small dataset and multi-class prediction,	Zero conditional probability Problem, and a very strong assumption of independence class features

LR	the algorithm can be regularized to avoid overfitting, and Outputs have an excellent probabilistic interpretation	weak to handle high dimensional data; Relies on transformations for non-linear features
Decision Tree (DT)	They are robust to scalable, outliers, and naturally, model non-linear decision boundaries.	Easy to overfitting; No ranking score as a direct result,
K-Nearest Neighbors (KNN)	High precision and accuracy, nonlinear classification, no assumption of features;	Sensitive to unbalanced sample set, heavy computing burden, poor interpretability

2.5.2.2 Deep Learning Used in Hate Speech Detection

Deep learning is part of a broader family of machine learning methods based on artificial neural networks. It is a machine learning technique that teaches computers to do what comes naturally to humans. Learning can be supervised, unsupervised or semi-supervised. Nowadays, Deep Learning models show promising in text mining tasks. It depends on the artificial neural networks but with extra depth. It tries to mimic the event in layers of neurons and attempt to learn in a real sense to identify patterns in the provided text. However, deep learning approaches are not always better than traditional machine learning supervised approaches[44]. The performance of deep learning is subject to the right choice of algorithm and number of hidden layers as well as the feature representation technique. For instance, research work [3] and [37], [44] propose a deep learning-neural networks based hate-speech text detection using Convolutional Neural Networks (CNN), Recurrent Neural Network(RNN), and Long short-term memory (LSTM) respectively. Also, their results showed a promising classifier's performance and accuracy for hate speech text detection.

2.6 Amharic language

Amharic is the official language of the Federal Democratic Republic of Ethiopia and spoken by 40% of the population as the first or second language. It is the second most spoken Semitic language in the world (after Arabic) and closely related to Tigrinya. It is probably the second largest language in Ethiopia (after Oromo, a Cushitic language) and possibly one of the five largest languages on the African continent. Despite the relatively large number of speakers, Amharic is

still a language for which very few computational linguistic resources have been developed for the language[12], [45], [46].

2.6.1 The Amharic Character Representation

Amharic utilizes Geez characters; the characters trace back to 4th century A.D. The first forms of the Geez script included only consonants, while the subsequent variants of the characters represent phoneme pairs of consonant-vowel. Like Geez, Amharic writing uses characters formed by a consonant-vowel combination. In Amharic, seven vowels are used, each in seven distinct forms that reflect the seven vowel sounds they are አ ፣ ኡ ፣ ኢ ፣ ኣ ፣ ኤ ፣ እ ፣ ኦ. There are 33 basic characters with seven forms representing a consonant and a vowel at the same time, which makes the Amharic script pronounced in the syllable. The first order is the basic form, and there are 33 basic forms with six derivations for each giving 231 characters[12]. Table 2.1 shows an example of the Amharic alphabet.

Table 2. 2 Amharic Characters Alphabet Example

	ä/e	***u***	***i***	***a***	***ē***	***ə/ e***	**o**
h	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
l	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
h	ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
m	መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
s	ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ

2.6.2 Amharic Punctuation

The Amharic language has around ten punctuation marks in but few of them used in a computer system. Also, most of them are sentence separator marks. Punctuation mark such as ፡ (hulet neteb)/ (word separator or space), ። (Arat Neteb)/ (full stop (period)), ፣ (Netela Serez)/(comma), and ፤ (Dereb Serez)/(semicolon).

2.6.3 Challenge in An Amharic Writing Scheme

Amharic writing scheme has some issues that are difficult to process Amharic text. One of these challenges is the redundancy of characters used in Amharic, more than one character to represent

the same sound. The various forms have their meaning in Ge'ez, but there is no clear rule that shows their purpose in Amharic[12]. The problem of the same sound with various characters is not only observed with core characters but also exhibited in the same order of characters. Those are, ሀ and ሃ; ሐ and ሓ; ኀ and ኃ; አ and ኣ. A word formed by using this character has the same meaning. For example, the word 'sun' could be written in a different way like ጸሀይ, ፀሀይ, ጸሃይ, ፀሃይ, ጸሐይ, ፀሐይ, ፀሓይ, ፀኀይ, or ጸኀይ. Amharic characters with different forms of the same sound are shown in Table 2.2

Table 2. 3 Amharic Characters with The Same Sound

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

2.7 Related Works

This section presents a comprehensive review of basic related works to the area of automatic hate speech detection on social media to clearly understand the general technique, method, and result of existing studies.

Amharic Hate speech, [1] Mossie and Wang perform preliminary study on hate speech detection for Amharic language, Creating a dataset of 1821 posts and comments from Facebook and 4299 instances keyword and phrase extracted for posts and comments, then binary classify the speech as "hate" and "not hate" using word2vec and TF-IDF for feature extraction, machine learning classifier algorithms NB model with achieved 73.02% and 79.83% and Random forest achieved 63.55 and 65.34% accuracy respectively for both features. The authors conclude that the result is promising to compute a large volume of data for a social network. The study considers hate speech as a binary problem.

Likewise, Ibrohim et al. [30] studied hate speech for the Indonesian language on social media. The authors collected tweets and create a binary class dataset comprising hate speech and non-hate

speech and classify using a different combination of feature and machine learning classifier algorithm, using BOW model, word n-gram, character n-gram, and negative sentiment feature extraction methods with Naïve Bayes, SVM, BLR, and RFDT models. Then achieved 93.5% f-measure the best performance with the combination word n-gram with RFDT than other combined models.

On the other hand, there is a problem of differentiating hate from offensive speech, Davidson et al. [11] studied the separation of hate for other instances of speech like an offensive speech for automatic hate speech detection. Using 33,458 English tweets and hate speech lexicon from hatebase.org for hate speech dataset, labeled into three categories hate, offensive, and neither, using bigram, unigram, and trigram features with TF-IDF, they also use part-of-speech, sentiment lexicon for social media. Logistic regression with L1 regularization uses as a classifier. The model has an overall precision 0.91, recall of 0.90, and an F1 score of 0.90. They conclude that high accuracy detection can be achieved by differentiating between these two classes of speech.

A deep learning-based hate speech detection, Gambäck et al. [3] present hate speech classification system for twitter using a dataset prepared by Benikova et al. [47] that have four class categories racism, sexism, both (racism and sexism) and not hate speech. Using four features embedding like word2vector, Random vector, character n-grams, and word vectors combined with character n-grams. These four features embedding with deep learning CNN, the models Tested by 10-fold cross-validation, the model based on word2vec embeddings performed best, with higher precision than recall, and a 78.3% F-score.

Traditional ML and deep learning, Del Vigna et al. [18] study an Italian online hate campaign on social network sites. They were using Facebook textual content of comments that appeared on an Italian public page as a source. The dataset labeled as no hate, weak hate, and strong hate, and by merging weak and strong hate as hate they form the second dataset. Leveraging morpho-syntactical features, sentiment polarity, and word embedding lexicons, the author's design and implement two classifiers algorithm for the Italian language, one traditional machine learning algorithm named SVM and the other is deep learning RNN named LSTM algorithm. Conducting two different experiments with both datasets that a least 70% of the annotator agreed on the class of the data. SVM and LSTM achieved an F-score of 80% and 79% for binary classification and 64% and 60%

for ternary classification, respectively. They were showing the same range of accuracy of both types of classification algorithms.

2.7.1 Summary of Related Works

The next Table 2.3 presents a summary of related work in hate speech detection research.

Table 2. 4 Summary of Hate Speech Detection Related Work.

Authors & year	Title	Feature extraction	ML Method & accuracy
Binary Class Hate speech			
Mossie and Wang (2017) [1]	Social network hate speech detection for the Amharic language	Word2Vec and TF-IDF	Random Forest and Naïve Bayes: 79.83% & 65.34% respectively
Alfin et al. (2017) [15]	Hate Speech Detection in the Indonesian Language: A Dataset and Preliminary Study	bag-of-word (BOW), word n-gram and character n-gram	NB, SVM, Bayesian LR (BLR)& (RFDT) :93.5% RFDT
Djuric et al. (2015) [34]	Hate Speech Detection with Comment Embeddings	paragraph2vec & BOW with TF & TF-IDF	logistic regression obtains 0.80 AUC
Watanabe et al. (2018) [2]	Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection	unigrams with sentimental, semantic features and pattern features	J48graft, SVM and RF: accuracy 87.4% for binary and 78.4% for ternary

Fauzi et al. (2018) [16]	Ensemble Method for Indonesian Twitter Hate Speech Detection	BOW with TF.IDF weighting	NB, KNN, Maximum Entropy, RF, and SVM, and two ensemble methods: hard and soft vote, F1 measure 79.8%. (SVM, NB, and RF)
Kiema Kiilu et al. (2018) [27]	Using naïve Bayes algorithms in the detection of hate tweets. (Kenya)	sentiment analysis & N-Gram feature	NB: 67.47% accuracy
Multi-class Hate speech			
Davidson et al. (2017) [11]	Automated Hate Speech Detection and the Problem of Offensive Language	bigram, unigram, and trigram features with TF-IDF	LR with L1 regularization: 90%
Tulkens et al. (2016) [14]	A Dictionary-based Approach to Racism Detection in Dutch Social Media	word2vec, Dictionary-based	SVM: 46% f1-Score
Gaydhani et al. (2018) [38]	Detecting Hate Speech and Offensive Language on Twitter using Machine Learning: An N-gram and TFIDF based Approach	N-gram and TF-IDF	LR, NB and SVM 95.6%
Shervin and Marcos (2017) [7]	Detecting Hate Speech in Social Media	word skip-gram, and surface n-gram	SVM: 78%
Binary and Multi-classification with deep learning			

Gambäck and Kumar (2017) [3]	Using Convolutional Neural Networks to Classify Hate-Speech	word2vec, Random vector, character n-grams, and word2vec+character n-grams	CNN with word2vec 78.3% F-score Multi-class
Biere and Bhulai (2018) [4]	Hate Speech Detection Using Natural Language Processing Techniques	word2vec with 300 dimensions	CNN accuracy of 91%, and a loss of 36%.
Badjatiya et al. (2017) [37]	Deep Learning for Hate Speech Detection in Tweets	Random Embeddings & Glove Embeddings	CNN, LTSM & Fast Text best accuracy is 93% f1- score CNN + Random & Glove Embeddings
Rule-based hate speech detection			
Gitari et al. (2015) [29]	A Lexicon-based Approach for Hate Speech Detection	semantic features, subjectivity features, and grammatical patterns features	subjectivity rule-based classifier called Subjclue lexicon F1-score 65.12%

CHAPTER THREE

3 Methodology

In this chapter, we discuss a research methodology to build a dataset and techniques in order to achieve research objectives and answer the research question. The following section in this chapter explains and justifies the methodology used in conducting the study on Amharic hate speech detection.

3.1 Building a Dataset

The objectives of this study are to detect Amharic hate speech. So, it needs to build a new Amharic hate speech dataset. This new dataset needed because there is no published or annotated dataset for this purpose. The process of building the dataset for Amharic hate speech consists of three main steps,

1. Gathering the Amharic post and comment textual data from public Facebook pages
2. Preparing, filtering, or consolidating gathered data into one file dataset. And
3. Annotating the dataset.

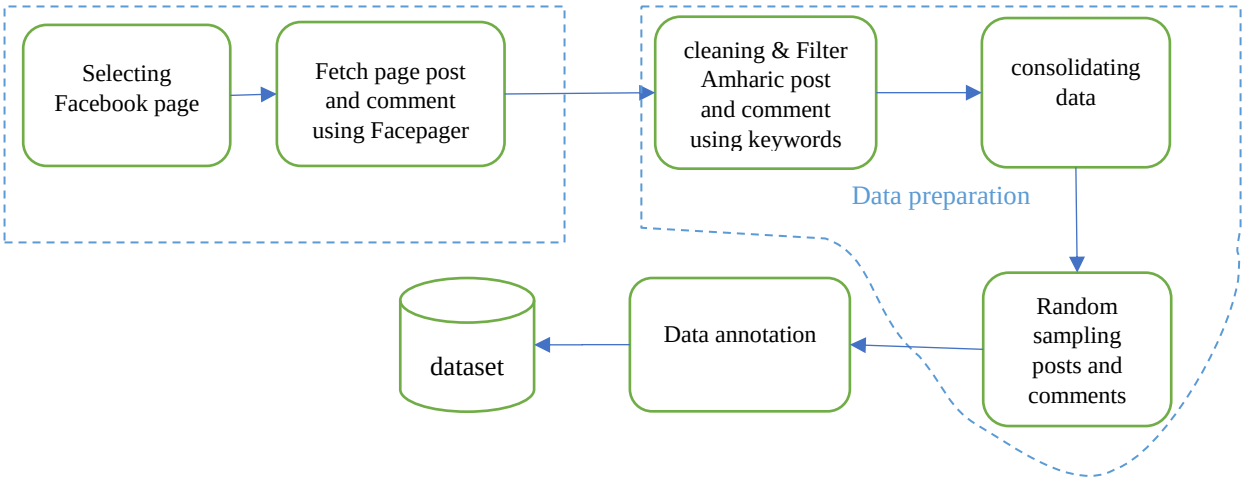

Figure 3. 1 Method for Building Amharic Hate Speech Dataset

3.1.1 Data Collection

The study gathers an Amharic textual data that is a post and comment from the Facebook platform. Amharic posts and comments were gathered from different categories of a popular public page because Facebook privacy policy does not allow access to the public content of a private page.

In order to collect the post and comments, a list of the Facebook public pages is gathered based on the content this page disseminates to the public view. Because the study aims are to have a diverse source of pages and have a representative dataset. Table 3.1 shows the list of Facebook page categories that this study used for dataset building. There are several sampling metrics or criteria that can be used to select these pages or a user on a social media platform. This study considered the following sampling criteria and metrics used for selecting a public page from the categories:

- Number of followers and likes greater than 50,000, which allow more active public pages included in categories.
- A page that posts news or hot issues on politics, ethnicity, religion, or gender daily.
- A page that uses the Amharic language most frequently for posts and comments.

Table 3. 1 Selected Social Media Page Categories

No	**Categories Name**	**Description**
1	News media and Broadcasting pages	delivering news to the general public or a target public
2	Bloggers and Journalists pages	a person who regularly writes material for a blog. Also, a person who writes for newspapers, magazines, or news websites or prepares news to broadcast.
3	Religious Media and Religious Group Pages	delivering religion news, religious teachings to the followers of that religion, or a target group.
4	Political Party, Politician and Government Office Official Pages	group of people often politicians, with common views, who come together to contest elections and hold power in the government. A group of politicians who have held the power of government.
5	Public figure Artist and Authors pages	A person or public figure who creates song, painting or drawing, and a writer of a book, article, or document.

6	Activist's, General or Interest Community Pages	A person who campaigns to bring about political or social change. Also, pages promote different issues on political, religious, ethnic, and other social issues on the page.

We decided to use Facebook from other social media platforms because of the popularity of the platform, and according to multiple statistics preform in Ethiopia social media. Facebook has 85% market share, and around over five million users in Ethiopia [48],[49].

The study collects posts and comments from a public page. The number of followers and likes is limited to larger than 50,000 in each category, because of a lack of resources needed to manage and to crawl all public Facebook pages in the categories. Also, collect all the posts and comments of the selected page that has been posted from April 2018 until April 2019. The one-year mark that the country experiences a political and socio-economic change in a different aspect, and also, the usage of social media, particularly Facebook in the country, has been increased significantly. The selected public Facebook pages information from each category listed in Table 3.2.

Besides the post and comment collection, this study collects keywords that are used for filtering the collected Facebook Amharic text data and used in the annotation process of the post and comment. These keywords are words that are deemed as offensives word or an indicter of offensive or hate speech text and words used to identify a target group. This study focuses on the following target groups and which have been presented in Appendix A annotation guideline.

- Political
- Ethnicity
- Religious and
- Gender

The keywords list contains a word that is offensive, violent, aggressive, disrespectful, and also a word used to identify a certain group of people (target group).

3.1.2 Dataset Preparation

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

- Removing All non-Amharic and non-textual posts and comments.
- Removing all Null, blank value, and whitespace.
- Filtering using keywords that are an indicator of hate and offensive language.
- Joining data of each page into one dataset and
- Removing duplication to ensure the uniqueness of each text in a dataset.

All the above preparation of the dataset considers the nature and behaviors of the Amharic language to keep the context of each text in a dataset for the annotation processes. The number of filtered posts and comments presented in Table 3.2.

The keywords are gathered for filtering purpose from university students, and also from different social media user pages known for using highly offensives words. The keywords contain offensive and words related to target groups. The sample keyword presented in Appendix B.

Table 3. 2 Selected Pages Information and Number of Post and Comment Filtered

Categories	No	Page Name	No of likes	No of Followers	Page created date	No of post and comment	No of filtered post and comment
News Media and Broadcasting Pages	1	Dire Tube	2,952,093	2,942,218	3-Feb-09	25416	1658
	2	Fana Broadcasting	1,385,636	1,429,074	18-Jul-11	96757	1016
	3	ESAT	1,255,042	1,281,716	17-Apr-10	71962	1805
	4	EBC	1,159,212	1,222,166	22-Feb-13	85329	2778
	5	VOA Amharic	891,164	907,374	21-Apr-10	21608	1317
Bloggers and Journalists' Pages	6	Yegna Tube - የኛ ቲዩብ	2,326,003	2,316,950	26-Jan-14	10618	616
	7	Yehabesha	1,898,675	1,891,431	23-Jan-11	15564	1017
	8	Haro Tube - ሐሮ ቱብ	1,572,501	1,570,095	11-Mar-13	10827	470
	9	Endewerede	988,982	988,895	15-Jun-16	3396	373
	10	Ethio Daily ኢትዮ ዴይሊ	685,710	685,616	6-Oct-15	4183	246
	11	Getu Temesgen	687,757	695,951	2-Nov-11	124024	2691
General or Interest Community and Activist's Pages	12	ከደራሲያን አለም ፔጅ(art)	504,580	513,666	14-Jan-18	45564	360
	13	አማራ ፋኖ-ዎሽንግተን ዲሲ	504,283	505,642	13-Feb-17	13772	1197
	14	ልሣነ ዐማራ- Amhara Press	310,867	321,544	22-Mar-16	17957	910
	15	ቤተ አማራ - Bête Amhâra	261,250	263,094	14-Mar-15	15629	1482
	16	Yene Siquar የኔ ስኳር ፔጅ	158,667	228,652	30-Jun-16	41354	1284
Religious Media and Religious Group Pages	17	ኦርቶዶክስ ተዋህዶ ለዘለአለም ትኑር	403,808	413,138	30-Apr-15	5994	165
	18	ማኅበረ ቅዱሳን ዋናው ማዕከል	349,299	349,414	4-Jan-14	819	31
	19	ቢቢኤን የናንተው ድምፅ	257,262	259,549	27-Dec-12	9628	91
	20	Ahmedin Jebel official	236,162	247,694	21-Feb-16	13914	192
	21	Memeher Dr Zebene Lemma	227,013	237,257	31-Oct-15	9477	58
	22	Ustaz Abubeker Ahmed	142,800	147,469	4-Jan-11	8653	102

	23	marsil TV	120,891	126,606	1-Jun-15	4187	98
Political Party and Politician Pages	24	Patriotic Ginbot 7	345,315	346,453	7-Jun-15	4024	660
	25	EPRDF Official	225,754	229,475	15-Nov-13	19157	1216
	26	የአማራ ብሔራዊ ንቅናቄ NMA	115,619	118,276	7-Jun-18	30451	1099
	27	Amhara Democratic Party /ADP/ CC	94,435	96,530	15-Nov-14	16316	978
Government Office Official Pages	28	PMO	434,772	450,964	20-Jan-17	33984	1334
	29	FDRE Attorney General	76,183	77,451	10-Feb-14	4663	125
	30	FDRE Communication Affairs Office Ethiopia	284,876	295,528	22-Jul-14	2883	245
Public figure Artist and Authors pages	31	Teddy Afro	1,191,985	1,182,398	29-Jun-12	20945	625
	32	Tamagne Beyen	297,860	314,445	24-Mar-12	17361	686
	33	Yared Shumete	171,326	174,620	27-Apr-15	18932	322
	34	lijyaredartist	137,023	141,341	6-Sep-12	3423	100
	35	aster bedane	63,783	65,922	1-Mar-18	7706	75
Total number of posts and comments						837077	
Total number of unique posts and comments						505609	
Total number of post and comment filtered							27422
Total number of unique post and comments filtered							27162

3.1.3 Dataset Annotation

Annotation is a procedure for adding information to the collected data or document at some level. In this case, the annotation process needs to label a post or comment for building hate speech datasets. The study uses a simple random sampling technique to select posts and comments to be annotated. The technique allows all the filtered posts and comments on each page to get an equal

chance to be annotated. The annotation conducted based on the instruction guideline provided by the researcher. The labeling conducted by at least four annotators.

3.1.3.1 Annotation Instructions

To make the annotation process clear, the instruction or a guideline was prepared and presented in Appendix A. The instruction is based on the definition of both hate and offensive speech and prepared by reviewing hate speech laws[20], [21], and previous research annotation guidelines [17] and also by consulting law experts. Using this guideline, the annotators label each post and comment to three different class 'Hate (HS),' 'Offensive (OFS),' and 'Neither (Ok).'

3.1.3.2 Annotation Procedure

The dataset annotation process, mainly managed by the researcher, also performs annotation with three additionally annotators. The number of annotators limited because of a lack of resources, mainly budget and time need for more annotators to participate in the process. The annotators were selected based on their willingness to perform the task and Amharic language skills. All the annotators are postgraduate students. The annotators were instructed to annotate a random subset of an equal number of posts and comments from the dataset. The three annotators instructed to annotate the same number of equal instances of posts and comments to measure the inter-rater agreement, which shows agreement level their annotation decisions match. Due to the challenging 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 comment into three different classes, as defined in Appendix A.

Finally, the annotated dataset with the three classes converted to the binary class dataset the classes are Hate and Non-Hate. Only by converting all offensives class to hate class. The importance of converting to binary class datasets for comparison of detection model using the two datasets and to the previous work[1].

3.2 Hate Speech Detection Modeling

3.2.1 Preprocessing

Amharic text preprocessing method used to clean up the post and comments based on the Amharic language and uses basic text mining preprocess techniques. This method used for making the dataset ready for the feature extraction method. The following methods used in the prepossessing processes:

- Removing (cleaning) Irrelevant character, punctuations, symbol, blank value and whitespace, URL, and emojis.
- Removing elongation from the post and comment.
- Performing text normalization.
- Performing Tokenization to get the individual word or tokens in text.

3.2.2 Feature Extraction Methods

Feature extraction methods based on state-of-the-art text mining; techniques applied for reducing redundant features and dimensionality. The methods involved in selecting a subset of relevant features that would help in identifying hate, offensive, and neither from the dataset and can be used in the modeling of the detection problems. The N-gram, TF-IDF, and word2vec feature extraction used in this study because they are better and popular feature extraction methods used hate speech detection studies and text classification problems. The following features extraction methods are used for modeling Amharic hate speech detection.

N-Gram: N-grams are one of the most used techniques in hate speech automatic detection. 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 in combining sequential words into lists with size N, where N is the number of words used in the probability sequences. E.g., if N=1, N=2, and N=3 referred them as unigram, bigram, and trigram, respectively. This study uses a word N-gram method to create N-gram of post and comment features. These three n-grams used because when the number of N increases, the model performance remains the constants.

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[39]. More detail discussed in chapter two.

The feature vectors extracted by each of the above methods used for training the detection models and then by combining each feature extraction method, for instance, N-gram weighted by TF-IDF [11].

Word2vec: it takes a large corpus of text as input and produces a vector space that will be used when training a model for Amharic 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 [14]. Simply the model gives relationships between the words in the text. Using Word2vec help to get a feature model not only from our annotated dataset but also from the whole gathered data collected for this study.

3.2.3 Machine Learning Classification Algorithm

The objective of this study is to automatically detect Amharic hate speech using a machine learning algorithm on categorized and labeled datasets. We use a multiple supervised machine learning algorithm for comparison and accuracy of the detection. The algorithms are selected because of their good classification performance, and there the popularity in solving hate speech detection problems on previous research work results in related work for other languages and English languages[19], [33], [2], [11], [38]. The following three machine learning algorithms used to build detection models.

Support Vector Machine (SVM): is a supervised machine learning algorithm that can be used in classification problems. SVM algorithm performs classification by finding hyperplane in n-dimensional space that separates the classes or data points in feature space. Hyperplanes are decision boundaries that help classify the data points. A hyperplane in n-dimensions is an affine subspace of dimension n-1[50]. In general, the equation below shows the form of a hyperplane for a set of input point X.

$$w * x + b = 0 \text{ ----------} (3.1)$$

Where: w are the support vectors to the hyperplane, the intercept (b) is found by the learning algorithm, and x is a set input data point. The SVM classifier for predicting a new input can be written as,

$$f_{w,b}(x) = g\,(wx + b) \text{ ----------} (3.2)$$

Here, then f(x) >0 for points on one side of the hyperplane, and f(x)<0 for points on the other.

This study used a one-vs-the-rest strategy of SVM classifier because SVM is inherently binary classifiers, but the goal here is to classify two datasets with three and two labeled classes. The approach of the one-vs-the-rest SVM classifier is to train a binary classifier for each class in the dataset.

Random Forest (RF): is a supervised classification algorithm as the name suggests, this algorithm creates the forest with several decision trees that operate by constructing a multitude of trees at training time. The building blocks of the RF model is Decision Trees (DT). RF uses bagging and feature randomness when building each decision tree and creates an uncorrelated forest of trees whose prediction by the group is more accurate than that of any individual tree [51].

Naïve Bayes (NB): is a probabilistic classifier technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, the NB classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. The equation for Bayes' Theorem[50].

$$P(c|x) = P(x|c) / P(c)P(x) \text{ ----------} (3.3)$$

Where, P(c|x) is the posterior probability of class c given to data x, P(x|c) is likelihood which is the probability of x data predicted of class c, P(c) is the prior probability of class c, and P(x) is the prior probability of the data x.

3.3 Evaluation

Evaluation of this study has been conducted on multiple phases of the development of the detection model for hate speech. After all, the evaluation conducted the results discussed and concluded by suggesting a model to detect Amharic hate speech.

3.3.1 Inter Annotator Agreement

Inter annotator agreement evaluation computes the agreement of human annotators of the dataset by calculating how similar the annotators labeled post and comments in the dataset. The study used Cohen's kappa for calculating the agreement level. We instructed them to annotate the same amount and a similar subset of post and comment in the dataset. To calculate the kappa using the dataset that receives annotation form all annotators. The Kappa value varies from 0 to 1. The standard interpretation of Kappa values [52] listed as follows.

- Poor agreement when kappa Less than 0.20
- Fair agreement when kappa between 0.20 to 0.40
- Moderate agreement when kappa between 0.40 to 0.60
- Good agreement when kappa between 0.60 to 0.80
- Very good agreement when kappa between 0.80 to 1.00 and 1.00 is perfect agreement

3.3.2 Detection Model Evaluation

The hate speech detection must be evaluated using validation techniques to see whether the model fits with the dataset and to validate the model correctly work for the unseen new post and comment. So, we used cross-validation methods to make sure that models got the pattern for the training dataset. Cross-validation is a resampling procedure used to evaluate machine learning models on a limited dataset sample. In order to validate the model, we used the K-fold cross-validation method. K-fold CV selected because of its common popularity and also its simplicity for implement the selected machine learning classifier, and it avoids overfitting and underfitting problem[53]. Model testing techniques K-fold CV used to evaluate machine learning models on a limited dataset. The entire dataset with the feature splits into K part and K-1 used for training, and one used test set and also helps to avoid overfitting and underfitting. The study uses the 5-fold CV to validating and test the models.

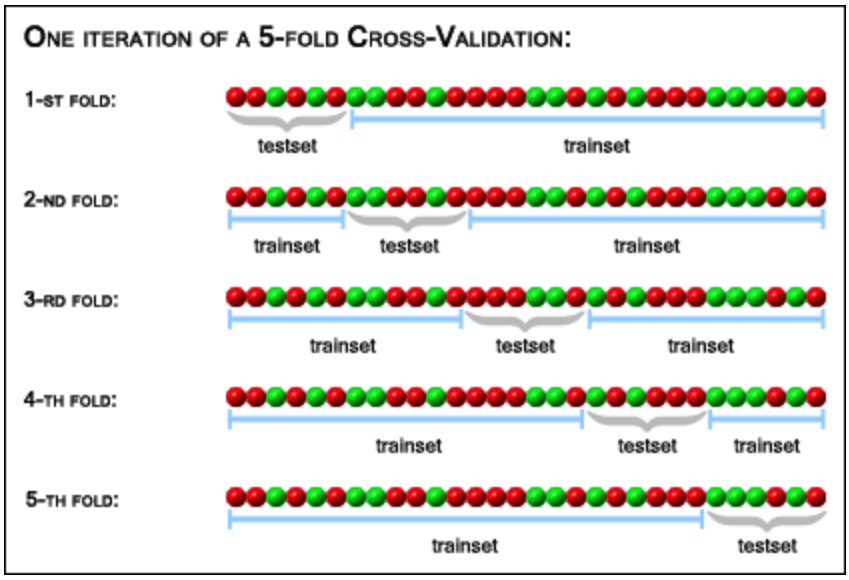

Figure 3. 2 A 5-fold Cross-Validation Evaluation[54]

3.3.2.1 Detection Model Performance Evaluation Metrics

Detection Model evaluation is required to quantify the detection model performance because of classifier algorithms gives a biased result. Besides the CV testing, this study used different metrics appropriate for model performance evaluation. Metrics used, namely are accuracy, precision, recall, F-Measure, and confusion matrix. Terms associated with performance evaluation:

- True Positives (TP): are the cases when the actual class of the data point was True and the predicted is also True.
- True Negatives (TN): are the cases when the actual class of the data point was False and the predicted is also False.
- False Positives (FP): are the cases when the actual class of the data point was False and the predicted is True.
- False Negatives (FN): are the cases when the actual class of the data point was True and the predicted is False.

3.3.2.1.1 Accuracy

Accuracy shows the classification problem correct prediction value and calculates as the total number of the model correct prediction divide by all number of data instances used for the model [53]. Accuracy calculated using the equation:

$$\text{Accuracy} = \frac{TN+TP}{for\ all\ number\ of\ instance\ data} \text{ ---------- (3.4)}$$

3.3.2.1.2 Precision

Precision measures the predicted value true, and it shows how many times the model predicts true. Precision answer what proportion of identifications was correct. To Precision calculated using the equation:

$$\text{Precision} = \frac{TP}{TP + FP} \text{ ---------- (3.5)}$$

3.3.2.1.3 Recall

Recall answers what proportion of actual positives was correctly identified. Recall calculated using the equation:

$$\text{Recall} = \frac{TP}{TP + FN} \text{ ---------- (3.6)}$$

3.3.2.1.4 F-Measure

The F measure (F1-score or F-score) is a measure of a test's accuracy and defined as the weighted harmonic mean of the precision and recall of the test. F1 is more useful than accuracy when the dataset contains uneven class distribution [53]. This score is calculated using this equation:

$$\text{F1-score} = 2* \frac{(Recall*Precision)}{(Recall+Precision)} \text{ ---------- (3.7)}$$

3.3.2.1.5 Confusion Matrix

A confusion matrix is a technique for summarizing the performance of a classification algorithm. The number of correct and incorrect predictions are summarized with count values and broken down by each class. It gives insight not only into the errors being made by the classifier but, more importantly, the types of errors that made. So, this study uses a normalized confusion matrix for both binary class models (Hate and OK only) and three class models (Hate, Offensives, Neither). The following Table 3.2 shows a sample format of a confusion matrix with 3-classes [55].

Table 3. 3 Sample Format of a Confusion Matrix for Ternary-Classes

		Predicted values		
		Hate	Offensive	Neither (OK)
Actual values	Hate	True hate	False Offensive	False OK
	Offensive	False Hate	True Offensive	False OK
	OK	False Hate	False Offensive	True OK

Table 3. 4 Sample Format of a Confusion Matrix for Binary Classes

		Predicted value	
		Hate	Non-Hate (NH)
Actual values	Hate	True Hate	False NH
	Non-Hate	False Hate	True NH

CHAPTER FOUR

4 The proposed solution for Automatic Amharic Hate Speech Detection

This chapter discusses the proposed solution of Amharic language hate speech detection using machine learning techniques suitable to solve the problem of understudy. This study creates two new hate speech datasets for posts and comments from Facebook public pages and used it as the main component in the proposed solution to detect hate speech on social media.

4.1 The proposed Hate Speech Detection Architecture

The proposed architecture used to detect Amharic posts and comments into hate, offensive, and neither speech. The proposed solution based on the architecture shown in figure 4.1. It takes Amharic datasets as input and then the preprocessed based on the language nature, which removing punctuations, normalization, tokenize, and another basic necessary preprocess. 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, models developed using SVM, NB, and RF machine learning algorithms and training set with feature vectors data frame of the whole dataset. The models evaluated using K-fold cross-validation. The evaluation result used to select the best detection model. The outcome of these tasks is a model used for hate and offensives speech detection. Finally, the detection model is evaluated and selected based on the results obtained using the model evaluation method discussion in chapter three. The final selected detection model used to develop a prototype that can take new Amharic texts as input and it classifies the input whether it contains hate, offensives or normal speech.

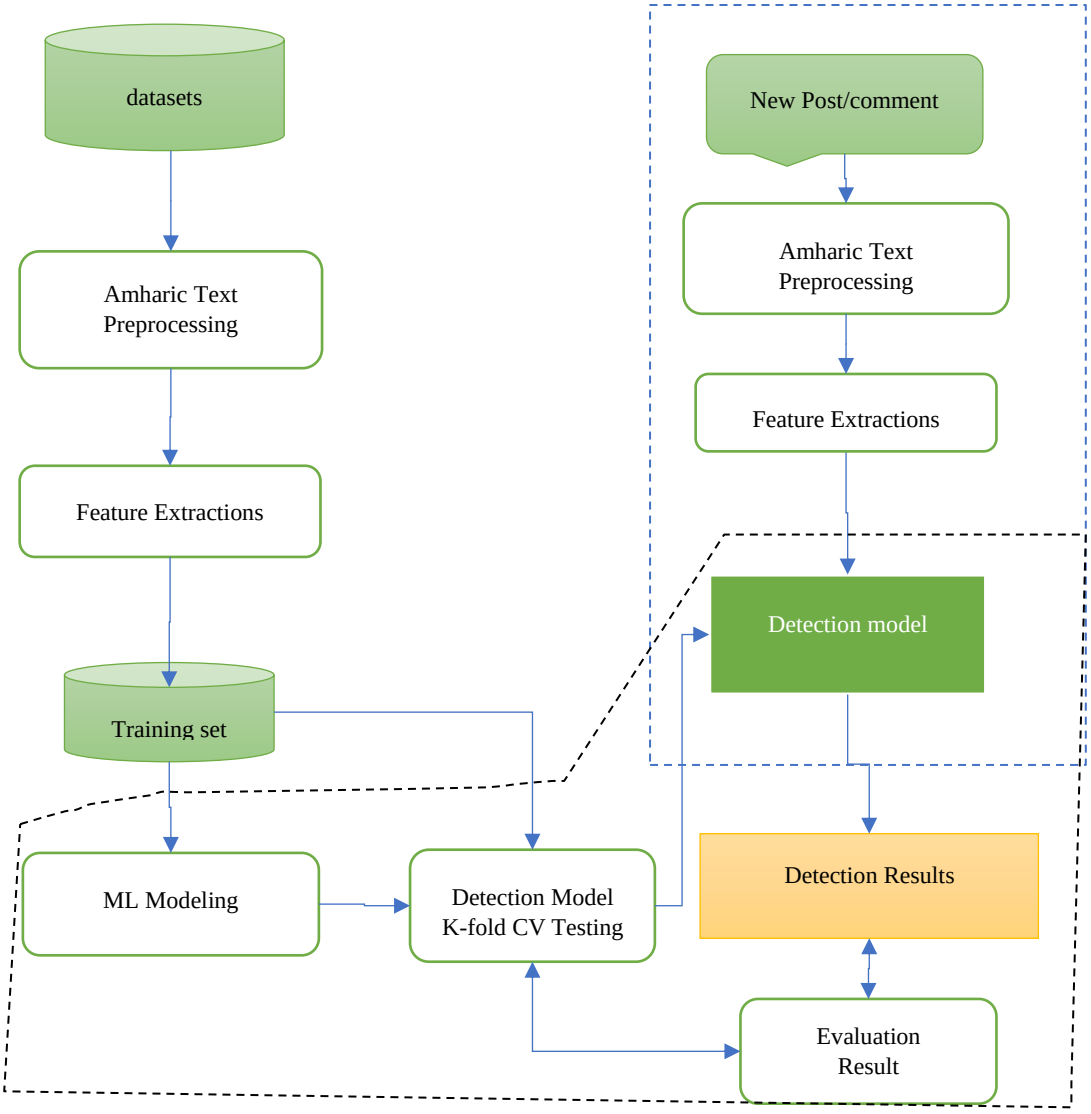

Figure 4. 1 Amharic Hate Speech Detection Architecture

4.2 Proposed Amharic Text Preprocessing

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

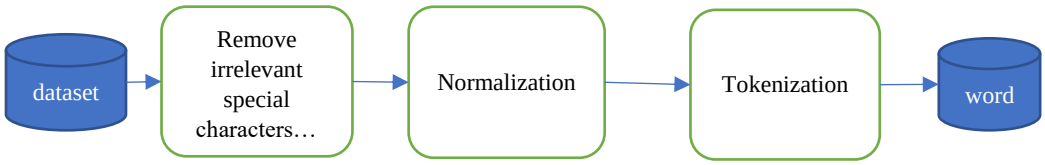

Figure 4. 2 Amharic Dataset Pre-Processing Steps

4.2.1 Removing (Cleaning) Irrelevant Character, Punctuations Symbol, and Emojis

The dataset is created using social media text, which is posts and comments usually contain a special character, punctuations, symbol, and emojis to express different opinions and feelings. This cleaning task removes all irrelevant special characters, symbols, and emojis. Also, remove all non-Amharic characters.

Pseudocode: 4.1 Removing Irrelevant Characters

Input: text in a dataset
Output: clean text
Begin:

1. Read the text in the dataset;
2. While (! end of the text in a dataset):
   - If the text contains special_char [,'! @#$%^&*] then
     - Remove special_char
   - If the text contains symbol [<>‹›« »=፦-`~_/] then
     - Replace symbol and add space
   - If a text contains geez_number [፩ ፪ ፫ ፬ ፭ ፮ ፯ ፰ ፱ ፲ ፳ ፴ ፵ ፶ ፷ ፸ ፹ ፺ ፻ ፼ ፽ ፾] then
     - Remove geez_number
   - If a text contains Amharic_Punc='[፤፡።፣, • ፧] then
     - Remove Amharic_Punc
   - If text contain English_word_num = [a-z A-Z] [0-9] then
     - Remove English_word_num
   - If text contain number = [0-9] then
     - Remove number
   - If a text contains emoji = [😂👍 😭 .....] then
     - Remove emoji
   - If a text contains extra white space then
     - Trim the text
3. Return clean_text;

End:

Algorithm 4. 1 Removing Irrelevant Characters (Preprocessing)

4.2.2 Normalization Amharic Character

Normalization used to solve the redundancy problem of Amharic language, the same sound character with different character form as shown in Table 2.2 of chapter two. It is a process of changing words into a single form by performing character replacement with a similar sound to one common form of character. The change of the character into one common representation does not cause a meaning difference, but it decreases the chance of getting the same feature with different characters, which leads to duplication of a feature.

Pseudocode 4.2: Normalization

Input: Amharic post and comment in dataset
Output: Normalization Amharic post and comment.
Begin:

1. Read text for dataset;
2. While (! end of text in dataset):
    If text contain characters [ሐ] [ሑ] [ሒ] [ሓ] [ሔ] [ሕ] [ሖ] [ሗ]; then
        Replace characters with [ሀ] [ሁ] [ሂ] [ሀ] [ሄ] [ህ] [ሆ] [ሀ] # *respectively*
    If text contain characters [ኀ] [ኁ] [ኂ] [ኃ] [ኄ] [ኅ] [ኆ], then
        Replace the characters with [ሀ] [ሁ] [ሂ] [ሀ] [ሄ] [ህ] [ሆ]
    If text contain characters [ሠ] [ሡ] [ሢ] [ሣ] [ሤ] [ሥ] [ሦ], then
        Replace the characters with [ሰ] [ሱ] [ሲ] [ሳ] [ሴ] [ስ] [ሶ]
    If text contain characters [ዐ] [ዑ] [ዒ] [ዓ] [ዔ] [ዕ] [ዖ] [ኣ], then
        Replace the characters with [አ] [ኡ] [ኢ] [አ] [ኤ] [እ] [አ][አ]
    If text contain character [ጸ] [ጹ] [ጺ] [ጻ] [ጼ] [ጽ] [ጸ], then
        Replace the character with [ፀ] [ፁ] [ፂ] [ፃ] [ፄ] [ፅ] [ፆ]
3. Return replaced text;

End:

Algorithm 4. 2 Normalization Amharic Character with the Same Sound

4.2.3 Tokenization

After the cleaning and normalizing tasks, the Tokenization method follows, which splits the post and comment text into individual words or tokens by using spaces between words or punctuation marks this is important because the meaning of text generally depends on the relations of words in that text and help the feature extraction methods to get appropriate feature form the dataset.

4.3 Proposed Feature Extractions

The proposed feature extractions component of the detection system performs the extraction of important features of a dataset. These methods take input of prepossessed and tokenized dataset words and perform extractions, as shown in Figure 4.3. Then extracted features used for training the models and also to predict the class of posts and comments as hate, offensive and normal.

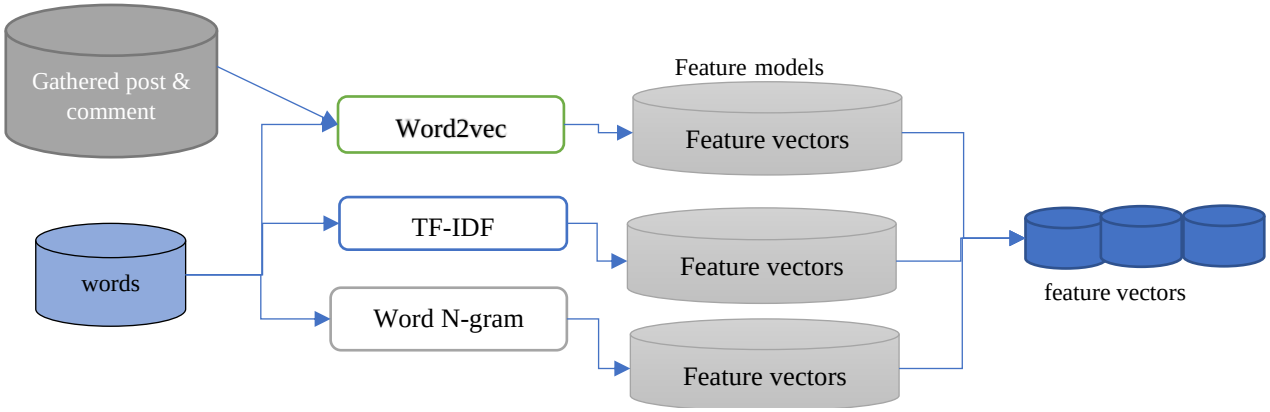

Figure 4. 3 Feature Extraction Flow Diagram

These components use Word2vec, TF-IDF, N-Gram feature extraction methods are well known in the text mining approach, as described in chapters two and three. Each method gives feature vectors used to train machine learning classifiers.

4.3.1 N-Gram Feature Extraction

The study proposed a word N-gram feature extraction method experimented on a different value of N that ranges from for one to three. Where N is the number of words used in the probability sequences. An n-gram of two words is called bigram (2-gram). The feature extraction performed by using unigram, bigram, trigram and combination n-grams. The performance of n-gram needs a proper choice of the n value. Also, it gives a different feature model to train and to comparing n-gram features to one another.

4.3.2 TF-IDF Feature Extraction

The study proposed a TF-IDF feature extraction is experimented to get the word frequency in the dataset by applying the TF and the importance of the word in the dataset represented by measuring

IDF of the word in the dataset. This featured model gives the classifiers the frequency and the importance of the word in the dataset as a feature vector for training.

4.3.3 Word2vec Feature Extraction

In this study, the proposed word2vec method performs word to vector modeling on a larger amount of data. These data are posts and comments gathered from all of the selected Facebook pages that we include in the data collection stage. The posts and comments used for building the model. This method is used due to the absence of standard Amharic 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 word in post and comment. These models used to extract features from the text in the dataset by calculating the average of all vectors using the model.

The proposed featured extraction in the study has not only to perform feature extraction based on single methods. However, also, we proposed a combination of some of these methods together like n-grams weighted by TF-IDF and combined n-grams. Multiple features used in order to demonstrate a comparison of each features' extraction methods performances. Because one of these study points is to select better feature extraction methods for hate speech detection.

4.4 Machine Learning Models Building

In these subcomponents of the proposed architecture for Amharic hate, speech detection performs machine learning classifier training on all feature vector constructed by the feature extraction components methods. This study builds a classifications model mainly by using a machine learning

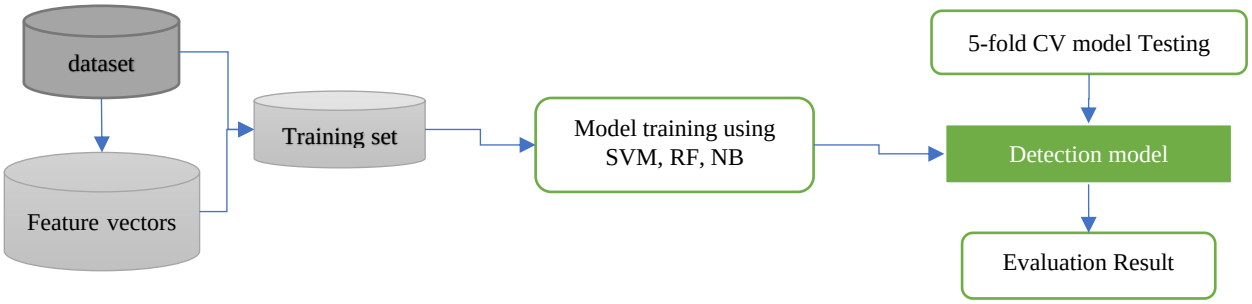

Figure 4. 4 Model Building Flow Diagram

classifier algorithm SVM, RF, and NB on the dataset features and labels. Which could be selected for the detection of the Amharic hate and offensive speech. The process of modeling is to find patterns in the training set of the dataset that maps the post and comments with their features to the target class by using learning algorithms. The output of these modeling is a trained model that can be used for detecting Amharic hate speech, making predictions on new input posts and comments.

This study proposed the One-vs-Rest (OVR) strategy of Support Vector Machine (SVM) machine learning classifier because a simple form of SVM algorithm is a binary classifier. It used to separate one group from another of the classes. In order to classify multiple groups of class, we applied a modified version of the SVM algorithm which is used for multiple class classifications. It is the OVR method. This method separates each class from the rest of the classes in the dataset.

The proposed classifier, Naïve Bays (NB) classifier is a probabilistic machine learning model that's used for classification. This study uses multinomial NB for modeling. NB classifier is a specific instance of the classifier which uses multiple distributions for each of the features in the dataset.

The proposed Random Forest (RF) classifier is a meta estimator that fits several decision tree classifiers on a subsample of the dataset. It means RF consists of a large number of individual decision trees that operate as an ensemble by bagging and feature randomness.

4.5 Models Evaluation and Testing

In order to test and estimate the learning ability of machine learning models trained on the hate speech dataset. The basic concern of the machine learning model is to find an accurate estimation of the generalization error of the trained models on finite datasets. Because of the reason this study used proposed K-fold cross-validation (CV) and different performance evaluation metrics appropriate for models, these metrics are confusion matrix, accuracy, precision, recall, and F-Measure, as discussed in chapter three.

CHAPTER FIVE

5 Implementation and Experimentation

This chapter presents the implementation of the proposed solution for Amharic Hate speech detection by using the methodology discussed in chapter three. Also, implement the proposed solution in chapter four. This study follows an experimental approach to determine the best combination of the machine learning algorithm and features extraction for final models' comparison. Moreover, performs two different experiments by using two datasets. The first experiment is using a binary-class dataset and the second is performed using the ternary class dataset. Finally, develop a prototype for hate speech detection by using selected best model.

5.1 Implementation Environment

This study used several development tools and packages to implement the proposed solution, Amharic hate speech detection. This study uses python programing language for implementing and experimenting with each proposed solution from the data preprocessing to the model building phase. Also, to evaluate the implemented proposed classifiers model. Python used because of its programming language of choice for developers, researcher and data scientists who need to work in machine learning models. 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 the Tools and Python Package Used During the Implementation

Tools		
Tools	Version	Description
Anaconda Navigator	1.9.6	allows us to launch development applications and easily manage conda packages, environments, and channels without the need to use command-line commands.
Jupyter notebooks	5.7.8	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	easy to learn, powerful programming language to develop a machine learning application.
Facepager	3.10	Social media content retrieval tools. Used for data collection tasks to build the dataset. This tool used to fetch posts and comment on Facebook a public page and store the data in SQLite database. Function to export the database to CSV file which is easy for manage datasets.
Microsoft Excel	2016	Used data preparation tasks in cleaning filtering and sorting the gather data. Also, used to manage the annotation task.
Python Packages		
Scikit-learn	0.20.1	A set of python modules for machine learning and data mining. This study uses it for feature extraction and training and testing model. The name of the package is called sklearn.
pandas	0.24.2	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.
NumPy	1.15.4	Array processing for number, strings, and objects. This study uses it for handling converting the text to numeric data for features and training and testing the model.
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 Amharic text.

gensim	3.4.0	Python library for topic modeling document indexing and similarity retrieval with large corpora. This study uses it for the constricting word2vec model.
nltk	3.4	Build python programs to work with human language data. This study uses it for tokenization.
matplotlib	3.0.3	Publication quality figures in python. This study uses it for data and results visualization.
flask	1.0.2	Microframework for making web services in python. This study uses it for implementing the prototype for selected models.

5.2 Deployment Environment

The tools which are discussed above in section 5.1 have been deployed on a personal computer equipped with a processor Intel® Core™ i5-4310M, CPU 2.70GHz, 2 Core(s), 8 Gigabyte of physical memory, 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 form Facebook using the content retrieval tools Facepager. First, 35 different Facebook public pages were selected. These pages belong to categories that contain a range of 3 to 6 selected pages based on the selection criteria of public pages. Then all posts on the page from April 2018 to April 2019 collected and also commented under each post. Then, filter the Amharic post and comments by removing all non-Amharic and non-textual data. This process results in a total number of 837077 Amharic posts and comments. Table 3.2 in chapter three shows the result of the total number of posts and comments that are filtered and the selected Facebook pages information in category. The total of 27162 unique pages posts and comments are filtered using the keyword. The keyword helps to filter the post and comments which likely to have a hateful or offensive speech in their content.

The final annotated three-class dataset contains a total of 5000 posts and comment that has been labeled with a class of Hate (HS), Offensive (OFS) and Neither offensive nor hate (OK). The detail

method for building the dataset are discussed in chapter three. Also, the result of the dataset annotation process presented in the next chapter. Dataset annotation resulted in a distribution of the ternary classes, and binary class datasets are shown in Table 6.3 and Table 6.4 respectively.

The dataset contains the Amharic text in a message, object_id, and created time of posts and comments along with one of the three labels. The post and comment with their labeled class only used for modeling the detection. The dataset contains the following four columns.

- **object_id** is a Facebook ID for the post and comment
- **message** contains the actual post and comments.
- **created_time** is the timestamp that the post or comment created.
- **label** is the class the annotator assigned to the post and comment. These labels for the ternary dataset are OFS for offensive, HS for hate, and OK for Neither and labels for the binary dataset are HS and NH post and comments.

5.4 Preprocessing Implementation

To implement the Amharic preprocessing method, the study used python programming language and modules. In order to perform the whole implementation of the methods, we perform the following common activities, which are importing important libraries packages and load the dataset file to the disk using the code in figure5.1. The loaded dataset using panda is used throughout the whole implementation of this study. The Amharic preprocessing method uses a Python RegEx (regular expression) re and nltk modules.

```
import pandas as pd
ds = pd.read_csv("dataset/hate_dataset.csv")
amh_post_comment =ds.message
labels = ds.label
```

Figure 5.1 Python Code for Load the Dataset Post and Comment with Labels

5.4.1 Implementation of Removing (Cleaning) Irrelevant Character

In order to Clean the post and comment on the dataset, we implement a method that performs removal and replace the punctuation mark, special character, symbol, emojis, etc. a method is written using the Python re module. The *preprocess ()* method accepts the post and comment text

and then removes or replaces with a single space the unwanted characters. The method returns are clean posts and comment text. The code for cleaning the post and comment of the dataset is shown in Appendix C.1.

5.4.2 Implementation of Normalization of Amharic Character in Text

To normalize the Amharic characters with the same sound into one selected common character form, we implement this method using the Python with the re module. The method *normalization_char()* accepts the post and comment text, replace the characters into one common character form. The method returns the is normalizing post and comments text. The code for normalizing the Amharic post and comment in the dataset shown in Appendix C.2

5.4.3 Implementation of Post and Comment Tokenization

To get token or word of the text in the dataset, we used a python module nltk. We used python splits function and nltk method for word tokenize. The output of this function is tokens or words, which is input for feature extraction methods or used by feature extraction methods.

```
def tokenize_text(post_comment):
    post_comment=preprocess(post_comment)
    post_comment=replace_same_sound_amh_char(post_comment)
    post_comment=remove_emoji(post_comment)
    tokens = []
    for sent in nltk.sent_tokenize(post_comment):
        for word in nltk.word_tokenize(sent):
            if len(word) < 2:     # discard a word with 1 character
                continue
            tokens.append(word)
    return tokens
```

Figure 5. 2 Sample Code for Tokenization

5.5 Feature Extraction Implementation

To extract features from the dataset that is used for training the model. We used the python Scikit-learn module for TF-IDF and N-gram and python Gensim module for word2vec. In order to implement the feature extractor method, first, the dataset must be clean and normalize using the preprocessing methods and each feature extraction use list of tokenized post and comments as the

input. Each method is implemented with vector transformation or vectorizer methods. 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. This class converts posts and comments in the dataset to a matrix of N-gram features for the defined N value. *CountVectorizer* class used to extract the word n-gram features vectors by set the parameter *ngarm_range* to the proposed N values and analyzer to a word. Also, the study experiments recursively different n-gram. The experimented n-gram are unigram, bigram, trigrams, and their combination. Figure 5.3 sample code returns the trigram feature vectors by using *fit_transform ()* method and it set on n_gramdata used for training the models.

```
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(amh_post_comment).toarray()
```

Figure 5. 3 Sample Code Used to Generate n-gram Features.

5.5.2 Implementation of TF-IDF

To implement the TF-IDF features extraction method, this study used a *TfidfVectorizer* class of sci-kit learns package. This class converts a post and comment of the dataset to a matrix of TF-IDF features vectors. Which contain the word frequency and their importance in the dataset. For feature modeling with TF-IDF, we instantiate the TfidfVectorizer class and pass the post and comment text in the dataset to *fit_transform ()* method of the class. The study experiment with different hyperparameter and the best parameter is applied for the final modeling.

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer( tokenizer=tokenize_text, preprocessor=preprocess,
    use_idf=True,
    smooth_idf=False,
    norm=None,
    decode_error='replace',
    max_features=50000, #change to best value iteratively
    min_df=3,
    max_df=0.75
    )
tfidf = vectorizer.fit_transform(amh_post_comment).toarray()
```

Figure 5. 4 Sample Code for Extracting TF-IDF

5.5.3 Implementation of Word2Vec

To implement **word2Vec,** this study uses a python Gensim module which is used to implement different embedding methods. Feature modeling with gensim word2Vec is straightforward. First import and instantiate word2Vec class with necessary parameter and builds the vocabulary, and training the Word2Vec model using the posts and comments that are gathered from the select pages.

```
from gensim.models import Word2Vec
w2v_model = Word2Vec(min_count=2,
                     window=10,
                     size=150)
w2v_model.build_vocab(amh_post_comment, progress_per=10000)
w2vmodel=w2v_model.train(amh_post_comment, total_examples=w2v_model.corpus_count, epochs=30, report_delay=1)
w2v_model.save('w2vmode21.bin')
```

Figure 5. 5 Sample Code for Building Word2vec Feature Model

The training resulted in a feature vector is also known as the embeddings. Embeddings are features that describe the target word. Then the result word2vce model used to extract features by computing similarity for a word in the dataset and used it as a feature to train machine learning models.

5.6 Machine Learning Models Implementations

To build machine learning models using the proposed algorithm, we used a python sci-kit learning library package and followed the conventional method that is importing the important library package for modeling and metrics for model evaluation.

```
# importing package used for modeling and evaluating
import pandas as pd
import numpy as np
import logging
import pickle
import nltk
import gensim
import sys
import re
from gensim.models import Word2Vec
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import LinearSVC, SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import MultinomialNB,GaussianNB
from sklearn.multiclass import OneVsRestClassifier
from sklearn.metrics import classification_report, accuracy_score
from sklearn.model_selection import cross_val_predict, cross_val_score
from sklearn.metrics import confusion_matrix
from sklearn.pipeline import Pipeline
import matplotlib.pyplot as plt
```

Figure 5. 6 Importing the Important Package for Modeling

After the feature's extraction process, the post and comments with the extracted features vectors used as X_train training and the labels which contain the class or labels of post and comment belong set as Y_train. Also, then the classifiers train by using the X_train and Y_train of the entire dataset. To train using the *fit()* method with the appropriate set parameter for each classifier instantiate the class.

```
X_train = pd.DataFrame(feature_extracted) # extracted feature with post and comment
y_train = ds['label'] # the lables which is class each posts & comments belonges
```

Figure 5. 7 Code for Preparing Dataset with Extracted Feature for Models Training

The study used the *LinearSVC()* classifier of the sklearn package to building the SVM model. This classifier is instantiating with a basic parameter like OVR to classify the multi-class dataset.

```
# instantiate SVM classifier
svm_model=LinearSVC(multi_class='ovr', penalty='l2', loss='squared_hinge', dual=True,
                    C=0.01, class_weight='balanced', verbose=0, max_iter=10000)
# to train SVM
svcmodel.fit(X_train, y_train)
```

Figure 5. 8 Instantiating SVM and Fit the Model

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

```
# instantiate naive_bayes classifier
NBmodel = MultinomialNB(alpha=1.0, fit_prior=True, class_prior=None)
# to train naive bayes
NBmodel.fit(X_train, y_train)
```

Figure 5. 9 Instantiating Naïve Bayes and Fit the 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 n_*estimators* parameters.

```
# instantiate RandomForest classifier
RFmodel = RandomForestClassifier(n_estimators=100,  max_features='auto', class_weight='balanced')
# to train RF
RFmodel.fit(X_train,y_train)
```

Figure 5. 10 Instantiating Random Forest and Fit the Model

5.7 Model Testing and Evaluation

The training that is performed is validated using the using K-fold CV technique implement using the sklearn *cross_val_score()* and *cross_val_predict()* methods using K value five. These methods use the models and dataset to validates the learning skill of each model. We use a 5-fold CV, which means that the dataset is divided into five different sets and four set use to train and one set used to test models each 1 to k iteration.

```
# methods for 5-fold cv
from sklearn.model_selection import cross_val_predict, cross_val_score
cvscore = cross_val_score(Models, X_train, y_train, cv=5)
cvscore, cvscore.mean() # to get the accuracy result of each fold and mean
# to get the prediction class by the models when 5-fold cv testing
y_pred = cross_val_predict(Models, X_train, y_train, cv=5)
```

Figure 5. 11 Performing 5-fold CV on Models

The study used a *classification_report (), accuracy_score()* and *confusion_matrix()* methods, which are used to evaluate the performance of the built models using predict the value of 5-fold CV for the entire dataset. These methods give a result of evaluation metrics such as accuracy, precision, recall and F1-score value of the model under testing.

Finally, A prototype for detecting hate speech is implemented using the selected best model then deployed using flask web services for it to be able to accept a new input post or comments and then return prediction the results.

CHAPTER SIX

6 Result and Discussions

In this chapter, the result of the experiments of the proposed solution for hate speech detection using machine learning is discussed. Here the result of the data annotation process and the two types of experiments to builds classification models for detection based on binary class and ternary class datasets are discussed. Finally, we discuss the significates of the result obtained by each experiment in this study.

6.1 Dataset Annotation Result

To select posts and comments to be annotated, the research utilizes a simple random sampling technique. This technique gives an equal chance for all the filtered posts and comments to be annotated. We then decide to annotate 5000 posts and comments. The size is limited to 5000 posts and comments due to the limitation of time for this research and resources for the annotation process for all filtered posts and comments.

The annotators were given 1500 posts and comments to label using the guideline. Three of the annotators have been given 500 similar instances and 1000 unique posts and comments. The same instances were given to calculate the inter-agreement between the annotators on the dataset. The fourth annotator is the researcher that oversees the whole process of the annotation and annotated 1500 unique post and comment. The whole process of building the dataset was tedious, challenging, and time-consuming; for this reason, we only consider the 500 same posts and comments to be annotated by three annotators and the researcher decided the final class using majority votes method.

The annotation process resulted in a distribution of classes shown in Table 6.1 for each annotator. The labeling result for the common 500 instances of posts and comments by the annotators presented in Table 6.2. The resulted of the three-class distribution in the dataset shown in Table 6.3. The dataset used to train the machine learning algorithms consists of 4500 uniquely annotated, and the final class 500 posts and comments decided using majority votes. A total of 5000 posts and comments are annotated and in the dataset.

Table 6. 1 Annotation Result of Unique Post and Comments

Label or Class	Annotator1	Annotator 2	Annotator 3	Annotator 4	Total unique annotated
Offensive (OFS)	484	628	304	619	2035
Hate (HS)	281	202	198	417	1098
Neither (OK)	235	170	498	464	1367
Total annotated	1000	1000	1000	1500	4500

Table 6. 2 Annotation Result of Common Post and Comments

Label	Annotator1	Annotator2	Annotator3	Final class by voting
Offensive (OFS)	251	227	221	264
Hate (HS)	93	114	151	95
Neither (OK)	156	159	128	141
Total	500	500	500	500

Table 6. 3 The Three-Class Distribution of The Dataset

Label or Class	Number of post and comments
Offensive (OFS)	2299
Hate (HS)	1193
Neither (OK)	1508
Total	5000

The annotator inter-rater agreement for 500 posts and comment that receives annotation from three annotators results in 0.54 Kappa. According to kappa value interpretation discussed in chapter three, it indicates that a moderate agreement between annotators.

In order to build the binary class dataset, the three-class dataset converted to two class datasets by considering all offensive language as hate speech. The result of the two-class distribution of the dataset shown in Table 6.4. All the OFS labeled converted to HS resulted in a dataset with 3492 HS class. The inter-annotators agreement for two-class on the 500 posts and comments resulted in 0.66 Kappa, which indicates a good agreement between the annotators.

Table 6. 4 The Binary Class Distribution of The Dataset

Label or class	Number of post and comments
Hate (HS)	3492
Non-Hate (NH)	1508

Finally, both kappa results show that the dataset contains a lot of ambiguity posts and comments because the same posts and comments differently labeled while having a guideline that specifies how to annotate the two most ambiguous class HS and OFS. However, the kappa results show a moderate and good agreement between annotator for ternary and binary datasets, respectively.

6.2 Feature Extraction Result

The features extraction process using the three methods N-gram, TD-IDF, and word2vec produced seven different sets of features vectors from the dataset. These features vectors are uses in the training of SVM, NB and RF models. Table 6.4 shows the extracted features vector size for the dataset.

Table 6. 5 Results of the Extracted Features Vectors Size

Features Extraction method	Features vectors size
Word unigram	10282
Word bigram	14298
Word trigram	14830
Word unigram + bigram + trigram (combined n-grams)	39410
TF-IDF	10279
TF-IDF + combined n-grams	49692
Word2vec	150

6.3 Models Evaluation Results

The Experiment resulted in twenty-one different models based on seven features and three classifiers for both binary and ternary classification. Testing these trained models is performed using 5-fold cross-validation. This method randomly split the dataset into five equal size sets. Train the models using 5-1 folds train and test using one remain fold. The process is iteratively for each fold. The obtained results are presented in binary and ternary Classification models Results below.

6.3.1 Binary Classification Models Evaluation Results

These classification models are built using the two-class dataset that is converted from the annotated three-class dataset that means the target classes are HS and NH. The trained models are tested using the 5-fold CV. The result of each test accuracy score for SVM, NB, and RF models based on extracted feature vectors are presented in tables separately below.

Table 6. 6 SVM Models Accuracy for Each Features using Binary class Dataset

Features	5-Folds Accuracy (%) for SVM model					Average Accuracy (%)
	1st	2nd	3rd	4th	5th	
Word unigram	72.12	70.72	68.10	73.27	73.07	71.46
Word bigrams	73.32	71.62	68.50	73.47	72.97	71.98
Word trigrams	73.22	71.52	68.40	73.47	73.07	71.94
Combined n-grams	72.82	70.72	68.20	72.27	73.47	71.50
TF-IDF	70.02	70.22	66.80	71.67	70.47	69.84
TF-IDF+ Combined n-gram	70.12	70.22	67.20	71.57	70.67	**69.96**
Word2vec	73.12	71.42	70.10	73.67	74.37	**72.54**

The results in Table 6.6 shows the CV accuracy scores for the SVM model based on feature with corresponding each fold test. The lower average accuracy of 69.84% results recorded on TF-IDF the feature model, and the higher accuracy 72.54% resulted using the word2vec feature.

Table 6. 7 NB Models Accuracy Scores on Each Features using Binary Class Dataset

Features	5-Folds Accuracy (%) for NB model					Average Accuracy (%)
	1st	2nd	3rd	4th	5th	
Word unigram	73.72	74.52	74.2	75.27	74.97	74.54
Word bigrams	74.62	74.22	72.80	75.97	75.37	74.60
Word trigrams	74.72	74.22	72.90	75.97	75.27	74.62
Combined n-grams	74.72	74.82	72.30	76.07	75.37	**74.66**
TF-IDF	70.62	71.52	71.40	73.97	73.77	72.26
TF-IDF+ Combined n-gram	73.12	72.72	71.90	75.27	75.37	73.68
Word2vec	71.32	67.33	69.5	72.97	72.77	**70.78**

The results in Table 6.7 shows the prediction accuracy scores for the NB model on each feature extracted with corresponding each fold test. The lower average accuracy of 70.78% result recorded on the word2vec feature model and slightly higher accuracy of 74.66% recorded using the Combined n-gram feature.

Table 6. 8 RF Models Accuracy Scores on Each Features using Binary Class Dataset

Features	5-Folds Accuracy (%) for the RF model					Average Accuracy (%)
	1st	2nd	3rd	4th	5th	
Word unigram	74.44	72.12	72.80	73.97	72.87	73.24
Word bigram	73.82	71.62	71.80	74.17	72.57	72.80
Word trigram	73.52	71.92	72.10	73.67	73.47	72.92
Combined n-gram	73.32	72.02	72.40	74.97	71.87	72.92
TF-IDF	74.52	71.92	71.7	73.87	71.97	72.79
TF-IDF+ combined n-gram	71.23	71.63	70.60	71.77	72.27	**71.50**
Word2vec	75.32	76.12	74.3	76.07	75.17	**75.39**

The results in Table 6.8 shows the prediction accuracy scores for the RF model on the extracted features with corresponding each fold test. The lower average accuracy of 71.5% results recorded on the TF-IDF with combined n-gram features. The higher accuracy of 75.39% recorded model using word2vec features.

The 5-Fold CV models testing for each extracted feature result for binary classification are summarized and illustrated on the bar chart. The bar chart in Figure 6.1 shows the visuals representation of the CV average accuracy of each model based on the features in the above three tables for binary class experiments. The blue represents the average classification accuracy of SVM models, the orange represents the average classification accuracy of NB models and the gray represents the average classification accuracy of RF models. The x-axis represented with the extracted feature and y-axis is based on the average accuracy result. Which shows the average accuracy of the NB models slightly greater than that of the SVM and RF models based on n-grams feature and TF-IDF combined with n-grams. So the RF obtains higher accuracy using word2vec and TF-IDF feature.

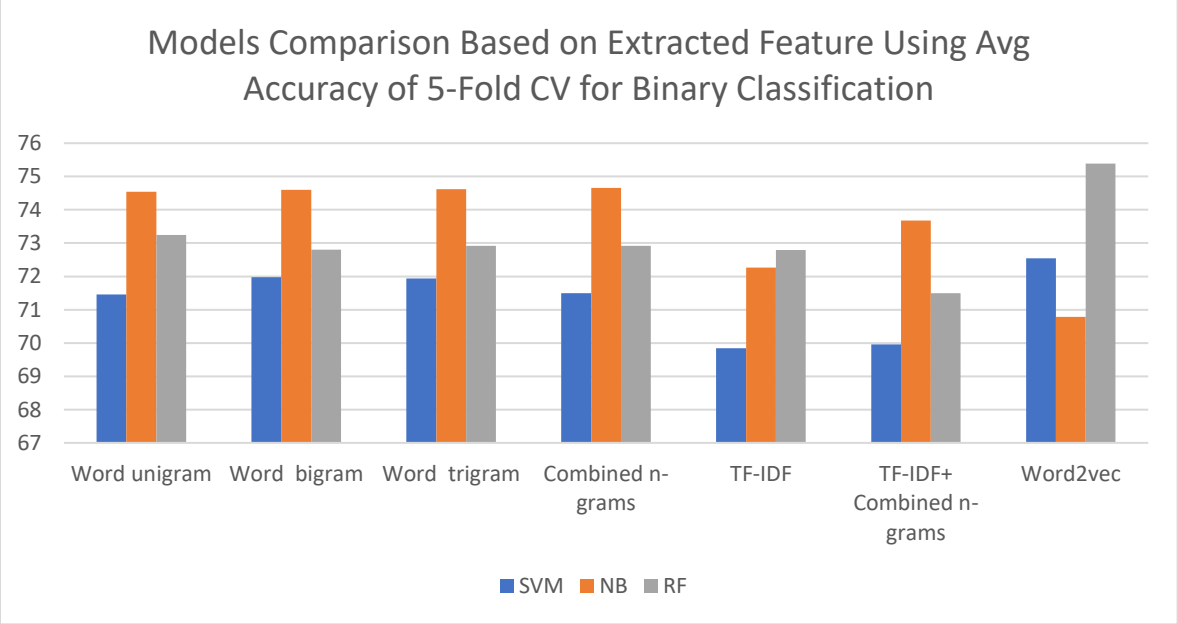

Figure 6. 1 Binary Models Comparisons using CV Avg Accuracy

Besides the result accuracy score obtained by five-fold CV. the study also uses other model's performance evaluation metrics, as discussed in chapters three and four. These metrics are Precision(P), Recall(R), and F1-score(F1). Table 6.9 shows the results of the evaluation metric for

each model based on features extracted. Also, use normalized confusion matrix of the models using a five-fold prediction result shown in figures 6.2 to 6.4 below.

Table 6. 9 Classification Performance Result of Binary Models

Feature	SVM			NB			RF		
	P	R	F1	P	R	F1	P	R	F1
Word unigram	72	71	72	73	75	71	71	73	70
Word bigrams	72	72	72	73	75	72	71	73	70
Word trigrams	72	72	72	73	75	72	71	73	70
Combined n-grams	72	71	72	73	75	72	70	72	70
TF-IDF	70	70	70	70	72	71	71	73	70
TF-IDF+ combined n-grams	70	71	70	72	74	72	70	72	70
Word2vec	76	73	**73**	73	71	72	75	75	**72**

Based on the F1-score, the SVM model based on word2vec obtains a higher score of 73% than RF and NB models. However, the accuracy of the RF model is 75.39% higher than SVM and NB using word2vec features see Figure 6.1. F1- score metrics selected to compare the models based on the features because its more useful than accuracy when the dataset contains uneven class distribution as described in chapter three.

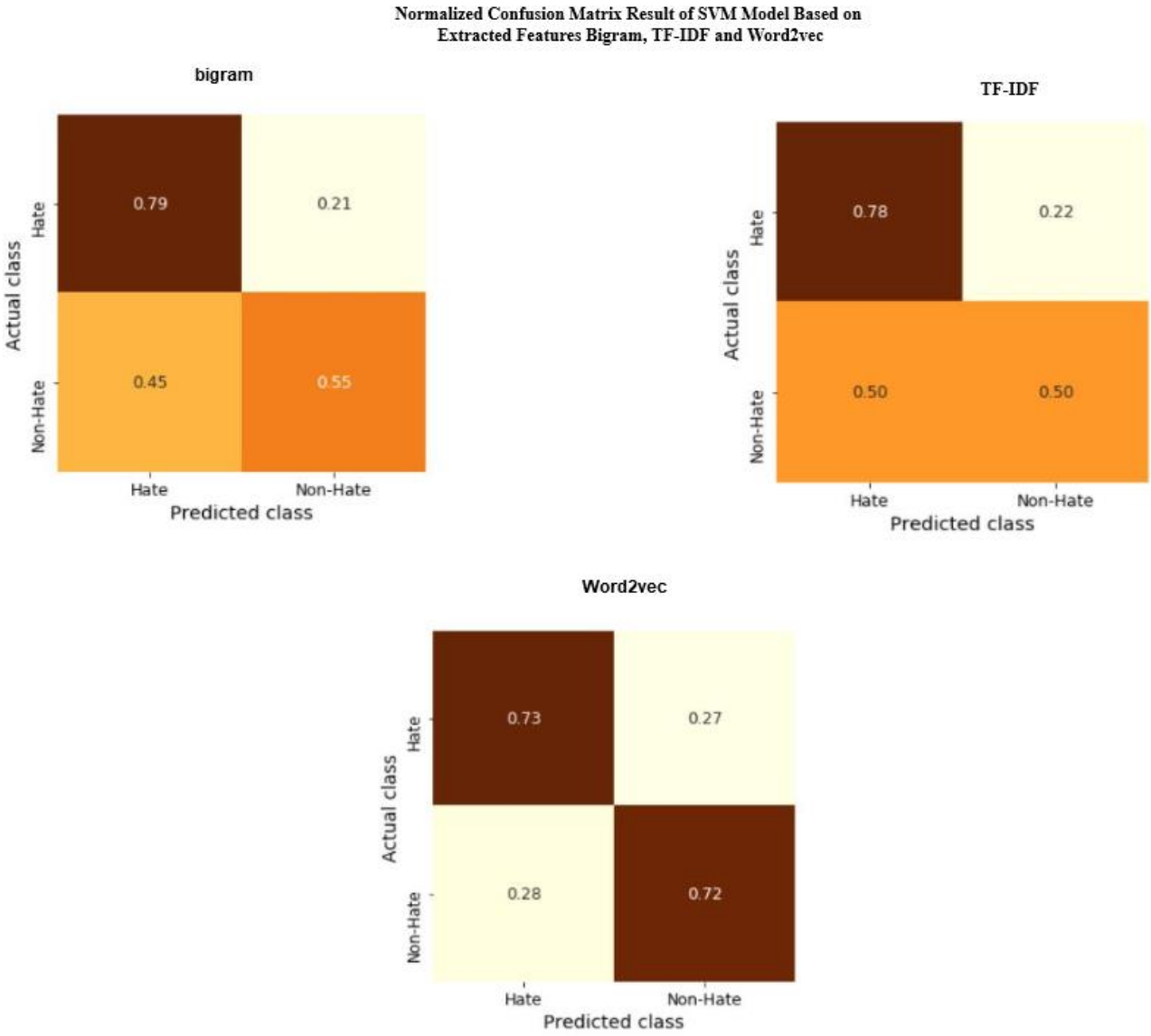

Figure 6. 2 Confusion Matrix of Sample Binary SVM Models Based on Extracted Features

Figure 6.2 illustrates the sampled confusion matrix for SVM models. SVM with bigram classify 79% of HS and 55% of NH correctly, and also 21% of HS and 45 % of NH are misclassified, SVM with TF-IDF classify 78% of HS and 50% of NH correctly and also 22% of HS and 50 % of NH are misclassified, and SVM with word2vec classify 73% of HS and 72% of NH correctly and also 27% of HS and 28 % of NH are misclassified.

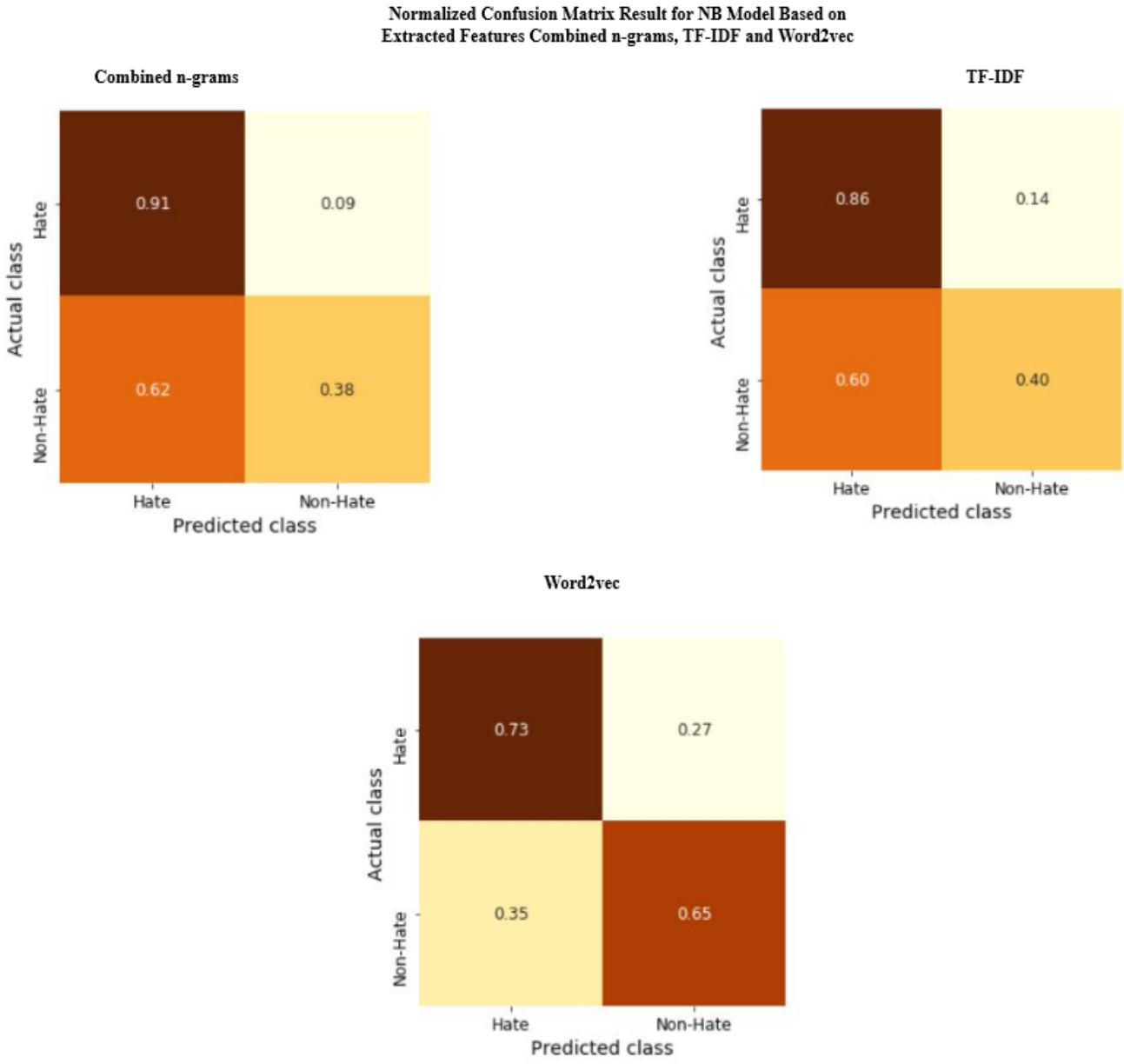

Figure 6. 3 Confusion Matrix of Sample Binary NB Models Based on Extracted Features

Figure 6.3 illustrates the sampled confusion matrix for NB models. NB with combined n-grams classify 91% of HS and 38% of NH correctly but 9% of HS and 62 % of NH are misclassified, NB with TF-IDF classify 86% of HS and 40% of NH correctly but 14% of HS and 60 % of NH are misclassified, and NB based on word2vec classify 73% of HS and 65% of NH correctly, but 27% of HS and 35 % of NH are misclassified.

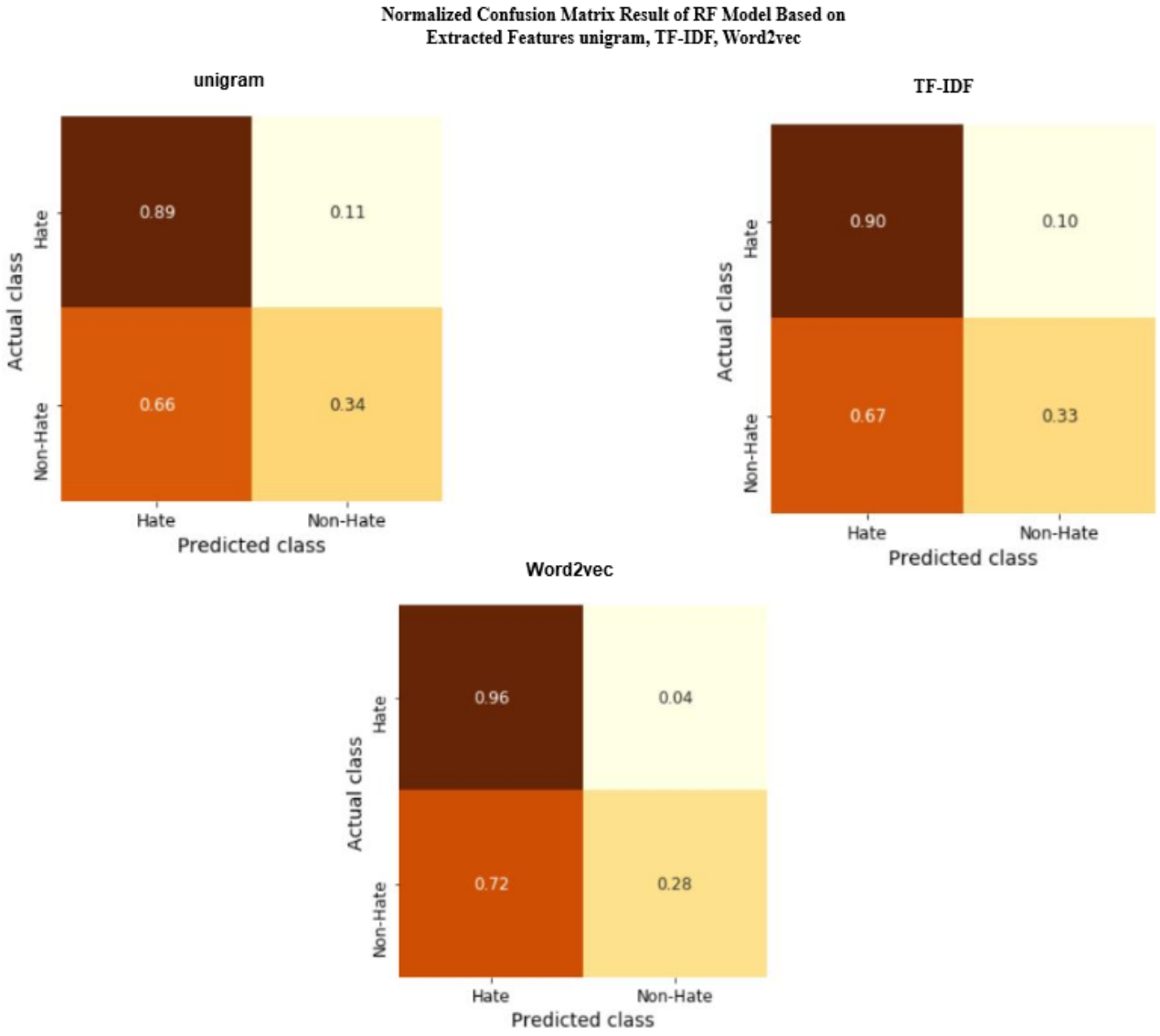

Figure 6. 4 Confusion Matrix of Sample Binary RF Models Based on Extracted Features

Similarly, Figure 6.4 shows the confusion matrix of the RF model. RF with unigram classify 89% of HS and 34% of NH correctly but 11% of HS and 66 % of NH are misclassified, RF with TF-IDF classify 90% of HS and 33% of NH correctly but 10% of HS and 67 % of NH are misclassified, and RF with on word2vec classify 96% of HS and 28% of NH correctly, but 4% of HS and72 % of NH are misclassified.

Normalized Confusion matrix for binary classifier models based on the feature extracted using the prediction result of 5-fold CV for the models shows in Figures 6.2, 6.3, 6.4, respectively. The actual class is the labels post or comment in the dataset and the predicted class is the prediction

labels made by the models. The heatmap represents the predicted or classified instance of posts and comments in each class.

Finally, the result of binary models demonstrates that the NB model based on the n-grams models shows better accuracy than the RF and SVM. Also, RF models based on word2vec and TF-IDF show higher accuracy than SVM and NB models. However, SVM models with word2vec give better classification results than both models.

6.3.2 Ternary Classification Models Evaluation Results

The models built using the prepared three-class dataset. The trained models are tested using the 5-fold CV. The result of each trained model testing accuracy score presented for SVM, NB, and RF, respectively, with feature vectors extracted in the tables below.

Table 6. 10 SVM Models Accuracy Scores on Each Features using Three Class Dataset

Extracted Feature	5-Folds accuracy scores (%) for SVM model					Average Accuracy (%)
	1st	2nd	3rd	4th	5th	
Word unigram	52.64	51.74	46.85	50.65	52.50	50.88
Word bigrams	53.54	51.14	48.45	50.65	52.40	51.24
Word trigrams	53.34	51.74	48.65	50.95	52.23	**51.40**
Combined n-grams	53.34	53.24	48.35	50.05	52.80	51.56
TF-IDF	50.24	50.04	45.45	48.04	48.79	48.51
TF-IDF+ combined	50.84	50.44	45.45	47.84	49.98	48.73
Word2vec	57.14	55.74	49.45	51.45	53.01	**53.35**

The results in Table 6.10 shows the accuracy scores for the SVM models on each feature with corresponding each fold test. The lower average accuracy of 48.51% results recorded on TF-IDF the feature and the higher accuracy 53.35% resulted using the word2vec feature.

Table 6. 11 NB Models Accuracy Scores on Each Features using Three Class Dataset

Feature models	5-Folds accuracy scores (%) for NB model					Average Accuracy (%)
	1st	2nd	3rd	4th	5th	
Word unigram	40.15	42.75	44.25	38.83	43.48	**41.89**
Word bigrams	41.75	44.05	47.15	42.64	43.78	43.87
Word trigrams	41.85	44.45	47.55	42.74	43.78	44.07
Combined n-grams	52.74	51.44	48.75	47.14	48.59	**49.73**
TF-IDF	48.35	48.25	47.35	45.94	47.39	47.45
TF-IDF+ combined	49.45	49.25	48.35	46.94	47.99	48.39
Word2vec	54.04	51.04	49.45	46.04	47.29	49.57

The results in Table 6.11 shows each fold's accuracy scores for the multi-class NB models based on the feature. The lower average accuracy of 41.89% recorded using the unigram feature and the higher accuracy 49.57% recorded using the word2vec feature.

Table 6. 12 RF Models Accuracy Scores on Each Features using Three class Dataset

Extracted Feature	5-Folds accuracy scores (%) for the RF model					Average Accuracy (%)
	1st	2nd	3rd	4th	5th	
Word unigram	52.64	50.64	50.44	52.75	48.89	51.07
Word bigrams	52.94	50.44	51.14	50.95	48.69	**50.08**
Word trigrams	52.44	51.74	48.95	51.35	48.99	50.69
Combined n-grams	53.34	50.54	50.64	50.85	48.39	50.75
TF-IDF	52.24	50.04	48.15	52.15	49.29	50.38
TF-IDF+ combined	52.94	50.14	49.65	49.94	50.30	50.59
Word2vec	56.04	56.74	53.04	54.25	55.21	**55.05**

The results in Table 6.12 shows each fold's accuracy scores for the multi-class RF models based on the extracted feature. The lower average accuracy of 50.08% recorded in the bigram feature and the higher accuracy 55.05% recorded on using the word2vec feature.

The bar chart in Figure 6.5 shows the visuals representation of the CV average accuracy of each model based on the features in the above three tables for ternary classifications experiments. The represented axis and colors are the same as figure 6.1. The bar chart shows that NB models result in a lower score than SVM and RF. SVM model achieves a higher score on bigram, trigram and combined n-gram. However, RF has the highest score using TD-IDF and word2vec based features.

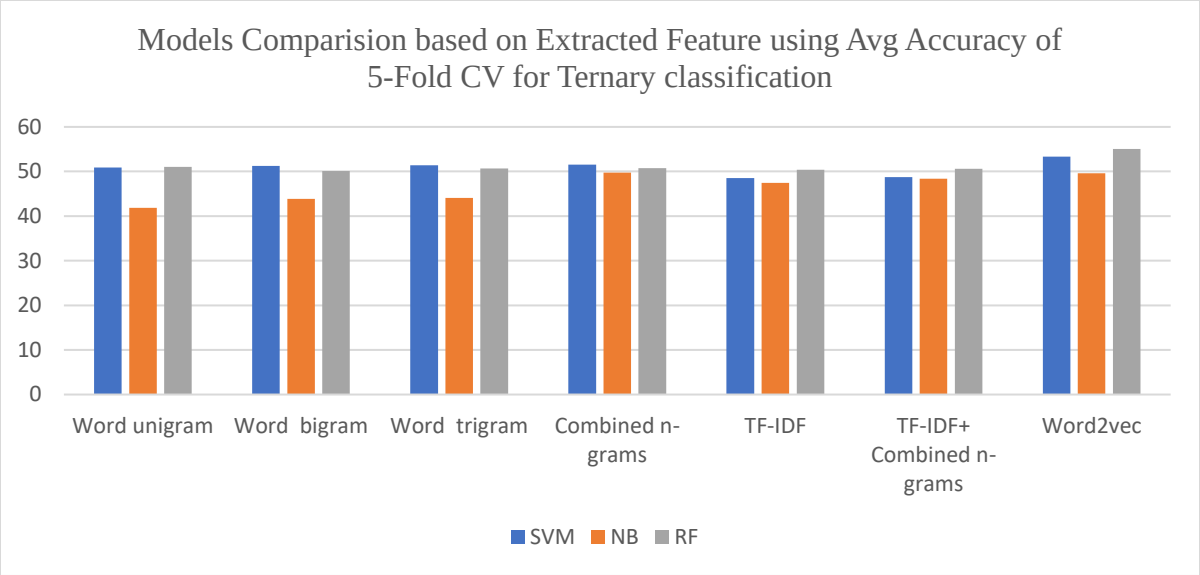

Figure 6. 5 Ternary Models Comparisons using CV Avg Accuracy

Table 6.13 shows the results of the model's evaluation metric for each Ternary classification models based on features extracted. Also, the normalized confusion matrix of the models using five-fold CV prediction results shown in figures 6.6 to 6.8 below.

Table 6. 13 Classification Performance Result of Ternary Models

Extracted Feature	SVM			NB			RF		
	P	R	F1	P	R	F1	P	R	F1
Word unigram	51	51	51	43	42	42	51	51	50
Word bigrams	51	51	51	45	44	44	51	51	50
Word trigrams	51	51	51	45	44	44	51	51	50
Combined n-grams	52	52	52	52	50	50	50	51	50
TF-IDF	49	49	49	50	47	48	50	51	50
TF-IDF+ combined n-grams	49	49	49	50	48	49	50	51	50
Word2vec	53	53	**53**	51	50	49	57	54	**50**

Comparing the result of F1-score for the models based on the features. SVM model based on a word2vec F1-score of 53% a higher score than the rest of the models. However, comparing the accuracy score of RF 55.5% highest than the NB and SVM model with features.

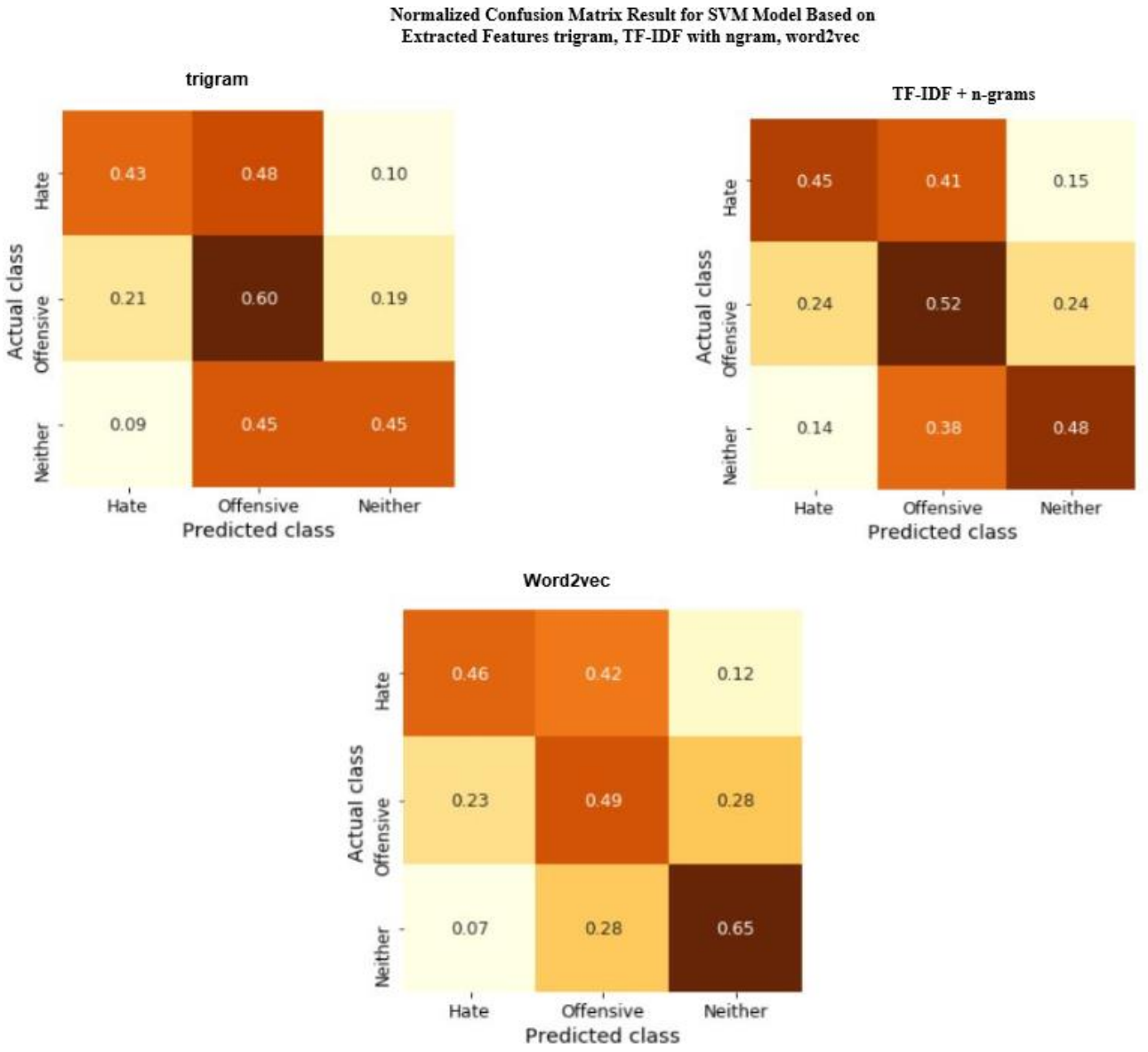

Figure 6. 6 Confusion Matrix of Sample Ternary SVM Models Based on Extracted Features

Figure 6.6 shows the confusion matrix of the SVM models. SVM with trigram classifies 43% of HS, 60% of OFS and 45% of Neither (OK) class correctly, but 48% of HS and 45% of OK class misclassified as OFS. A misclassification between HS and the OK class is 10% and 9%, respectively. SVM with TF-IDF+n-grams classifies 45% of HS, 52% of OFS and 48% of OK class

correctly but 41% of HS and 38% of OK class misclassified as OFS. A misclassification between HS and the OK class is 15% and 14%, respectively. Similarly, SVM with word2vec classifies 46% of HS, 49% of OFS and 65% of OK class correctly, but 42% of HS and 28% of OK class misclassified as OFS. A misclassification between HS and OK class are12% and 7%, respectively.

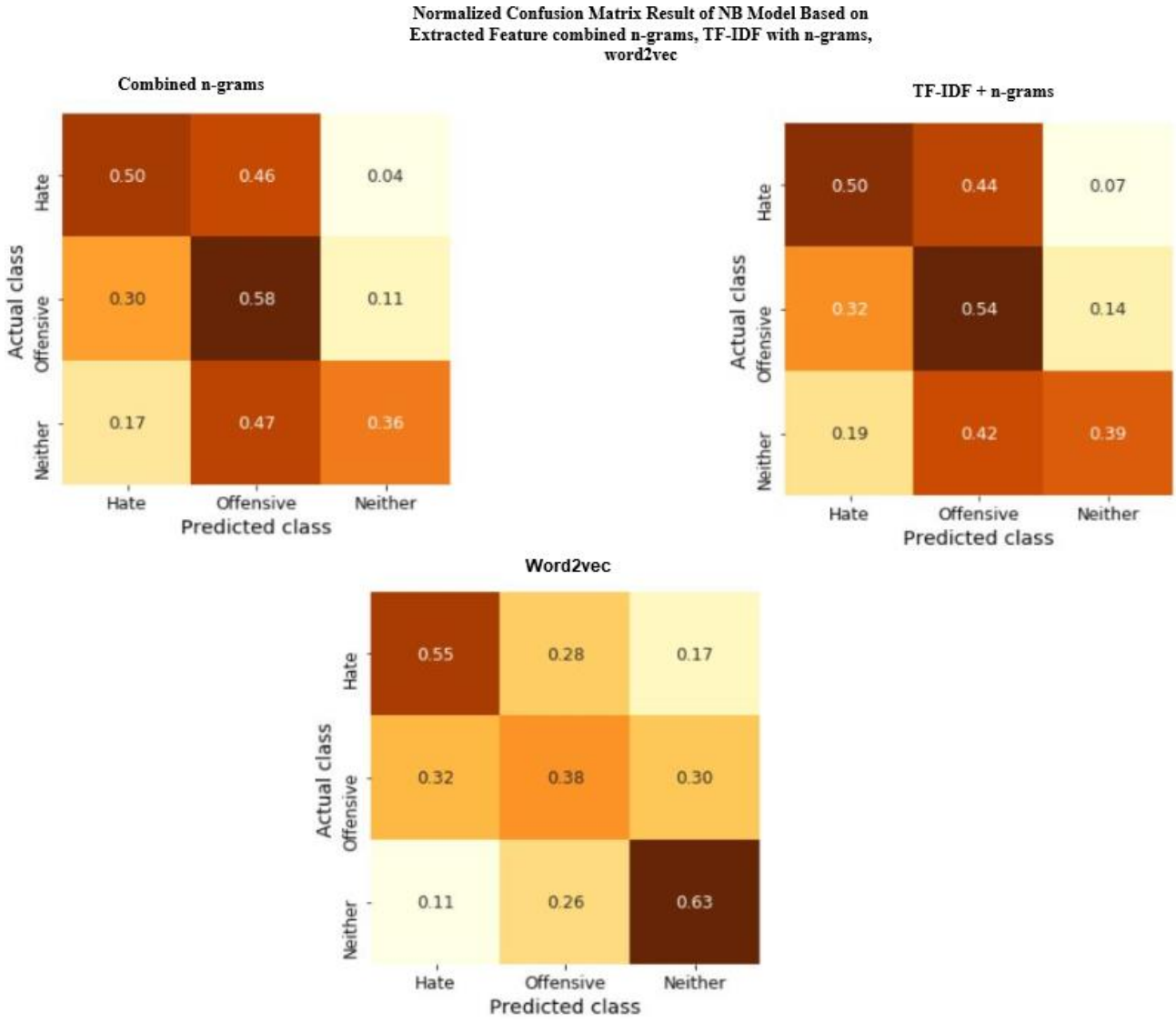

Figure 6. 7 Confusion Matrix of Sample Ternary NB Models Based on Extracted Features

Figure 6.7 shows a confusion matrix of the NB models. NB with combined n-gram classifies 50% of HS, 58% of OFS and 36% of OK class correctly but 46% of HS and 47% of OK class misclassified as OFS. A misclassification between HS and OK class 4% and 17%, respectively. NB with TF-IDF+n-grams classifies 50% of HS, 54% of OFS and 39% of OK class correctly, but

44% of HS and 42% of OK class misclassified as OFS. A misclassification between HS and OK class 7% and 19%, respectively. Also, NB with word2vec classifies 55% of HS, 38% of OFS and 63% of OK class correctly, but 28% of HS and 26% of OK class misclassified as OFS. A misclassification between HS and OK class 11% and 17%, respectively.

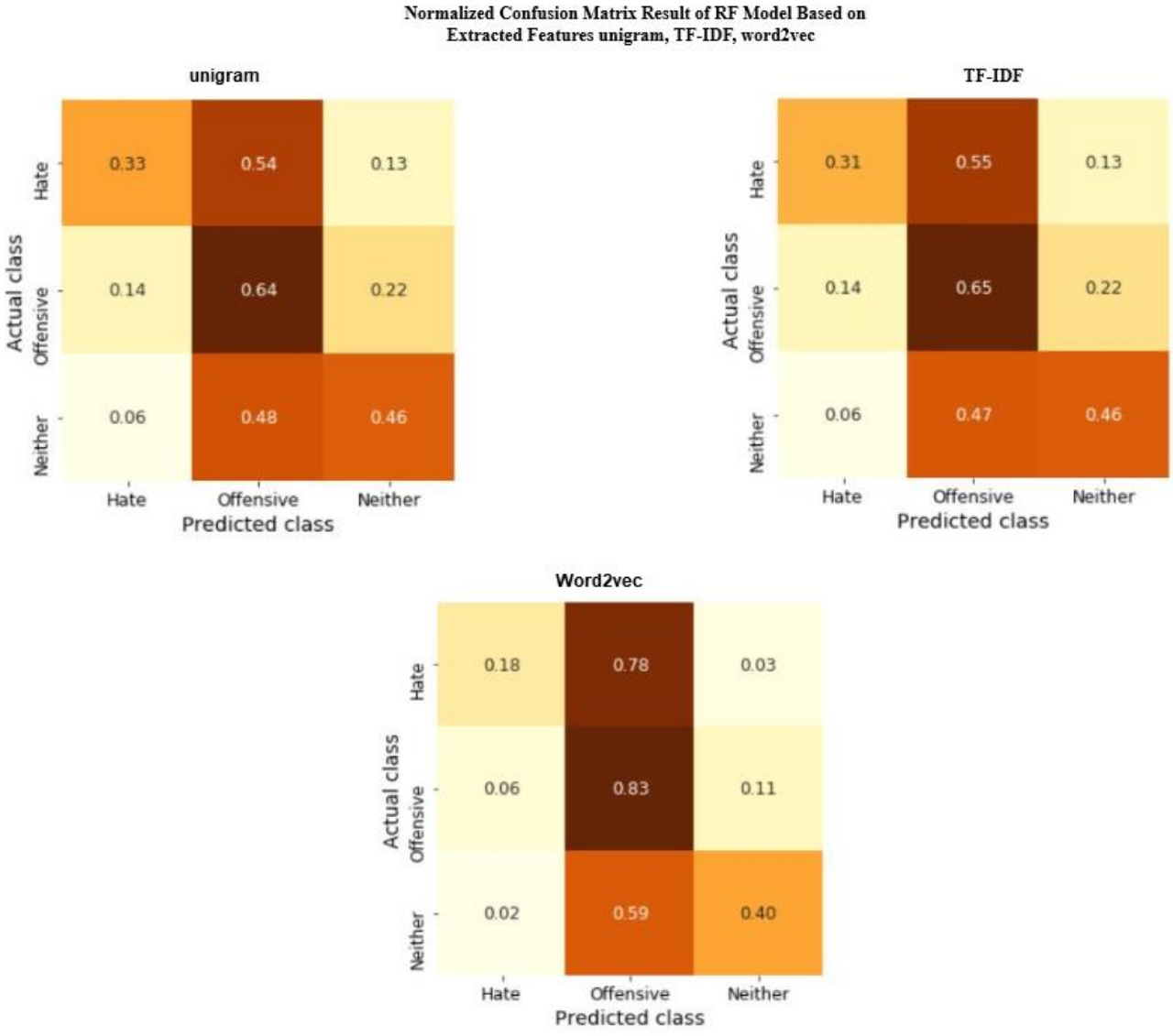

Figure 6. 8 Confusion Matrix of Sample Ternary RF Models Based on Extracted Features

Similarly, Figure 6.8 shows the confusion matrix of RF models. RF with unigram classifies 33% of HS, 64% of OFS and 46% of OK class correctly but 54% of HS and 48% of OK class misclassified as OFS. A misclassification between HS and OK class 6% and 13%, respectively. RF with TF-IDF classifies 31% of HS, 65% of OFS and 46% of OK class correctly but 55% of HS

and 47% of OK class misclassified as OFS. A misclassification between HS and OK class 6% and 13%, respectively. Also, RF with word2vec classifies 18% of HS, 83% of OFS and 40% of OK class correctly, but 78% of HS and 59% of OK class misclassified as OFS. A misclassification between HS and OK class 3% and 2%, respectively.

Normalized Confusion matrix for ternary classifier models based on the feature extracted in Figures 6.6, 6.7, 6.8 respectively shows the classification result of a 5-fold CV of each model. The actual class is the labels post or comment in a dataset and the predicted class is the prediction labels by the models the class are Hate, Offensive, and Neither (OK). Finally, SVM with word2vec performs slightly proper classification than the NB and RF model.

6.4 Discussions

The binary and ternary models used for comparing the performance with the previous work performed by Mossie and Wang[1] that used binary class datasets for detecting Amharic hate speech and also to compare the ability of models to detect the hate speech based on the ternary and binary class datasets. This study experimented with seven different feature extraction and machine learning algorithms SVM, NB, and RF to model hate speech detection using both datasets evaluated by 5-fold CV.

Initially, Comparing the inter-agreement for the binary class dataset 0.66 kappa value to the Mossie and Wang work [1], according to their claim that they obtain 0.64 kappas for 1821 binary class labeling by six annotators without a detailed guideline for labeling. The close range of the kappa result shows that the whole annotation process of hate speech content is complicated. It also shows that the subjective nature of the annotation process affects the agreement level not the size nor the number of annotators. When we looked at the 0.54 kappa results of the ternary class inter-agreement, it indicates that the number of class to be annotated affected the agreement level between the annotators and adds more ambiguity to the annotation process.

Feature extraction methods an important process to capture patterns from dataset to models using a machine learning algorithm. The result of this study shows that n-gram and word2vec perform better accuracy than TF-IDF features for both binary and ternary SVM models. However, the binary NB models-based n-grams resulted in higher accuracy than the SVM and RF, but preform

less on the ternary models. On the other hand, models based on word2vec perform a better classification result than both feature due to the ability to capture the similarity of a word in a text and use it as a feature to train models. The resulted word2vec features are substantially better than both TF-IDF and n-gram. This claim also shared by most research that used word2vec as a feature extraction method.

Binary detection models developed by Mossie and Wang [1] using NB and RF with the extracted feature word2vec and TF-IDF. They reported performance accuracy results of 79% and 73% for NB and 65% and 63% for RF with both features, respectively. They used a total of 6,120 instances as dataset among those 1821 are labeled post and comments and a dictionary of hateful word and phrase that extracted from the annotated dataset. The dataset contains 3296 non-hate and 2824 hate speech and also use 80% for training and 20% for testing.

Even though it is not recommended to compare models with a different dataset and experimental setup, nevertheless, comparing the NB and RF binary models with word2vec and TF-IDF features based on accuracy results only shows that the RF models of Mossie and Wang [1] perform better than RF models of this study by 4% and 1% margin for both feature respectively. However, the NB models of this study perform better than the previous NB models by a 7% margin for both features. This small margin difference of accuracy results obtained by both studies for both binary NB and RF models with the same feature extraction method shows that the size of the dataset nor the setup used is not the main problem. The problems are the ambiguity of dataset labels, which resulted in a more significant percentage of non-hate class to be misclassified as hate by the models. However, the SVM models with word2vec used in this study show a better classification performance than NB and RF models.

The ternary detection models developed to address the problem of many studies that consider hate speech problem as a yes or no binary class solution. We can argue that hate speech problems should not be considered as a binary class solution because there are other types of level of speech that are not constituted in the binary category of hate and non-hate speech.

The ternary models using three-class datasets the result obtained show that RF models with word2vec show higher accuracy than the SVM and NB models. However, SVM models with word2vec perform better classification performance than the NB and RF models. SVM performs

well in both binary and ternary models due to the ability model to handle an uneven distribution of class in the dataset. Regardless of the fact of the ternary model's low accuracy score, the SVM models able to avoid the majority misclassification between hate and neither (OK) class. However, the models misclassify the hate and OK class to the offensive (OFS) class and also the OFS class to hate and OK classes. It also indicates that the models are more biased to classifying hate and offensive the post and comments. The binary models have higher accuracy than the ternary model. However, the ternary models show promising performance by reducing the number of non-hate posts and comments classified as hate speech. This result indicates that to its better to have more categories or levels of speech to address the hate speech problem.

The problem of misclassification posts and comments occurs in both proposed models, as we have discussed earlier. This occurs due to several reasons but the major one is the complex nature of hate speech which means the existing laws, procedure, and guideline use to separate hate speech from other types of speech are inefficient to describe what constitutes hate speech and lack a clean-cut rule to separate this speech also cause the whole process to be subjective. This problem is shown on the annotator's inter-agreements value, which indicates that ambiguity occurs in labeling these types of speech. The other reasons are typos errors in posts and comment text cause the models to fail to incorporate essential features, and also the proposed features extraction methods somewhat depended on the occurrence of a word in the posts and comments could cause incorrect features to be used by the models to classify this posts and comments erroneously. Finally, these discussed problems reflected in the classification performance of the developed models.

CHAPTER SEVEN

7 Conclusion, Recommendation and Future work

In this chapter, the proposed hate speech detection using machine learning and the finding of this research is concluded. It also described the recommendation and future research directions.

7.1 Conclusion

This research proposed to develop a solution 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 Amharic language.

To successfully execute the study, it was essential to understand and define hate and offensive speech on social media, explore existing various techniques used to tackle the problem and understand the Amharic language, as discussed in chapter two. Also, the different methods followed to implement and design models that have the capability of detecting hate speech. These methods include collecting post and comment for building the dataset, develop annotation guidelines, preprocessing, features extraction using n-gram, TF-IDF and word2vec, models training using SVM, NB, and RF, and models testing. Finally performs models' comparisons based on 5-fold CV evaluation metric results.

In this study, we manually annotated the posts and comments into three classes of hate (HS), offensives (OFS) and neither (OK) speeches. The annotated dataset converted into two classes labels dataset by converting all OFS to HS classes. This process resulted in two datasets with 5000 instances of posts and comments one with binary classes and the other with ternary classes dataset.

Based on the two datasets, the models developed using SVM, NB, and RF with seven feature extraction methods implemented. The experiment performed using these two datasets resulted in 21 binary and ternary models for each dataset. The binary models using RF with word2vec resulted in better accuracy than both SVM and NB. However, SVM with word2vec performs better in classification results with 73% F1-score, and it shows better performance than models based on NB and RF.

The ternary models perform better in handling misclassification between the hate and non-hate posts and comments than the binary models. Also, the ternary SVM model with word2vec resulted in 53% F1-score, which shows better performance than the models with NB and RF.

Finally, the models based on SVM using word2vec slightly better performance than NB and RF models in the case of both datasets used in this research. So using ternary classes dataset is one of the first experiments to the best of our knowledge to use in Amharic hate speech detection study.

7.2 Recommendations

Since the detection models have shown some capability of detecting hate speech for Amharic language using the small dataset with a lot of vague labeling process by annotators of posts and comments. As a result, the researcher recommends that a social media company to develop models that can improve the flagging system or assist a human content moderator by reducing the number of posts or comments that they have to go through to detect hateful content on their platform. Also, we recommend that lawmaker or any stockholder that participates in making of hate speech laws and policy to provide an ambiguous rule and regulation to tackle the problem.

7.3 Future works

The proposed solution used a supervised learning algorithm with a text mining feature extraction method used to builds models. It is better to see the difference in performance results using unsupervised or deep learning algorithms models.

Next, in Ethiopia, hate speech can be expressed on social media using more one language. This study limited only on Amharic language, but they are more than 80 different languages used in the country. For a more comprehensive detection model, future research can focus on developing datasets and models for the dominant language used in the country, such as Oromo, Somali, Tigrinya, and other languages that are used on social media platforms. Additionally, posts and comments often contain non-textual content images, emojis, and videos, which may also contain hateful expressions. Future research could focus on such type of contents.

References

[1] Z. Mossie and J.-H. Wang, "Social Network Hate Speech Detection for Amharic Language," in *Computer Science & Information Technology-CSCP*, 2018, pp. 41–55.

[2] H. Watanabe, M. Bouazizi, and T. Ohtsuki, "Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection," *IEEE Access*, vol. 6, pp. 13825–13835, 2018.

[3] B. Gambäck and U. K. Sikdar, "Using Convolutional Neural Networks to Classify Hate-Speech," *Proc. ofthe First Work. Abus. Lang. Online*, no. 7491, pp. 85–90, 2017.

[4] Biere Shanita and S. Bhulai, "Hate Speech Detection Using Natural Language Processing Techniques," Vrije university Amsterdam, 2018.

[5] J. B. Gagliardone, Iginio, Matti Pohjonen, Zenebe Beyene, Abdissa Zerai, Gerawork Aynekulu, Mesfin Bekalu, "MECHACHAL :Online debates and elections in Ethiopia. From hate speech to engagement in social media.," Oxford & Addis Ababa university, 10.2139/ssrn.2831369, 2016.

[6] I. Gagliardone, A. Patel, and Matti Pohjonen, "Mapping and Analysing Hate Speech Online: Opportunities and Challenges for Ethiopia," Oxford & Addis Ababa university , 10.2139/ssrn.2601792, 2014.

[7] S. Malmasi and M. Zampieri, "Detecting Hate Speech in Social Media," *arXiv Prepr.*, no. arXiv:1712.06427, pp. 1–7, 2017.

[8] Z. Zhang and L. Luo, "Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter," *arXiv Prepr.*, vol. 1, no. arXiv:1803.03662, pp. 1–5, 2018.

[9] A. Al-Hassan and H. Al-Dossari, "Detection of Hate Speech In Social Networks: A Survey On Multilingual Corpus," *CS IT-CSCP*, pp. 83–100, 2019.

[10] B. Mathew, R. Dutt, P. Goyal, and A. Mukherjee, "Spread of hate speech in online social media," *arXiv Prepr.*, no. arXiv:1812.01693, 2018.

[11] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated Hate Speech Detection and the Problem of Offensive Language," *arXiv Prepr.*, no. arXiv:1608.08738, 2017.

[12] K. Worku, "Automatic Amharic text news classification : Aneural networks approach," *Ethiop. J. Sci. Technol*, vol. 6, no. 2, pp. 127–135, 2013.

[13] "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: 17-Nov-2018].

[14] S. Tulkens, L. Hilte, E. Lodewyckx, B. Verhoeven, and W. Daelemans, “A Dictionary-based Approach to Racism Detection in Dutch Social Media,” *arXiv Prepr.*, no. arXiv:1608.08738, 2016.

[15] I. Alfina, R. Mulia, M. I. Fanany, and Y. Ekanata, “Hate speech detection in the Indonesian language: A dataset and preliminary study,” *2017 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSIS 2017*, vol. 2018-Janua, no. October, pp. 233–237, 2018.

[16] M. A. Fauzi and A. Yuniarti, “Ensemble method for Indonesian twitter hate speech detection,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 11, no. 1, pp. 294–299, 2018.

[17] F. Poletto, M. Stranisci, M. Sanguinetti, V. Patti, and C. Bosco, “Hate speech annotation: Analysis of an Italian twitter corpus,” *CEUR Workshop Proc.*, vol. 2006, 2017.

[18] F. Del Vigna, A. Cimino, F. Dell’Orletta, M. Petrocchi, and M. Tesconi, “Hate me, hate me not: Hate speech detection on Facebook,” *Proc. First Ital. Conf. Cybersecurity*, no. January, pp. 1–10, 2017.

[19] P. Fortuna and S. Nunes, “A Survey on Automatic Detection of Hate Speech in Text,” *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–30, 2018.

[20] U. Nations and I. Introduction, “I.8. International Convention on the Elimination of All Forms of Racial Discrimination (Excerpts),” *Basic Doc. Int. Migr. Law*, vol. 7, no. 7, pp. 26–28, 2013.

[21] Council of Europe, “Recommendation No. R (97) 20 of the Committee of Ministers to Member States on ‘Hate Speech,’” *Coe.Int*, no. 97, pp. 106–108, 1997.

[22] C. E. Ring, “Hate Speech in Social Media: An Exploration of the Problem and Its Proposed Solutions,” University of Colorado, 2013.

[23] Facebook, “Objectionable Content:Hate speech,” 2019. [Online]. Available: https://www.facebook.com/communitystandards/objectionable_content. [Accessed: 05-Mar-2019].

[24] Twitter, “Hateful conduct policy:Rules-and-policies,” 2019. [Online]. Available: https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy. [Accessed: 05-Mar-2019].

[25] Google, “youtube hate speech policy.” [Online]. Available: https://support.google.com/youtube/answer/2801939?hl=en. [Accessed: 05-Mar-2019].

[26] B. Raufi and I. Xhaferri, “Application of machine learning techniques for hate speech detection in mobile applications,” *2018 Int. Conf. Inf. Technol. InfoTech 2018 - Proc.*, no. 46116, pp. 1–4, 2018.

[27] K. K. Kiilu, G. Okeyo, R. Rimiru, and K. Ogada, “Using Naïve Bayes Algorithm in detection of Hate Tweets,” *Int. J. Sci. Res. Publ.*, vol. 8, no. 3, pp. 99–107, 2018.

[28] Z. Waseem and D. Hovy, “Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter,” in *In Proceedings of the NAACL student research workshop*, 2016, pp. 88–93.

[29] H. D. J. L. Njagi Dennis Gitari Zhang Zuping, “A Lexicon-based Approach for Hate Speech Detection,” *Int. J. Multimed. Ubiquitous Eng.*, vol. 10, no. 4, pp. 215–230, 2015.

[30] M. O. Ibrohim and I. Budi, “A Dataset and Preliminaries Study for Abusive Language Detection in Indonesian Social Media,” in *Procedia Computer Science*, 2018, vol. 135, pp. 222–229.

[31] N. Albadi, M. Kurdi, and S. Mishra, “Are they our brothers? analysis and detection of religious hate speech in the Arabic Twittersphere,” *Proc. 2018 IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Mining, ASONAM 2018*, pp. 69–76, 2018.

[32] B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky, and M. Wojatzki, “Measuring the Reliability of Hate Speech Annotations: The Case of the European Refugee Crisis,” *arXiv Prepr.*, no. 1701.08118, 2017.

[33] A. Schmidt and M. Wiegand, “A Survey on Hate Speech Detection using Natural Language Processing,” in *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, 2017, no. 2012, pp. 1–10.

[34] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, “Hate Speech Detection with Comment Embeddings,” in *In Proceedings of the 24th international conference on world wide web*, 2014.

[35] L. Silva and I. Weber, “Analyzing the Targets of Hate in Online Social Media,” in *In Tenth International AAAI Conference on Web and Social Media.*, 2016, no. March, pp. 687–690.

[36] W. Warner and J. Hirschberg, “Detecting Hate Speech on the World Wide Web,” *Proc. 2012 Work. Lang. Soc. Media*, pp. 19–26, 2012.

[37] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, “Deep Learning for Hate Speech Detection in Tweets,” in *Proceedings of the 26th International Conference on World Wide Web Companion*, 2017, no. 2.

[38] A. Gaydhani, V. Doma, S. Kendre, and L. Bhagwat, “Detecting Hate Speech and Offensive Language on Twitter using Machine Learning: An N-gram and TFIDF based Approach,” *arXiv Prepr.*, no. arXiv:1809.08651, 2018.

[39] tfidf.com, “TF-IDF: A Single-Page Tutorial - Information Retrieval and Text Mining,” 2019. [Online]. Available: http://www.tfidf.com/. [Accessed: 10-May-2019].

[40] S. Zimmerman, C. Fox, and U. Kruschwitz, “Improving hate speech detection with deep learning ensembles,” *Lr. 2018 - 11th Int. Conf. Lang. Resour. Eval.*, pp. 2546–2553, 2019.

[41] J. Lilleberg, Y. Zhu, and Y. Zhang, “Support vector machines and Word2vec for text

classification with semantic features," *Proc. 2015 IEEE 14th Int. Conf. Cogn. Informatics Cogn. Comput. ICCI*CC 2015*, pp. 136–140, 2015.

[42] P. Burnap and M. L. Williams, "Us and them : identifying cyber hate on Twitter across multiple protected characteristics," *EPJ Data Sci.*, 2016.

[43] N. Chetty and S. Alathur, "Hate speech review in the context of online social networks," *Aggress. Violent Behav.*, vol. 40, pp. 108–118, 2018.

[44] G. K. Pitsilis, H. Ramampiaro, and H. Langseth, "Effective hate-speech detection in Twitter data using recurrent neural networks," *Appl. Intell.*, vol. 48, no. 12, pp. 4730–4742, 2018.

[45] B. GAMBÄCK and U. K. SIKDAR, "Named Entity Recognition for Amharic Using Deep Learning," *IST-Africa*, pp. 1–6, 2017.

[46] G. Mezemir, "Automatic stemming for Amharic an Experiment Using Successor Variety Approach," Addis Ababa, 2009.

[47] D. Benikova, M. Wojatzki, and T. Zesch, "What does this imply? examining the impact of implicitness on the perception of hate speech," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10713 LNAI, pp. 171–179, 2018.

[48] "Social Media Stats Ethiopia | StatCounter Global Stats," 2019. [Online]. Available: http://gs.statcounter.com/social-media-stats/all/ethiopia/#monthly-201804-201904. [Accessed: 10-May-2019].

[49] "Social Media Demographics - Facebook," 2019. [Online]. Available: https://napoleoncat.com/stats/facebook-users-in-ethiopia/2019/04. [Accessed: 10-May-2019].

[50] P. Harrington, *Machine Learning in Action*, vol. 37, no. 3. 2012.

[51] "Random forests - classification description." [Online]. Available: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm. [Accessed: 18-Jul-2019].

[52] "Cohen's Kappa Statistic - Statistics How To." [Online]. Available: https://www.statisticshowto.datasciencecentral.com/cohens-kappa-statistic/. [Accessed: 25-Aug-2019].

[53] J. D. Novakovi, "Evaluation of Classification Models in Machine Learning," *Theory Appl. Math. Comput. Sci.*, vol. 7, no. 1, pp. 39–46, 2017.

[54] "ProClassify User's Guide - Cross-Validation Explained." [Online]. Available: http://genome.tugraz.at/proclassify/help/pages/XV.html. [Accessed: 26-Aug-2019].

[55] C. Manliguez, "Generalized Confusion Matrix for Multiple Classes," *Springer, Cham*, no. November, pp. 4–6, 2016.

Appendixes

Appendix A: Amharic Hate Speech Corpus Annotation Guidelines

The goal of this guideline to give a direction for annotating(labeling) posts and comments to the class of Hate, Offensive, or neither (OK) speech by their content. For this task, it is essential to define both Hate and Offensive speech with specific rules to be considered while annotating. The Specific rules presented to point out that we have aimed to a more inclusive and general definition of 'hate speech' with some other perspectives found in literature, laws and general recommendations. Also, we want to better describe hate speech and differentiate it from offensives speech.

Hate Speech

Hate Speech (HS): is language and expression or kind of writing that attacks or diminishes, that incites violence or hate against groups based on specific characteristics such as race, ethnic origin, religious affiliation, political view, physical appearance, gender or other characteristics. The term 'hate speech' in this guideline, besides the above definition, is being used to describe post and comment that constitutes a slur or an instance of written abuse against a wide range of targets.

To make it even more precise, these three characteristics are taken into account for the identification of a post and comment is hate speech those are:

1. **The target**: as the definition of hate speech state that, a target is a specific group with specific characteristics that belongs to the group. We consider following as target in this study.
    - Ethnicity
    - Political group and view
    - Religious
    - Gender
2. **An action**: a post or comment contains an action that suggests the following action:
    - spreads promote or justify hatred,
    - suggesting, inciting, or calling for threating violence.
    - Discriminating or dehumanizing.
    - Suggests killing, beating, evicting, or intimidating a target group.

3. Use “**Us vs. Them**" a verbal expression that references to the alleged inferiority or superiority of one target group concerning to other groups.

Therefore, we consider that a post or comment that has a joint presence of these characteristics must be marked as hate speech (HS). The following are specific rules for labeling hate post or comment.

a) A post or comment is considered HS if there is the reference that explicit. Incitement to discrimination or just implied to hostility or violence actions of any kind or actions list above to a target group,
b) A post or comment is considered HS, references to the alleged inferiority or superiority of some target groups concerning others or dissemination of ideas based on target group's superiority or inferiority, by whatever means. (Us Vs. Them).
c) A post or comment is considered HS if there is a combination of hateful expression, which is an insult, threats, or denigrating toward a target’s groups.
d) A post or comment is considered HS if its dehumanization or association the target group with animals or beings considered inferior on grounds in (b) above,
e) If the post or comment has direct target group slur or uses a direct word that the society denounced as hatred, hostile or disrespectful nickname of the target group.
f) A post or comment is considered HS Accusing or Condemning people based on their target groups.
g) A post or comment is considered HS if it contains stereotype which means over-generalized belief about a given target.

Offensive Speech

Offensive Speech (FS): can be defined as a language or expression that offends and negatively characterizes an individual or groups of people. This kind of speech causes someone to feel upset, annoyed, insulating, angry, hurt, and disgusting. It can occur with different linguistic styles, even humor or jokes.

As we mentioned above in the hate speech section, there are three characteristics that should be considered for labeling a post or comment to HS. However, if these characteristics are not detected

or if just one of these characteristics presented with less degree of impact, HS is assumed not to occur. This guideline assumes offensive speech (FS) occurs when there is a low degree of hate speech characterizations occur. Also, with the character of offensive speech, which is to make someone feel upset, annoyed, insulating, angry, hurt, and disgusting which may not refer to a wide range of target groups. If these cases detected we assume that FS occurred. Otherwise, the speech is normal (Ok). For a better understanding of FS Specific rules listed below. The following are specific rules for labeling offensive post or comments:

a) If post or comment contains insulting, dirty, disgusting, or upsetting words but not contain any action listed above.
b) If post or comment contain violent or insulting words but not possible to explicitly identify a target group in the post/comment
c) If post or comment described or considered the target as unkind or unpleasant people.
d) If the post or comment described or associated the target with a negative feature or quality typical human flaws.
e) If the post or comment contains or refers to a given target with mocking intent.
f) If the post or comment contains defamation, which is a false accusation person or attack on a person's character.
g) If a post or comment contains insulting and disgusting word quote for other people posts to condemn the posts or comments.

Note that **targets** in offensives speech Specific rules refer to individuals', a thing that the post or comment described and our target groups listed above.

Neither (OK) speech

Neither(ok): the that does not contain a character that described in both hate and offensives speech section. Finally, each post or comment should be marked with class labels using the definition and Specific rules as the number and the abbreviation can be used interchangeably.

1. **Neither** hate nor offensives = Ok = 0.
2. Offensives = OFS = 1
3. Hate speech = HS = 2

Appendix B: Sample Keyword Used for Filtering Post and Comments

Table B. 1 Sample Keyword Related to Target Group Used for Filtering Post and Comments

Ethnicity	Political	Religious	Gender
ሀራሪ	ህወሃት	ሀዋሪያት	ልጃገረድ
ሲዳማ	ስልጣን	ሃይማኖት	ሴት
ስልጤ	ብሄርተኛ	ሀጥያት	ሴቶች
ሶማሌ	ብአዴን	ሙስሊም	ወንድ
ራያ	አብን	ኦርቶዶክስ	ወንዶች
ትግሬ	ግንቦት ሰባት	ክርስቲያን	
አማራ	ኢህአዴግ	ኢስላም	
አፋር	ኦህዴድ	እምነት	
ኦሮሞ	አብነግ	እስልምና	
ከንባታ	አነግ	እግዚአብሔር	

Appendix C: A Sample Source Code

Appendix C.1 A Sample Code for Preprocessing

```
def preprocess(text_string):
```

```
    space_pattern = '\s+'
    url_regex = ('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|'
        '[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+')
    englishword_num ='[a-zA-Z]|[0-9]+'
    AmhPunc='[፤፡።፣:,.?/()•“”*፨]+'
    special_char = "[@#$%^&=?×!,;:_.(){}`'/+*<>\"¤— „\ ®¯™¡¡\x10»€«·‘0e1b§”¬¦…"
"f÷\~¨©±¥£¶–°•˜’“|]'
    geez_number=' [፩ ፪ ፫ ፬ ፭ ፮ ፯ ፰ ፱ ፲ ፳ ፴ ፵ ፶ ፷ ፸ ፹ ፺ ፻ ፼ ፽ ፾] ' # for removing geez number
    RE_EMOJI = re.compile('[\U00010000-\U0010ffff]', flags=re.UNICODE)
    Clean_text = re.sub(url_regex,'', text_string)
    Clean_text = re.sub(AmhPunc,' ',Clean_text)
    Clean_text=re.sub(special_char,' ',Clean_text)
    Clean_text= re.sub(englishword_num,'',Clean_text)
    Clean_text= re.sub(r'(.)\1+', r'\1\1',Clean_text) #removing elongation in text
    Clean_text = RE_EMOJI.sub(r'', Clean_text)
    Clean_text =re.sub(geez_number,'',Clean_text)
    Clean_text=re.sub('-','',Clean_text)
    Clean_text=re.sub(r'<[^>]*>','', Clean_text)
    Clean_text=re.sub('-',' ',Clean_text)
    Clean_text = Clean_text.replace("\\", "");
    Clean_text = Clean_text.replace("[", "");
    Clean_text = Clean_text.replace("]", "");
    Clean_text = re.sub(space_pattern,' ',Clean_text)
    return Clean_text
def remove_emoji(string):
    emoji_pattern = re.compile("["
                        u"\U0001F600-\U0001F64F"  # emoticons
                        u"\U0001F300-\U0001F5FF"  # symbols & pictographs
                        u"\U0001F680-\U0001F6FF"  # transport & map symbols
                        u"\U0001F1E0-\U0001F1FF"  # flags
                        u"\U00002702-\U000027B0"
                        u"\U000024C2-\U0001F251"
                        "]+", flags=re.UNICODE)
```

```
    emojifreetext = emoji_pattern.sub(r'', string)
    return emojifreetext
```

Appendix C.2 A Sample Code for Normalization

```
def normalization _char(Amhtext):
    haa='[ሃ]'
    he='[ሐ]'; hu='[ሑ]'; hi='[ሒ]'; ha='[ሓ]'; hie='[ሔ]'; h='[ሕ]'; ho='[ሖ]';
    he1='[ኀ]'; hu1='[ኁ]'; hi1='[ኂ]'; ha1='[ኃ]'; hie1='[ኄ]'; h1='[ኅ]'; ho1='[ኆ]';
# replace the ሐ with ሀ b/c they have same sound and ሓ ሃ ኃ to ሀ ሁ ሂ ሃ ሄ ህ ሆ
    replace_text= re.sub(he,'ሀ',Amhtext);
    replace_text= re.sub(hu,'ሁ',replace_text);
    replace_text= re.sub(hi,'ሂ',replace_text);
    replace_text= re.sub(ha,'ሀ',replace_text);
    replace_text= re.sub(hie,'ሄ',replace_text);
    replace_text= re.sub(h,'ህ',replace_text);
    replace_text= re.sub(ho,'ሆ',replace_text);
    replace_text= re.sub(haa,'ሀ',replace_text);
# replace the ኀ with ሀ b/c they have same sound
    replace_text= re.sub(he1,'ሀ',replace_text)
    replace_text= re.sub(hu1,'ሁ',replace_text)
    replace_text= re.sub(hi1,'ሂ',replace_text)
    replace_text= re.sub(ha1,'ሀ',replace_text)
    replace_text= re.sub(hie1,'ሄ',replace_text)
    replace_text= re.sub(h1,'ህ',replace_text)
    replace_text= re.sub(ho1,'ሆ',replace_text)
# replace the ሠ with ሰ b/c they have same sound
    se='[ሠ]'; su='[ሡ]'; si='[ሢ]'; sa='[ሣ]'; sie='[ሤ]'; s='[ሥ]'; so='[ሦ]';
    replace_text= re.sub(se,'ሰ',replace_text)
    replace_text= re.sub(su,'ሱ',replace_text)
    replace_text= re.sub(si,'ሲ',replace_text)
    replace_text= re.sub(sa,'ሳ',replace_text)
    replace_text= re.sub(sie,'ሴ',replace_text)
```

```
    replace_text= re.sub(s,'ስ',replace_text)
    replace_text= re.sub(so,'ሶ',replace_text)
# replace the አ with ዐ b/c they have same sound also ኣ and ዓ
    aa1='[ኣ]'
    ae='[ዐ]'; au='[ዑ]'; ai='[ዒ]'; aa='[ዓ]'; aie='[ዔ]'; e='[ዕ]'; ao='[ዖ]';
    replace_text= re.sub(ae,'አ',replace_text)
    replace_text= re.sub(au,'ኡ',replace_text)
    replace_text= re.sub(ai,'ኢ',replace_text)
    replace_text= re.sub(aa,'ኣ',replace_text)
    replace_text= re.sub(aie,'ኤ',replace_text)
    replace_text= re.sub(e,'እ',replace_text)
    replace_text= re.sub(ao,'ኦ',replace_text)
    replace_text= re.sub(aa1,'ኣ',replace_text)
replace the ጸ with ፀ b/c they have same sound
    tse='[ጸ]'; tsu='[ጹ]'; tsi='[ጺ]'; tsa='[ጻ]'; tsie='[ጼ]'; ts='[ጽ]'; tso='[ፀ]';
    replace_text= re.sub(tse,'ፀ',replace_text)
    replace_text= re.sub(tsu,'ፁ',replace_text)
    replace_text= re.sub(tsi,'ፂ',replace_text)
    replace_text= re.sub(tsa,'ፃ',replace_text)
    replace_text= re.sub(tsie,'ፄ',replace_text)
    replace_text= re.sub(ts,'ፅ',replace_text)
    replace_text= re.sub(tso,'ፆ',replace_text)
    return replace_text
```

Appendix C.3 A Sample Code for Word2vec Feature Extraction

```
# # word2vec feature extraction
from gensim.test.utils import datapath
w2vamh= Word2Vec.load('word2vecmodel/w2vmode21.bin')
w2vamh.init_sims(replace=True)
def word_averaging(w2v, words):
    all_words, mean = set(), []
    for word in words:
        if isinstance(word, np.ndarray):
```

```
            mean.append(word)
        elif word in w2v.wv.vocab:
            #mean.append(w2v.wv.syn0norm[wv.wv.vocab[word].index])
            mean.append(w2v.wv.vectors[wv.wv.vocab[word].index])
            all_words.add(w2v.wv.vocab[word].index)
    if not mean:
        logging.warning("cannot compute similarity with no input %s", words)
        return np.zeros(wv.wv.vector_size,)
    mean = gensim.matutils.unitvec(np.array(mean).mean(axis=0)).astype(np.float32)
    return mean
def  word_averaging_list(wv, text_list):
    return np.vstack([word_averaging(w2v, word_list) for word_list in text_list ])
```

Appendix C.4 A Sample Code for Building and Evaluating Models

```
#  # SVM

from sklearn.svm import LinearSVC
svmmodel_w2v= LinearSVC(C=0.01, multi_class='ovr', max_iter=10000,
class_weight='balanced',penalty='l2' )
svmmodel_w2v=svmmodel_w2v.fit(X_train, y_train)
cvscoresvm =cross_val_score(svmmodel_w2v, X_train, y_train, cv=5)
cvscoresvm, cvscoresvm.mean()
y_predcvsvm = cross_val_predict(svmmodel_w2v, X_train, y_train, cv=5)
reportclf(y_train , y_predcvsvm) # classification report
normalcm(y_train,y_predcvsvm) # normalized confusion matrix

# # NB

from sklearn.naive_bayes import MultinomialNB
clfnb_w2v = MultinomialNB () # default parameter
clfnb_w2v.fit(X_w2v, y_train)
cvscorenb =cross_val_score(clfnb_w2v, X_train, y_train, cv=5)
cvscorenb, cvscorenb.mean()
y_predcvnb = cross_val_predict(clfnb_w2v, X_train, y_train, cv=5)
reportclf(y_train, y_predcvnb)
```

```
normalcm(y_train,y_predcvnb)

# # RF

from sklearn.ensemble import RandomForestClassifier
clfRf_w2v=RandomForestClassifier(n_estimators=100, class_weight='balanced')
rfmodelw2v=clfRf_w2v.fit(X_w2v,y)
cvscorerf =cross_val_score(clfRf_w2v, X_train, y_train, cv=5)
cvscorerf, cvscorerf.mean()
y_predcvrf = cross_val_predict(clfRf_w2v, X_train,y_train, cv=5)
reportclf(y_train, y_predcvrf)
normalcm(y_train,y_predcvrf)
```