Developing an Automated Machine Learning Based Sentiment Analysis for Afaan Oromoo

By: Sultan Jenbo Kuyu

A Thesis Submitted to the Department of Computer Science and Engineering
School of Electrical Engineering and Computing

Presented in Partial Fulfilment of the Requirement for the Degree of Masters in Computer Science and Engineering

Office of Graduate Studies
Adama Science and Technology University

Adama, Ethiopia
February 2021

Developing an Automated Machine Learning Based Sentiment Analysis for Afaan Oromoo

By: Sultan Jenbo Kuyu

Advisor: Dr. Teklu Urgessa

A Master's Thesis Submitted to the Department of Computer Science and Engineering School of Electrical Engineering and Computing

Presented in Partial Fulfilment of the Requirement for the Degree of Masters in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

Adama, Ethiopia

February 2021

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: **Sultan Jenbo Kuyu**

Signature: _____

This MSc thesis has been submitted for examination with my approval as a thesis advisor.
Name: **Dr. Teklu Urgessa**

Signature: _____

Submission Date: _____

Approval of Board of Examiners

We, the undersigned, members of the Board of Examiners of the final open defense by Sultan Jenbo have read and evaluated his thesis entitled "**Developing an Automated Machine Learning Based Sentiment Analysis for Afaan Oromoo**" 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

ACKNOWLEDGEMENT

First, I would like to say Alhamdulillah for everything that Allah has given me. My sincere gratitude to my advisor **Dr. Teklu Urgessa** for his friendly approach, readiness, and willingness to advise on the research, his critical comments, his commitment to guide, and his support greatly improved this thesis work.

Next, I would like to acknowledge ASTU for giving me the chance to study the second-degree and the remaining thesis committee. I also would like to thank **Dr. Mesfin Abebe** for his kindness in leading me to this title, during title selection, encouragement, and hard questions. In particular, I am grateful to **Dr.Tilahun Melak** for his generousness always and cooperation, and helped me during a presentation of the proposal with the help of **Dr. Teklu Urgessa.**

I also want to express my gratitude to my whole family they are always with me and supported me as much as they can without any hesitation and giving me all possible they can do.

I am also thankful for Mettu University for the support they have given me starting from paying me my full salary and some extra benefit during my study.

Finally, yet importantly, I would like to say thank my entire classmate for the love and cooperation you have for one another.

Dedication

I would like to dedicate my work to all my family, and specifically my elder brother **Rameto Jenbo** (May his soul rest in heaven).

Acronym and abbreviation

AI:	**Artificial Intelligence**
AMLSA:	Automated Machine Learning based Sentiment Analysis
ANN:	Artificial Neural Networks
BOW:	Bag-of-Words
CBOW:	Continuous Bag-of-Words
CNN:	Convolutional Neural Network
F1:	F1-score
FN:	False Negative
GPU:	Graphics Processing Unit
GUI:	Graphical user interface
LIGA:	Language Identification Graph Algorithm
LR:	Logistic Regression
LSTM:	Long Short-Term Memory
MNB:	Multinomial Naïve Bayes
NB:	Naïve Bayes
NLP:	Natural Language Processing
NLTK:	Natural Language Toolkit
OM:	Opinion Mining
RNN:	Recurrent Neural Network
SA:	Sentiment Analysis
SVC:	Support Vector Classifier
SVM:	Support Vector Machine
TF-IDF:	Term Frequency-Inverse Document Frequency
VSM:	Vector Space **M**achine

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURE

LIST OF EQUATIONS

Abstract

Social media users are rapidly increasing from time to time, and it is becoming a common source of information for many sectors. The social media content contains several attitudes and sentiments or feelings. However, those sentiments on social media have no structure and difficult to identify their polarities for decision-making. Therefore, we are motivated to design, develop, and implement automated Machine Learning based sentiment analysis (AMLSA) for Afaan Oromoo. The reason to select 'Afaan Oromoo' is that even though Afaan Oromoo has a large number of speakers it is an under-resourced language. For this study, 6670 statements collected from the Facebook public page, and manually labeled into three classes namely positive 2270, neutral 2210, and negative 2190. The statement contains words, phrases and sentences. The dataset was split into 80% for training and 20% for testing sets. The research followed an experimental approach to determine the best combination of the conventional machine learning and Neural Network algorithm and features extraction for models. The researcher applied six different techniques; four of the six techniques are conventional machine learning techniques. The rest two of them are neural network techniques. The four, Conventional machine learning techniques are Multinomial Naïve Bayes (MNB), Logistic Regression (LR), Support Vector Machine (SVM), and Random Forest (RF). The two neural network techniques experimented in this research are Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) techniques. To evaluate the performance of each technique, the researcher used several performance evaluation metrics such as Accuracy, F-score, Precession, and Recall. The feature extraction techniques used for conventional machine learning techniques are unigram with TF-IDF, Bigram with TF-IDF, and trigram with TF-IDF, and neural network techniques used word embedding of word2vec methods. We used the trigram with collaboration TF-IDF feature extraction, and accuracy for performance evaluation because they achieved the highest performance result. The Conventional Machine Learning achieved accuracy for, MNB 80.12%, for LR 82.52%, for RF 80.62%, and for SVM 84.62%. The Neural network techniques achieved accuracy for CNN 72.91%, for LSTM 85.01%. According to the classification performance result from the entire techniques applied, the LSTM technique achieved the highest accuracy and we used LSTM to deploy our models.

Keywords: Sentiment, Sentiment Analysis, Afaan Oromoo, Machine Learning, Neural network.

CHAPTER ONE

INTRODUCTION

1.1 Background of The Study

Sentiment analysis or opinion mining is a process of extracting information from user's opinions. Many people share their opinion on the website, blogs, microblogs, web forums, and social networks. It is the method used to extract intelligent information based on individuals' or organizations' opinions from raw data available. In the past, when an individual needed opinions, he/she ask people but, nowadays, it is no longer limited to asking people. [1]. It is estimated that 80% of the world data is unstructured and has no organized definition to be used immediately to be used per or need. Most of this data can be in the form of text from emails, chats, social media, surveys, journal articles, and documents. So this data is difficult to be used, time-consuming, the analyzing cost is too high [2].

Sentiment analysis is widely applied for analyzing reviews like movie reviews, consumer product reviews, and other applications on social media for the purpose of determining the attitude and sentiments of the writer/opinion holder with respect to some topics or the overall contextual polarity of the document. It determines the subjectivity whether the expression is subjective or objective and uses automated tools to detect the subjective expressions of opinion holders. In natural language, subjective expressions are the aspects of the language that are used to express opinions, feelings, evaluations, and sentiments. Subjective expression adds an additional level of granularity by further classifying the subjective texts into positive, negative and neutral polarities. In general, sentiment analysis is widely used in various fields to analyze people's sentiments on various application domains can be movies, products, politics, hotels, and other specific topics with their attributes [3].

Opinions are fundamental to almost all human activities and are central influencers of our behaviors. Our feelings and perceptions of the world, and the selections we make, are, to a considerable degree, depending upon how others perceive and evaluate the reality. Due to this reason, when we need to determine a decision we often look for the opinions of others. This is also true for organizations [4].

Sentiment can be considered as "what one feels regarding something", "individuals experience", "one's own impression the mental state involving beliefs and feelings and values and dispositions

toward something" or "an opinion". Sentiment analysis brings together various research areas such as data mining and text mining, text classification, natural language processing. It is becoming of greater importance to organizations as they integrate online commerce into their operations, socio-politically become a decision-making point as it helps to know people sentiment The concentration of research in sentiment analysis is on processing the opinions stated in the text in order to discover the opinionated information. Which is different from retrieval and mining of factual information. This is the objective of much of the existing research in natural language processing and text analysis [4].

Sentiment analysis can be used at the same time or in connection with another system such as information extraction systems, recommendation systems, and question answering systems. The recommendation system performance may not be improved by recommending items that receive a lot of negative feedback. Opinion mining systems may help Information extraction systems by eliminating certain types of information that existed in subjective sentences. Different types of questions such as sentiment-oriented and definitional questions may acquire different types of treatments in question answering systems, which is another field that extracted from opinion mining or sentiment analysis [4][5].

Sentiment can be classified based on training algorithms of machine learning, which is on a set of selected features for specific missions and tests on another set whether it is able to detect the right feature and give the right classifications. Sentiment polarities can be categorized into negative, positive, and neutral. [3].

The investigation of sentiment analysis has base on a different perspective, some of the most popular perspectives are Document level, sentence level, and feature aspect level. The Document-level perspectives determine the overall sentiment of an entire document at once. The sentence level perspective is related to subjectivity classification and the task at this level is limited to the sentences and their expressed opinions, the aspect level perspective is directly related to each aspect of words, which is more challenging than both document level and sentence level [6].

Sentiment can be analyzed based on different methods and algorithms such as *Rule-based* (manually crafted rule), *Automatic* (machine learning technique to learn from data) and *Hybrid* (to combine both rule base and Automatic) techniques [7]. For this research work, we will use the Automatic method which will be based on machine learning classification techniques. Under this classification technique, the sentiment analysis task is usually modeled as a classification problem

where a classifier is given input with a text and returns the corresponding category of the polarities; namely positive, negative or neutral.

In Ethiopia, different social media are available which include: Facebook, YouTube, and Twitter. Ethiopian social media users can use different languages like Amharic, Afaan Oromoo, Tigrigna, Somali and English to express their sentiments on topics posted on social media. Hence, the sentiments can contain multiple languages because of the availability of various languages [8]. It is very necessary to do research on Sentiment Analysis in Afaan Oromoo language because many people forward their sentiments continuously in AO. So, to solve these problems there is a need for automated AO sentiment analysis and summarization system for an organization. Nowadays sentiment analysis has become an accepted mechanism in judging people/consumer sentiments towards various economic, business and socio-political issues. Sentiment Analysis offers both the people and the organization to choose the right product and/or services and also to improve the quality of the product and/or the services.

The main goal of sentiment Analysis is to find out the attitude of a writer (owner of the sentiment) with respect to some topic or the overall contextual polarity of a document. The attitude may be personal judgment or evaluation, affective state that is to say the emotional state of the writer or blogger when writing or the intended emotional communication.

This chapter provides an introductory part of sentiment analysis problems. The focusing points of this chapter are on highlighting the several challenges researchers encounter in sentiment analysis for Afaan Oromoo and some of the recent methods to address them. Since, recently, some researchers have begun addressing sentiment analysis or opinion mining by using, extending and modifying, topic modeling techniques. The remaining part of this chapter has structured as follows: section 1.2 I describe the motivation for the sentiment analysis for Afaan Oromoo. Section 2.3 define the problem statement of sentiment analysis including research questions to be answered by this study. In section 1.4 the Objectives of this research study described, in section 1.5 scope and limitation of the work identified, in section 1.6 the significance of the described. Finally, section 1.7 summarizes the overall organization of study.

1.2 Motivation of The Study

Since the numbers of social media users are increasing drastically by using different languages such as English, Amharic, Afaan Oromoo and many other languages, it is obvious that the data on the internet is also increasing proportionally. This huge amount of data has no structure for further

usage, and need to be categorized for proper management. Sentiment analysis involves text analysis, process natural languages to determine and evaluate the subjective information from source data [4]. In real-world business companies, governments, political campaigns, politicians and other organizations post their product, service, agenda, strategy and other issues on social media platforms in order to get opinions and feedback from consumers, public and citizens. Social media users can also express their feeling, attitude on the posted topics through various natural languages. However, there are challenges to analyze the sentiments, which is written in non-English languages such as Afaan Oromoo on the posted topics. There were a few researches done Afaan Oromoo sentiment analysis on social media. Jamal Abate[8] used word lexicon method for unsupervised Afaan Oromoo sentiment Analysis, the problem of lexicon method is the learning capacity of this algorithm is very low as compared to other machine learning algorithms, and incorrect sentiment scoring of sentiment word (Asghar et al [9]). Megersa Oljira [10] used multinomial Naïve Bayes, LSTM and CNN as classification algorithm on Bigram data (positive and negative sentiment). The problem of Bigram data is, it missed the neutral data these does not answer the social media demand in full-fledged structures that contains all polarities (Positive, Neutral, and Negative) in Afaan Oromoo. The limited availability of previous work on sentiment analysis in Afaan Oromoo motivated us to design and develop a machine learning-based Afaan Oromoo sentiment analysis that helps to analyze sentiment texts written in Afaan Oromoo language on social media platforms regarding socio-political data.

1.3 Statement of the problem

The number of opinions on social media is increasing from time to time due to the increasing number of users. As a result, identifying opinion sources and monitoring them on the Web can be a difficult task. And also, in many cases, opinions are hidden in long forum posts, blogs, and microblogs. It is very difficult for a human reader to find relevant information, extract important sentences, read them, summarize them and organize them into usable forms or to decide something using such unstructured information.

It is estimated that 80% of the world data is unstructured and has no organized definition to be used immediately to be used per or need. Most of this data can be in the form of text from emails, chats, social media, outliers, journal articles, and documents. So this data is difficult to be used, time-consuming, the analyzing cost is too high [2]. It is worth mentioning that most of the literature has considered the orientation of the opinion as a binary variable: either positive or negative. There

should some work that includes the neutral class as well. In general, the sentiment analysis could be regarded at different scales rather than just binary. Most of the reviewed literature was depends on the data that was gathered for another purpose.

Having the sentiment analysis review report makes it possible to be more focused on the agenda of interest and seek others' opinions regarding them. If one wants to get an idea about popular or unpopular expressions of contemporary sentiment, creating a frequency list of different discovered expressions could be beneficial. One of the benefits of sentiment analysis is that it empowers individuals, organizations, political parties or government to track the people's opinions over time. How things change from time to time has critical value for individuals, organizations, companies, political parties or government.

To the best the researcher's knowledge, eventhough there were some research work for sentiment analysis on social media for Afaan Oromoo languages that overcome the existing problems. Due to the absence of Afaan Oromoo, sentiment analysis on social media huge amount of comments, feedbacks and opinions written in language has been not analyzed. In addition to this, socio-political activities, governments, and other organizations cannot know consumers, public and citizen's opinions and feedback on their product, service, agenda, and strategies when the sentiment texts are written in AO language. This the great problem for the users Afaan Oromoo due to absence of full filleged structured sentiment Analysis model. To solve these problems, the following research questions answered in the study.

- *What are the preprocessing procedures and tools applied to come up with quality data sets for training and testing the data?*
- *What are the best features extracted from opinionated Afaan Oromoo texts that have associated with socio-political data?*
- *Which machine learning approach produces a better classification model for Sentiment Analysis for Afaan Oromoo text?*

1.4 Objectives of The Study

1.4.1. General Objective

The main objective of this research is to design and develop an automated machine learning-based Sentiment Analysis for Afaan Oromoo.

1.4.2. Specific Objective

In order to achieve the general objective, the following specific objectives have been accomplished.

- Build the datasets using Afaan Oromoo posts and comments from the Facebook specified public page.
- Perform and identify reprocessing procedures and tools applied to come up with quality data sets for Afaan Oromoo
- Develop annotation guidelines for labeling posts and comments.
- Design the general architecture of the Afaan Oromoo sentiment analysis system.
- Identify effective features extraction and machine learning algorithms for Afaan Oromoo sentiment analysis.
- Test and evaluate the model.

1.5 Scope and limitation of The Study

1.5.1 Scope of The Study

This study concentrated on the design, develop and implement Sentiment analysis for Afaan Oromoo language. We selected this language for the reason that despite the fact that having large number of speakers but it is less resourced language or less number of research were conducted by using this language. The study collected post and comments from social media particularly form the Facebook page of popular organizations in Ethiopia, and only uses Afaan Oromoo language for this work those Facebook pages are FBCAfaanOromoo, BBCAfaanOromoo, OBNAfaan Oromoo. The study provide annotation or labeling guideline to label the collected dataset. The study has used several Conventional machine learning algorithm like NB, RF, SVM, LR, and two other neural network algorithms namely LSTM and CNN. For feature extraction the research used TF-IDF in combination with n-gram, and word2vect techniques.

The research work focuses on the Polarity checking of sentiment analysis for Afaan Oromoo language. The polarity can be Positive, Negative, and neutral.

1.5.2 Limitation of The Study

Since there is no sufficient research that has been conducted on Afaan Oromoo language it is difficult to get data set, and also time constraints are grate challenge. We collected and labeled or

annotated the dataset for ourselves, so the following are the limitation of the research work, which will be included in our future work of the extension of this study.

- Detecting spelling error and identifying emojis are not included in in this study.
- This work does not include identification of the subjectivity of sentiments and unable to identify sarcastic sentiments.
- This work also not intended to include identification of the holder of the sentiment.

1.6 Application of The Study

1.6.1 General Benefit

Sentiment analysis helps the enterprise to estimate the public opinion, conduct market research, conduct product report and understand customer experiences.

1.6.2 Socio-political

Now in the current era, the sentiments of the people are easily reflected on social media, among those sentiments political issue is hot and repeatedly discussed. People always give comments regarding socio-political issues, then the government shall refer the people sentiment for media which helps the governments to know the direction of what people want, and what people do not want.

1.6.3 Individual needs

The outcome of this work can be used by any internet users at any time on social media or related web applications. The model is designed to generate structured reviews that can be easily used by the individual for decision making based on public opinion.

1.7 Organization of The Thesis

This study contains seven chapters, which has been organized as follows:

The first chapter gives the overall introductory information of the study. Under this chapter, we try clear up what are the researchable problems found, the motivation of the study, the research hypothesis used to answer relevance of the problem it is addressing the research problem, the purpose of general and specific objective to solve the research problems, and the general research methodology carried out to accomplish research objectives are. The scope and limitation briefed, and finally the significance of the study has been state on this chapter.

The second chapter describes literature review. This chapter describes the literature part and source of the research area, related work of the study structure of the Afaan Oromoo languages and to learn from others' work, and to evidence, the hypothesis of the research, different kinds of literature on the problem domain are reviewed. Additionally, this chapter identifies the gap of related paper.

The third chapter describe the research methodology used in this research work, methods used to build the dataset, methods used to develop Afaan Oromoo sentiment analysis models which are text preprocessing, feature extraction methods, and machine learning classifier algorithm's neural network classifiers, and finally, it presents the evaluation metrics.

The fourth chapter describes the proposed solution for Afaan Oromoo sentiment analysis, which describes the overall architecture of the system, the proposed text preprocessing, feature extraction, machine learning training, and evaluating.

The fifth chapter describe the implementation and experimentation of the proposed solution. It contain working environment, description of dataset, also the implementation of preprocessing, feature extraction implementation, models implementations, and evaluation of the models.

The sixth chapter explains the discussion and result of the proposed solution compared with previous (chapter five) result findings absorbed from this research. The effectiveness, of the proposed solution (model) of this conducted research is also explained in this discussion and result chapter.

Chapter Seven: under this chapter we concludes the research and give something useful or necessary an insight into the recommendations and valuable suggestions. The impacts of this research are also described in this final chapter.

CHAPTER TWO

LITERATURE REVIEW AND RELATED WORK

2.1 Introduction

This chapter reviews relevant literature to further comprehend the concept and investigate the research problem. Which starts with defining and describing sentiment analysis, and provides a technical review on Sentiment analysis and existing techniques. In addition, the study followed by a review of Sentiment Analysis on social media, Sentiment Analysis in Ethiopia, and techniques or classifiers that have been used in Sentiment Analysis, characteristics of Afaan Oromoo language, and finally explains the related work previously studied on the problem.

2.2 Sentiment Analysis

The definitions of Sentiment Analysis form Different sources described below, these sources are website platforms, and scientific communities. The following are some of the list of definitions and sources of definitions:

Definition 1 English Wiktionary: "Sentiment Analysis is the use of computers to process natural language in an intention to identify and extract information about the writer's affective state." This system also defined the sentiment analysis as the analysis of text to identify subjective information about the writer [11]

Definition 2 Wikipedia: Sentiment analysis also known as opinion mining or emotion Artificial Intelligence: defined as "the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine"[12].

Definition 3 Jurafsky, Dan Stanford: Sentiment analysis is the detection of attitudes enduring, affectively beliefs, tendencies along wise of objects or persons: source of attitude, aspect of attitude, and type of attitude type of attitude most of the time referred as polarities of attitude which is positive polarity, neutral polarity and negative polarity [13].

Definition 4 Alsaeedi, Abdullah: "Sentiment analysis is also known as 'opinion mining' or 'emotion Artificial Intelligence' and alludes to the utilization of natural language processing

(NLP), text mining, computational linguistics, and bio measurements to methodically recognize, extricate, evaluate, and examine emotional states and subjective information" [6].

The majority of the sources we have referred and other entities have similarities between their definitions of sentiment Analysis. From the above definitions we have summarized as follows: Sentiment analysis or opinion mining is part of Natural Language processing which is about analysis of opinions or emotions from text data. Sentiment analysis identifies specific events from generated opinion or sentiment of each person polarity towards a particular event is either positive, negative or neutral. Sentiment analysis helps to understand the emotional state of the author when writing for sentiment analysis document or text has to be given to the model then can be analyzed and generates the polarity for each event which represents a summarized form of the opinion of the given document. Sentiment analysis is mostly used for review of movies, political activities, social opinion, products, customer services, opinions about any event, etc. This helps the user of sentiment analysis to decide whether a specific item or service is bad or good or favored or not favored. It is also applied to distinguish feelings of people about any event or persons and also finds polarity of text whether positive, negative or neutral[11],[12],[13][14]. Natural Language Processing (NLP) is one of the fields of, computational linguistics, artificial intelligence and computer science concerned with the interaction between computer and human languages, specifically related to programming computers to productively process huge natural language corpora [16]. It is a manner for computers to recognize, analyze, and deduce meaning from human languages in a smart and useful way. Using NLP, developers can coordinate and structure knowledge to accomplish tasks such as sentiment analysis, machine translation, named entity recognition automatic text summarization, speech recognition, and automated question answering.

2.3 Sentiment analysis on social media

Social media is one of the most substantial means of information exchanging technologies this century. Social media exploiters use social media platforms to post messages, images, videos, and others about their everyday activities. Several channels of social media such as Twitter, Facebook, YouTube, and LinkedIn allow for convenient and effective means of communication and sharing of information publically. Because of this reason, the exploiters of internet drastically increase from time to time and the high accessibility of user texts on the Internet, extraction of useful information automatically from great quantity of documents and unstructured texts becoming the

concern for a lot of researchers in many fields, in particular in NLP community [12][15][17][15][19].

The need for Sentiment Analysis is increasing as the rise of social networks and blogs. With the increment of reviews, ratings, recommendations and other forms of online expression, online opinion has becoming medium of virtual currency for businesses looking to market their products, discover new opportunities and manage their reputations. As socio-political and business affairs look to automate the process of filtering out the noise or interference, understanding the conversations, identifying the relevant content and achieving it appropriately, many are now looking to the area of sentiment analysis. If social media on the web was all about democratizing publishing, then the next level of the web may well be based on democratizing data mining of all the content that is getting published [20].

2.4 Sentiment analysis Techniques

There are two technics of sentiment classification

i. Machine learning technique and
ii. Lexicon method

Classification is used to assign a sentiment sentence into a specific class also concerned as a class or categories depending on their sentiment polarities. Sentiment analysis (SA) system can classify sentiments in the sentence by following different techniques or approaches. Under this section, we tried to discuss the sentiment analysis and classification techniques.

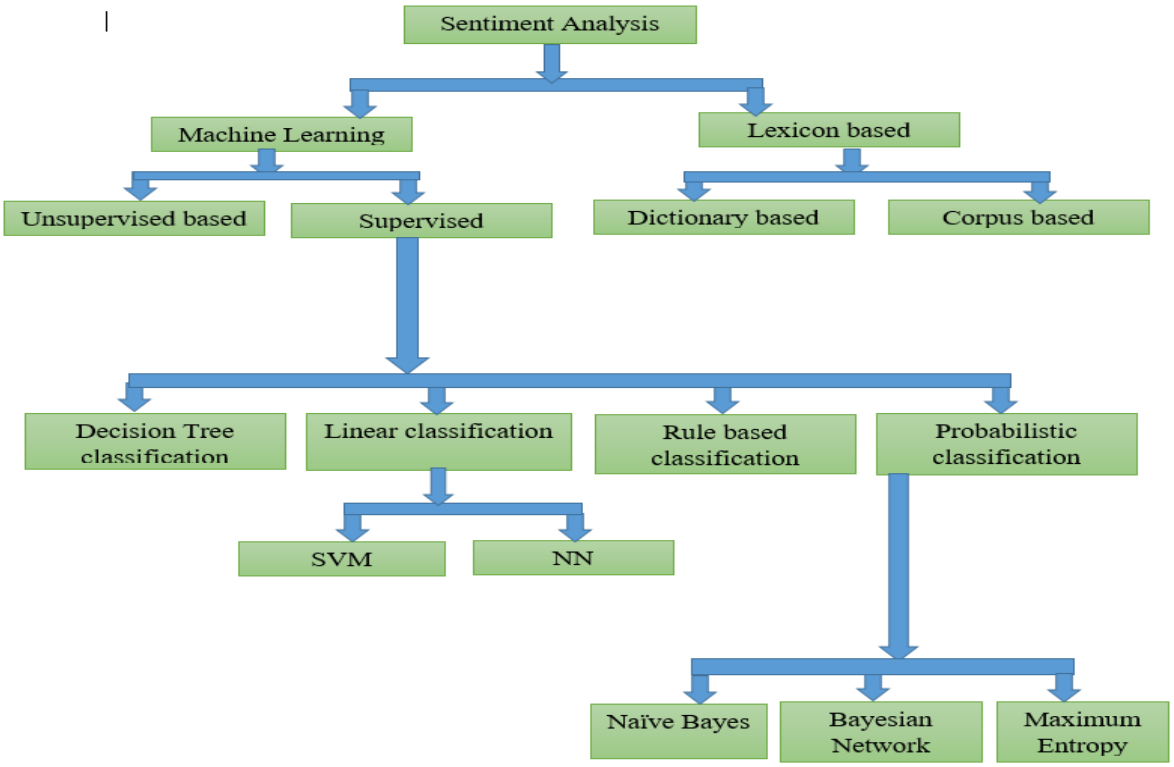

Figure 2. 1 Techniques of sentiment analysis[21]

2.4.1 Lexicon Based Approach

The Lexicon based technique employees calculating the sentiment polarity of a sentence from the semantic orientation of words or phrases in the sentence and use the sentiment of lexicons, collection of known and precompiled sentiment terms with their polarity values [22]. The system classifies the sentiments into specific categories of the sentiment classes which are positive, negative or neutral based on the positive or negative sentiment terms that occurred in the sentence. This can be done using rule-based classifier methods, and if there are positive words only or more numbers of positive words than negative words in the sentence, the semantic orientation of the sentence is classified into positive. If there are negative words only or more negative words than positive words in the sentence, the semantic orientation is classified into negative. If there are equal numbers of positive and negative sentiment terms or there are no sentiments terms in the sentence, the semantic orientation of the sentence will be neutral and categorized in neutral class. In this approach, overstatement and understatement words are considered (i.e., overstatement words increase the semantic orientations and understatement decrease the semantic orientations of the sentiment words in the sentence).

Lexicon approach is one of the two main approaches to sentiment analysis and it employees calculating the sentiment from the semantic orientation of word or phrases that found in written text. This approach uses a dictionary of positive and negative words, with a positive or negative sentiment value that can be given to each of the words. Different approaches to creating dictionaries have been proposed, including manual and automatic approaches. Generally, in lexicon approaches a separate part of text messages is represented as a bag of words. By adopting this representation of the text, sentiment values from the dictionary are allotted to all positive and negative words or phrases within the opinion or written text. The aggregating function, such as sum or average, is used in order to make the targeted prediction regarding the overall sentiment for the message. Apart from a predicted sentiment value, the expression of the local context of a word is most of the time taken into consideration, such as negation [23][24][25].

2.4.2 Machine Learning Approach

Machine learning (ML) is the sub-field of computer science that gives a computer to learn without being explicitly programmed. Machine learning Techniques carries out sentiment classifications based on training algorithms, the classification can be performed according to the set of selected characteristics for specific mission and test on other sets whether it is able to detect the right characteristics and give the right classifications. For sentiment classification, we can use a machine learning algorithm both supervised and unsupervised sentiment classification models. The supervised approach of sentiment classification model uses a large number of labeled training datasets, while the unsupervised approach sentiment classification model uses when there are labeled training datasets. The supervised approach sentiment classification model can classify or categorize the sentiment sentence based on algorithms like SVM classifiers, Maximum Entropy and Naïve Bayes [31].

Naïve Bayes (NB) classifier uses Bayes rule as its main equation, this uses the assumption of conditional independence: each individual character is considered to be an indication of the assigned class, independent of each other. An NB classifier makes a model by adapting a distribution of the number of occurrences of each feature for all the documents. Most of the time this classifier selected because of its simple for implementation, its computational efficiency and straightforward incremental learning [26].

Multinomial Naïve Bayes is a simple probabilistic model based on Bayes theorem along with a strong independence assumption. It computes the posterior probability of class based on the distribution of words in the document. It uses Bayes‟ theorem to predict the probability that a given feature set belongs to a particular label.

$$P\left(\frac{feature}{lebel}\right) = \frac{P\left(\frac{lebel}{feature}\right) * P(feature)}{P(lebel)} \qquad \textit{Equation 2. 1}$$

Where:

- **P(feature/lebel):** is posterior probability which randaomly observed data conditional probability
- **P(lebel/feature):** is likelihood probability
- **P(features)** is prior probability which is about possibility of outcome of the event.
- P(lebel): is evidence

Maximum Entropy classifier is one of the probabilistic classifiers which becomes categorized as the class of exponential models. The models that correspond to the training examples are computed, where each of these characteristics corresponds to a constraint on the model. The model with the maximum entropy overall models that satisfy these constraints is selected for classification. Maximizing the entropy assures that no biases in the classification system. The Maximum Entropy (ME) classification approach ensures that the probability that a document belongs to a specific class in the given context must maximize the entropy of the classification system. By maximizing entropy, it is assured that no biases are introduced into the system. The maximum Entropy can be used when we don't know anything about the prior distributions when it is unsafe to make any such assumptions, and moreover when we can't assume the conditional independence of the characteristics [27].

A Support Vector Machine (SVM): can be operated by constructing a hyperplane with maximal Euclidean distance to the closest-fitting training examples. This can be considered as the distance between the classifying hyperplane and the other two parallel hyperplanes at each slope, representing the edge of the examples of one class in the characteristics space. It is accepted that the best generalization of the classifier can be found when this distance is maximal or uttermost.

If the data is not divided or not separated, a hyperplane will be taken to splits the data with the least error possible. An SVM is known to be strong enough to withstand or overcome challenges in the event of many (possibly noisy) characteristics without being doomed by the curse of dimensionality and has afforded high accuracies in sentiment classification [26]**.**

2.4.3. Hybrid approach of machine learning and lexicon

The combination of both lexicon-based and machine learning-based approaches benefits from their synergy effect that rises to the hybrid approach. The paper [28] have proved that this combination gives improved performance in sentiment classification. In the hybrid approach, sentiment lexicons are used as seed resources and to detect sentiment polarities and the results from the lexicon-based method are used to train machine learning algorithms. Then, sentiment sentences are analyzed using machine learning classifiers based on the knowledge acquired from the training and the lexicon resources [28].

2.4.4. Neural Network for sentiment analysis

The analysis techniques mentioned so far are the traditional approach of machine learning and the lexicon approach, under this topic we tried to mention some of the deep learning approaches which are the latest machine learning techniques.

Deep learning is one of the applications of artificial neural networks (ANN) to learning tasks by using multiple layers of networks. It can exploit much more learning (representation) power of neural networks, which once were viewed as to be applicable only with one or two layers and a small amount of data. The following image is taken from [29]paper, which describes the Feedforward neural network between neurons.

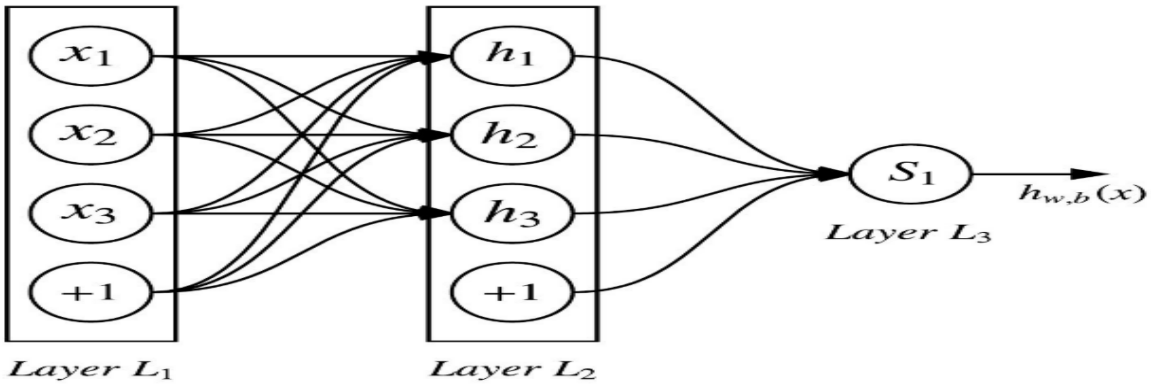

Figure 2. 2 Feedforward neural network

ANN inspired by the structure of the biological brain, neural networks consist of a large number of information processing units (called neurons) organized in layers, which functioned in unison of agreement. It can learn to execute tasks like text classification or sentiment analysis by adjusting the connection weights between neurons, corresponding to the learning process of a biological brain [29][30][31]. Some of the neural Network algorithm this study can use are CNN and LSTM.

Convolutional Neural Network (CNN): is a convolutional layer to extract information by a larger piece of text, so we work for sentiment analysis with the convolutional neural network, and we design a simple convolutional neural network model and test it on benchmark [31]. Convolutional neural networks are comparative to feed-forward neural systems. Where the neurons are capable of learning better point (weights) and predispositions. In CNN, the statement model of **e**ach word in the input data is associated to a vector representation, which consists in a dimensional vector.

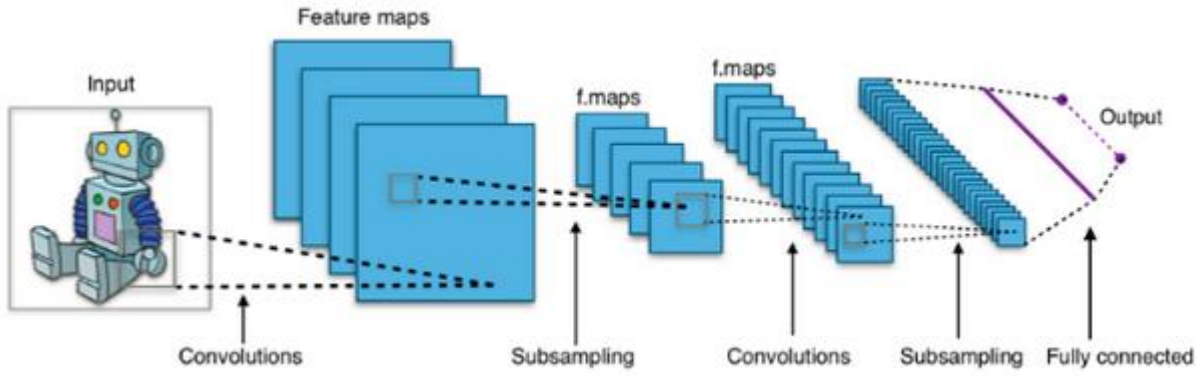

Figure 2. 3 Convolutional Neural Network architecture [32].

CNN classification technique is one of the artificial neural networks; which has strong adaptability and good at mining data local features. The way of its sharing network structure make it more similar to the biological neural networks, reduce the complexity of the network model, a reduction in the number of weights, makes the CNN be applied in various fields of pattern recognition, and achieved very good results[33][32]. CNN convolutes and abstracts word vectors of the original text with filters of a certain length, and thus previous pure word vector become convolved abstract sequences.

Long Short-Term Memory (LSTM): is one of the classification techniques of artificial recurrent neural network (RNN) architecture widely used in the area of deep learning. A Recurrent Neural Network is one of artificial neural network in which the output of a unique layer is preserved and fed back to the input. This assist to predict the result of the layer. The first layer made in the same way as it is in the feedforward network. This can be done in the way that the product of the sum

of the weights and features. However, in later layers, the recurrent neural network process starts[32].

Each node will remember some information from each time-step to the next node that it had in the previous time-step. In other expression, each node acts as a memory cell while computing and performing the operations. The front propagation of the neural network begins with as usual but it remembers the information it may need to use later. The system self-learns and works towards making the right prediction, even if the prediction is wrong during the backpropagation. This type of neural network is very effective in text-to-speech conversion technology. [32][34].

In LSTM architecture[34], designs a memory cell which maintains its state for a long period of time and non-linear providing units regulating information flow into and out of the cell. By applying this memory cell, LSTM has the ability to catch efficiently long distance dependencies of sequential data without brooking the exploding or vanishing gradient problem of recurrent neural network

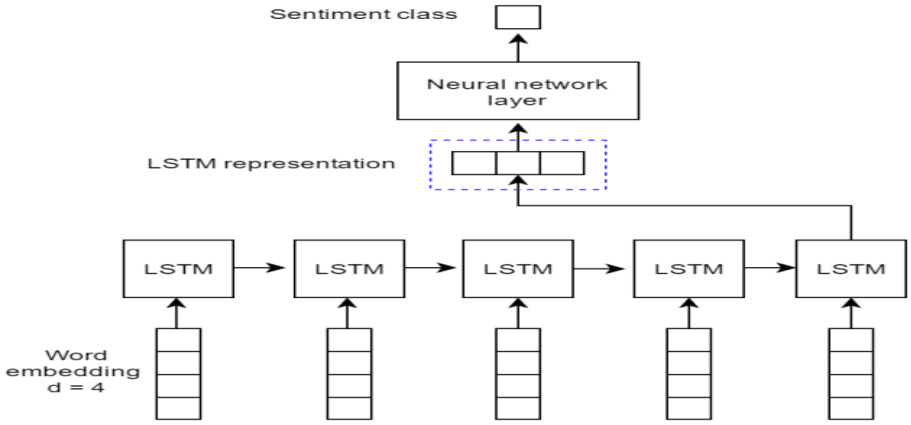

Figure 2. 4 LSTM architecture[34]

2.5 Feature extraction Techniques

Feature extraction is repeatedly used solution to overcome the curse of dimensionality in text classification problems in NLP. Feature selection held in the way that extracts text information is an extraction to represent a text message; it is the basis of a large number of text processing. This subtopic describe the feature extraction used in text classification specifically sentiment analysis[35]. The majority of the research papers follow the strategies already known in text classification to the specific problem of sentiment analysis [33], [35]–[40], but there are some of research papers use specific sentiment analysis features used to undertake the problem.

Bag-of-words (BOW): this model is a common method repeatedly used for text representation, it is a mode to extract features from the text to be use in algorithms for machine learning algorithms. The approach is simple and flexible, and can be used in a multiple of ways for extracting features from dataset.

A bag-of-words is a representation of text that depicts the occurrence of words within a dataset. BOW involves two things: A vocabulary of known terms and a measure of the presence of known terms within dataset. It is known as a "*bag*" of words, for the reason that any information about the order or structure of words in the Dataset is discarded. The model concerned only whether known words found in the document, not where in the document[38][37].

N-grams: is a set of simultaneous occurring words in the text. N gram is used for formulating features for supervised machine learning model, it uses probabilistic methods to predict the next word after observing the previous words in a text [41]. N-gram appropriates for improving classifiers' performance than BOW because it integrates, the context of each word to some extent [26][42] [43].

Term Frequency-Inverse Document Frequency TF-IDF: is one of the most commonly used weighting approach for assessing the relationship of words and documents. It has been applied to word feature extraction for text classification or other NLP tasks. The words which has higher record TF-IDF weights are marked as more representative and are kept while the ones with lower weights are less representative and are discarded. An appropriate selection of word features is able to increase the speed the information retrieval process while preserving the model performance. However, for many works, the cutoff values or the thresholds of TF-IDF for selecting relevant words is only depend on guess or experience [44].

Word embedding: is a representation of text where words that have similar meaning have the same representation in the document. In other expression it represents words in a organize system where related words, based on a dataset of relationships, are putted closer to each other. TensorFlow, Keras frameworks of the deep learning part is usually managed by an embedding layer which stores a lookup table to map the words acted by numeric indexes to their dense vector representations [45][46].

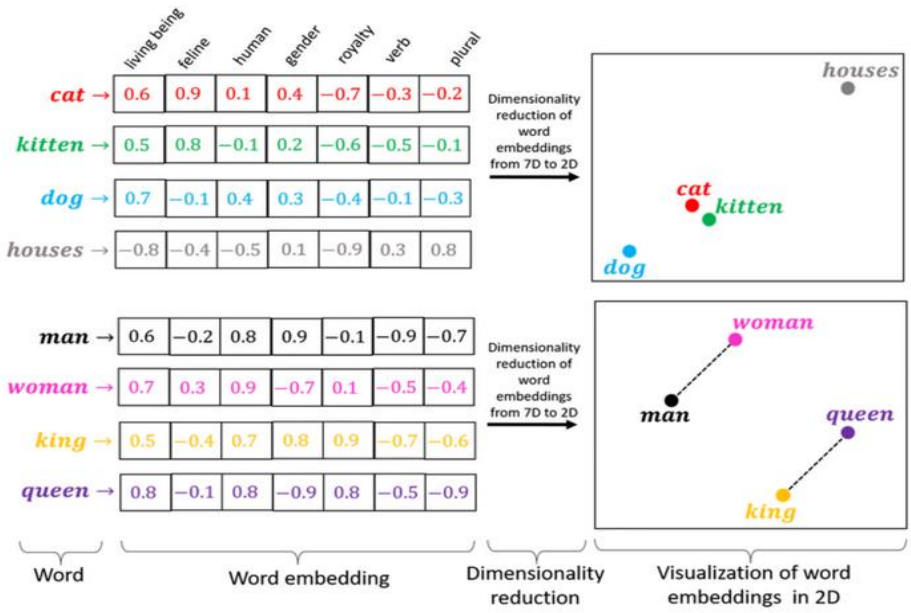

Figure 2.5 Word embedding architecture[46]

We need word embedding for the following porpuse.

- Computer only understands numbers.
- Word Embedding are the texts converted into numbers
- A ve*ctor* representation of a word may be a one-hot encoded vector like [0, 0, 0, 1, 0, 0].
- It is able of catching context of a word in a dataset, syntactic and semantic similarity, relation with other words

Word2vec is one of the word embedding feature extraction technics. Word2vec is a technique that was introduced by Mikolov[47], it uses a shallow neural network to train word embedding. It is widely used in training word embedding from raw text. The concept of word2vec (word embedding) developed from the concept of distributed representation of words, it uses a shallow neural network to train word embedding and predicts between every word and its context words to words occurring in similar contexts are related.

2.6 Related Work

To meet the objective of the research, a review different related work is a very important concept to be concerned. Many researchers were conducted research that related with sentiment analysis and text classification on the different languages; in Ethiopia, few researchers conducted their research on this area that closely related to the study.

Jamal Abate[8] used word lexicon method for unsupervised Afaan Oromoo sentiment Analysis, he conducted experiment three time, in his first experiment he was conducted by considering unigram words as a feature. For his unigram feature extraction, 73% of the feature extracted, and for the 79% of the extracted unigram features correctly extracted. On his second experiment of bigram feature extraction and classification, his experiment used most informative words than the first unigram experiment. On this stage, the system performed 70% for extracting features and from the extracted bigram feature 82% of them correctly extracted. On his third experiment, he applied trigram feature extraction with most informatory words than that of the unigram and bigram feature extraction had the performance of 66% to extract the usable trigram features and 69% performance to right extract.

Megersa Oljira [10] used multinomial Naïve Bayes, LSTM and CNN as classification algorithm and used TF-IDF and n-gram plus word2vec feature extraction techniques for his Sentiment Analysis of Afaan Oromoo Social media content using machine learning approaches. He used Facebook page for data sources and used binary dataset. His experiment result shows 93% for Multinomial Naïve Bayes, 89% CNN, and 87% for LSTM.

Mabrahtu Tedese did on Trilingual sentiment Analysis on social media(Amharic, Tigregna, and English, by using word lexicon unsupervised method. For the performance of the system evaluation he used precision, recall and F-measure, and he achieved an average precision 87.49%, recall 84.78% and F-measure 85.99% [48].

Sentiment analysis can provide valuable deep perception from social media platforms by detecting emotions or opinions from a huge volume of the unstructured format of data [49]. The [49] uses a variety of sentiment analyzers such as TextBlob, SentiWordNet, and WSD was to give a comparison of their accuracies. Among the three sentiment analyzers the researcher compared the research, they found that TextBlob had the highest rate of tweets with positive sentiment, 3380 in number and 54.08% in percentage. SentiWordNet gave 3054 the highest negative sentiment rate, 48.86%, and SentiWordNet for positive 2593 (41.488%) for neutral 603 (9.648%) and for negative 3054 (48.864%) however the researcher mainly concentrate on positive and negative sentiments which are not fair it is better to use equivalent number of all neutral, negative and positive sentiments.

The paper of A. Gupta, J. Pruthi, and N. Sahu [50] tried to use various machine learning approaches in a hybrid manner to give better results as compared to approaches used in isolation or used

separately. Moreover, as the tweets are very raw in nature, the researcher tries to come up with the use of various preprocessing steps so that to get useful data for input in machine learning classifiers. They were done on two machine learning algorithms K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) in a hybrid manner. The analytical observation is obtained in terms of classification accuracy and F-measure for each sentiment class and their average. The evaluation analysis shows that the accuracy of KNN is 67.78 and SVM also 67.78 and the Hybrid is 76.17 this indicates that the hybrid approach is better both in terms of accuracy and F-measure as compared to individual classifiers. This will be work for few data set and may not always true for a large data set and for the language that is not English language data set.

Wondwossen Mulugeta [51] studied the Multi-scale sentiment analysis model for Amharic online posts written in Ethiopic scripts by using a supervised machine learning approach. The main objective of the work is to identify multi-scale sentiment sentences based on the polarity weight value of sentiment words. The author has prepared a corpus from posts of social media, the researcher collected 608 posts. The author has done preprocessing the data he collected before the actual classification of the sentiment polarity. After preprocessing of data, he annotated the corpus manually by giving polarity values and sentiment intensity scale values. The author adopted two-scale schemes which are scale positive sentiment further as +1 and +2 for less and more positive respectively and negative sentiment polarities as -1 and -2 for less and more negative respectively, hence neutral sentences were annotated as 0. To distinct the polarity strength of Amharic sentiment words, the author limited to five polarity rank scales. Naïve Bayes algorithm was used to classify sentiment texts based on the trained on the entire corpus. The algorithm used n-gram features to acquire the knowledge from the training corpus, and then the algorithm classifies the sentiments of the test posts based on the acquired knowledge. Among the sample corpus, 486 posts were selected for the training dataset and 122 posts were for testing dataset. The author achieved an accuracy of 43.6%, 44.3% and 39.3% for unigram, bigram, and hybrid language models respectively. However, the accuracy of the system was very low since the training dataset was very insufficient and the tool used for morphological analysis was not effective.

The paper [20] presents classification results for the analysis of sentiment in political news articles. The author conducts a two-step classification model, distinguishing first subjective and second positive and negative sentiment texts. More specifically, they propose a shallow machine learning

approach where only minimal features are needed to train the classifier. The main goal of the paper is to experiment with simple feature combinations in a two-step binary classification process.

Table 2. 1Summary of related work

Authors	Title	Technique/Algorithm used	Result	Drawback
Jamal Abate	Unsupervised Opinion Mining Approach for Afaan Oromoo Sentiments	word lexicons	The evaluation of performance to extract bigram feature was 70% recall, 82% precision and 76% F-measure.	The classification technique he used is unsupervised, it is difficult for large dataset by using this technique, he also use 600 reviews only
Megersa Oljira	Sentiment Analysis of Afaan Oromoo Social media content using machine learning approaches	Multinomial Naïve Bayes, LSTM, and CNN	Experiment result shows 93% for Multinomial Naïve Bayes, 89% CNN, and 87% for LSTM.	He used binary data that is positive and negative datasets, this lead to lost or misclassify Neutral polarity of sentiments
Mebrahtu Tadesse G/Medhin	Trilingual sentiment Analysis on social media.	Used word lexicons for multiple language	He came up with the following performance evaluation methods: an average precision 87.49%, recall 84.78% and F-measure 85.99% were obtained	He used few data set for three difffernt language and also the technique he used is not compatible for large dataset.
A. Gupta, J. Pruthi, and N. Sahu,	Sentiment Analysis of Tweets using Machine Learning Approach	K-Nearest Neighbours (KNN) and Support Vector Machines (SVM) in a hybrid	The accuracy of KNN is 67.78 and SVM also 67.78 and the Hybrid is 76.17 this indicates that the hybrid approach is better.	There were only a few data set and may not always true for a large data set and for the language that is not English language data set.

A. Hasan, S. Moin, A. Karim, and S. Shamshirband,	Machine Learning-Based Sentiment Analysis for Twitter Accounts	uses a variety of sentiment analyzers such as TextBlob, SentiWordNet, and WSD	TextBlob 54.08% for positive, SentiWordNet 48.86% for negative, SentiWordNet (41.488%) for positive, and (9.648%) for neutral.	The researcher mainly concentrates on positive and negative sentiments which are not fair it is better to use an equivalent number of all neutral, negative and positive sentiments.
Wondwossen Mulugeta,	A Machine Learning Approach to Multi-Scale Sentiment Analysis of Amharic Online Posts,	Naïve Bayes algorithm was used to classify sentiment texts based on the trained on the entire corpus.	The authors used sample corpus of 486 posts and achieve the result of the accuracy of 43.6%, 44.3% and 39.3% for unigram, bigram, and hybrid language models respectively	The accuracy of the system was very low since the training dataset was very insufficient and the tool used for morphological analysis was not effective.
Bakken, Patrik F. Bratlie, Terje A. Marco, Cristina Gulla, Jon Atle	Political news sentiment analysis for under-resourced languages	the shallow machine learning approach	By performing 10-fold cross-validation, we obtain close to state-of-art results in this task, over 70%	They do not investigate further on how to deal with negation in real-time systems, again they didn't consider neutral sentiment

2.7 Afaan Oromoo Language

Oromia is the largest Regional States in the current Ethiopian Federal Government that is mainly inhabited by Oromoo People. Oromoo People speak Afaan Oromoo which is their own native language. Afaan Oromoo literal meaning Oromoo Language, which is the most widely spoken of the Cushitic family of is an Afro-Asiatic language. It is one of the major indigenous African languages that is widely spoken and used in most parts of Ethiopia and some parts of the

neighboring countries. According to [52] Afaan Oromoo is spoken by 40 percent of the Ethiopian population in Ethiopia, Afaan Oromoo is spoken as a lingua franca by other people who are in contact with Oromoo people. Afaan Oromoo is not only spoken by Oromoo people but also used by different nationalities such as Harari, Sidama, Anuak, Gurage, Amhara, Koma, Kulo, and Kaficho which is mainly as a means of communication and trade with their neighboring Oromoo people. In connection to this, [52] states that "It [Afaan Oromoo] is also used as a language of inter-group communication in several parts of Ethiopia." In addition to this, Afaan Oromoo can be spoken outside Ethiopia, some of them are in Kenya, Somalia, Sudan, and Tanzania. These make Afaan Oromoo one of the most widely spoken languages in Africa.

Afaan Oromoo is the official working language of the Oromia region and is used as a learning medium in elementary school and high schools. and also taught at degree, masters, and Ph.D. level at some universities of Ethiopia, in addition to this, currently Afaan Oromoo is becoming a federal working language in Ethiopia an addition to four other languages such as Amharic, Afar, Somali, and Tigrigna. It is spoken in a vast territory of Ethiopia ranging from Tigray in the North to Central Kenya in the South, and from Wollagga in the West to Harar in the East. In these areas, it is spoken with several dialects [52]. Afaan Oromoo shows variations based on the geographical areas where it is spoken. Different scholars tried to classify the dialects of Afaan Oromoo based on the geographical background of the speakers. However, there have been differences and inconsistencies among scholars and researchers regarding the number and categorizations of the dialects as the following examples illustrate.

2.7.1 Writing System of Afaan Oromoo

Afaan Oromoo's writing script known as Qubee has been officially decided, which is based on the Latin orthography [53]. Qubee has 33 which is consists of 26 single characters and 7 compounds, those letters can be separated as five vowels (a, e, i, o,u) and the rest are considered as a consonant, each character representing distinct sounds and it has both capital and small letters. There is a considerable amount of glottal stops in Afaan Oromoo. An apostrophe and less commonly a hyphen (‘) used to represent this sound in place of consonants in writing. In this writing system H, which acts as the closest glottal sound, is also used in place of an apostrophe. The apostrophe will be considered as a distinct symbol (can be, as the 27th letter of the alphabet Afaan Oromoo writhing system). In Afaan Oromoo writing system, Geminated consonants and long vowels are

represented by double letters. In addition to seven compound symbols not all the 26 letters correspond with their English sound representation as shown in table below.

Table 2. 2 Qubee and IPA (International Phonetic Alphabet (IPA)

Qubee(single characters)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
IPA	/a/	/b/	/č'/	/d/	/e/	/f/	/g/	/h/	/i/	/ğ/	/k/	/l/	/m/	/n/	/o/
Qubee(single character)	P	Q	R	S	T	U	V	W	X	Y	Z				
	p	q	r	s	t	u	v	w	x	y	z				
IPA	/p/	/k'/	/r/	/s/	/t/	/u/	/v/	/w/	/t'/	/y/	/z/				
Qubee(compound character)	CH	DH	NY	PH	SH	TS	ZH								
	ch	dh	ny	ph	sh	ts	zh								
IPA	/č/	/đ/	/ň/	/p'/	/š/	/s'/	/ž/								

When we come to the writing system for Afaan Oromoo, the alphabet "Qubee" (Latin-based alphabet) has been adopted and became the official script of Afaan Oromoo since 1991 (Gamta 1992). This writing system was adapted largely from the fact that its characters do explicitly represent the vowels and the consonants of a language (Gamta 1992). The "Qubee" writing system has a total of 33 letters that consists of all the 26 English letters with an addition of 7 combined consonant letters. All the vowels in Afaan Oromoo are like that of English, are also vowels in "Qubee dubbachiiftuu". Vowels have two natures in the language and they can result in a different meaning. The natures are short for single vowels and long for double vowels. A vowel is said to be short if it is one. If it is two, which is the maximum, then it is called a long vowel. Consider these words: lafa (ground), laafaa (soft). The rest of all alphabets "Qubee" are consonants referred to as "Qubee dubbifamaa". The combined consonant letters are known as "Qubee dacha" and they are ch, dh, sh, ny, ts, ph and zy.

2.7.2. Sentiment in Afaan Oromoo

Sentiment is a way to identify the emotions by tracking the tone, context, and feelings behind the words the document [8] In Afaan Oromoo sentiment can be identified at different level for instance we can categorise the full document in to single polarity of the sentiment and also the sentences or the aspect can be categorized in to polarities. For example.

“Uummanni miidiyaa hawaasaa baayyinnaan fayyadama” this atatement is neutral statement

“Uummanni miidiyaa hawaasaa haalaan yoo fayyadame bu’uu guddaa irraa argata” this statement positive sentiment .

“Uummanni miidiyaa hawaasaa karaa sirrii hin taaneen yoo fayyadame miidhaa guddaan biyyarra gahuu danda’a” this statement is Negative statements .

CHAPTER THREE

METHODOLOGY

3.1 Introduction

Under this chapter, we presented the methodology of the Development of Sentiment Analysis by stating the research design, process model, ways of data collection, and sampling with their techniques. It focused on the design issues and process model, algorithms, and techniques that are used for preprocessing and texts for sentiment analysis before it's further processed by applications of feature extraction to feed the machine learning or neural network.

The methodologies we applied for designing, developing, and implementing for Automated machine Learning Based Sentiment Analysis of Afaan Oromoo(AMLSA) underwent four phases; namely data collection phase, preprocessing phase, feature extraction phase, and classification modeling phase. Data collected from the popular Facebook page that uses Afaan Oromoo, Preprocessing is the process of filtering and clearing the collected textual data. Then the preprocessed data become underwent feature extraction techniques. Then classification conducted and the model was developed based on the features extracted from the labeled text sample. The matching is a process of computing a matching score, which is a measure of the similarity of the features extracted from the unknown text sample and compare with a reference stored in the labeled dataset model for the prediction.

3.2 Research design

After spending time reviewing the literature by reading and digesting in a particular research area, it needs further investigation, and identified some potentially important variables and probably has become familiar with the methods commonly used to measure that behavior. Research design is the main technique or strategy that we chose to integrate different elements of the research in a coherent and logical way for ensuring effectively addressing the research problem to the establishment of the variables.

Selecting a suitable research design is very important to the success of the project. The determinations we make at this stage of the research process to determine the quality of the conclusions can draw from our research results. As a scientific study, this research exploratory data collection and analysis, this is aimed toward classifying behaviors within a given area of

research, identifying potentially important variables, and identifying relationships between those variables and the behaviors. Finally, the researcher has chosen a research design for this case of study quantitative, experimental approach by exploratory data collection and analysis to address the research problem which was mentioned in chapter one.

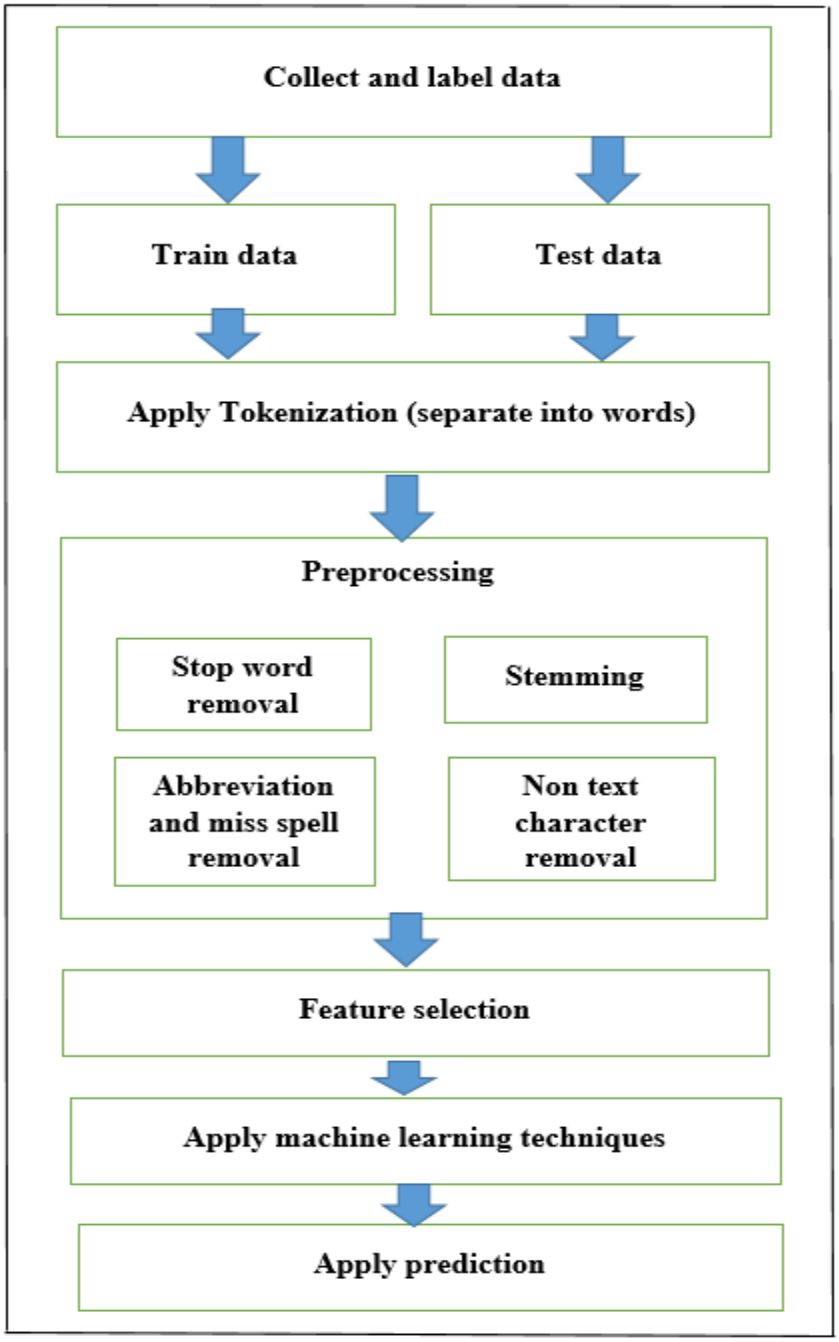

Figure 3. 1 Research Design Methodology

3.3 Building a Dataset

This study is intending to analyze the sentiment of Afaan Oromoo. So, the new dataset is needed because there is no data set label for this purpose formerly. The process of building the dataset for Afaan Oromoo consists of the following main steps:

- Collect manually by copy-pasting the Afaan Oromoo post from the selected Facebook page such as OBNafaanOromoo, BBCafaanOromoo, and FBCafaanOromoo
- Conduct preprocessing: Preparing, filtering, or consolidating gathered data into one file dataset of comma-separated value (CSV), and
- Label the dataset into its respective polarities (positive, negative, and neutral).

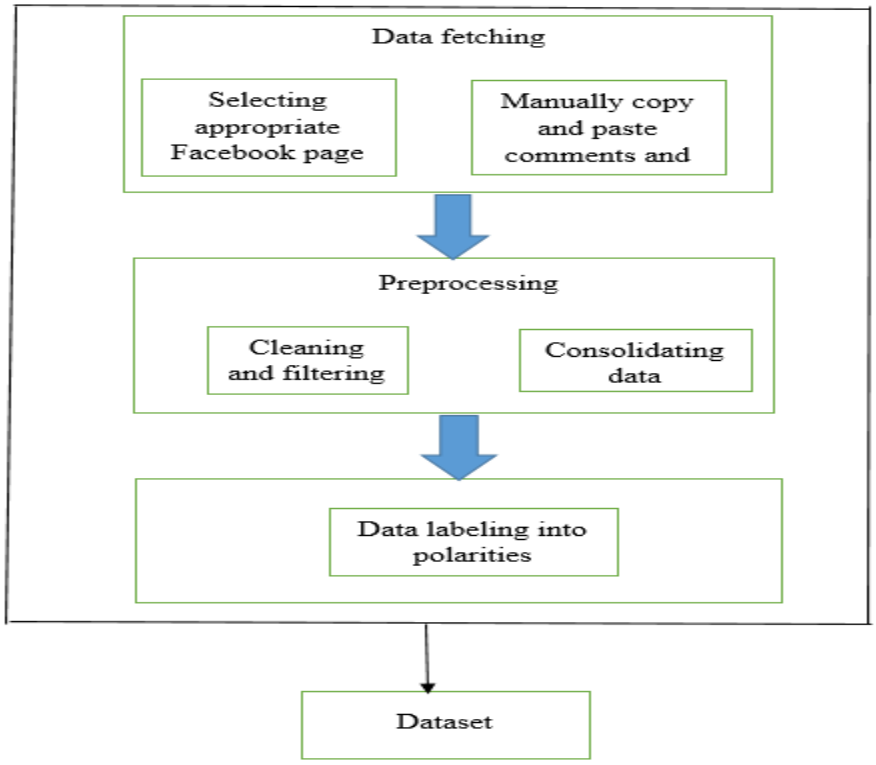

Figure 3. 2 Dataset building method

Data fetching: Data has been fetched from comments and posts public Facebook pages have been selected based on socio-political information they broadcast in the form of textual data. Because the study aims to have balanced socio-political data which may result in a balanced dataset.

Preprocessing: The preprocessing component is mainly concerned with the process of cleaning the input texts to the greater extent of analysis. Social media texts usually contain several less useful characters such as hashtags (#), HTML scripts, special characters, URL, and others. The

noisy or less useful texts are removed while tokenization component and prepare the texts into more useful formats. This component contains two subcomponents: tokenization and normalization.

Data Labeling: the data labeling process has been described in section 3.3.3

3.3.1 Data collection

The study gathers an Afaan Oromoo textual data that is a post and comment from the Facebook platform. those Afaan Oromoo posts and comments were gathered from large media categories of a popular public page because Facebook privacy policy does not allow access to the public content of a private page and also to come up with quality data because those media are legally accessible and their posts and comments are balanced of polarity (positive, negative, and neutral).

To collect the comments and posts public Facebook pages have been selected based on socio-political information they broadcast in the form of textual data. Because the study aims to have balanced socio-political data which may result in a balanced dataset. Various selection of a suitable sample for study metrics or criteria that can be used to select these pages or a user on a social media platform is there. This study carefully weighed the following sampling criteria and metrics used for selecting a public page from the many existing Facebook pages:

- Those pages use Afaan Oromoo frequently for their working purpose of posting and comments.
- The page those who working on socio-political issues purposely the balanced post and comments data.
- Facebook pages, which have a large number of audience, which means large numbers of followers and page likes. Each Facebook page has greater than a half-million followers.
- Facebook pages, which concerned as to give adequate data, due to frequently posting socio-political issues.

The reason why I decided to use Facebook from other social media platforms is that the popularity of the platform, and according to multiple statistics preform in Ethiopia social media. Facebook has 61.3% of the total social media market share, and around over five million users in Ethiopia [54].

Table 3. 1 selected page information

Name of the facebook pages	Number of Page followers	Numberas of Page Likes
BBC News Afaan Oromoo	794,441	648,456
OBN Afaan Oromoo	719,970	569,742
FBC Afaan Oromoo	575,377	530,948

For this study purpose, we gather posts and comments from public pages which has a large number of followers and likes that are limited to larger than half a million, because of a lack of resources needed to collect from those Facebook pages we collected manually by copy-pasting the posts and comments. We did not collect all the posts and comments of the selected page sequentially on time series, but we collected the selected posts and comments randomly because of the manual collection mechanism. The collection concern is on the time interval of September 2019 to March 2020. The seven months 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 because of it the period of the election in the country.

3.3.2 Dataset

After the data collected, the next steps are as follows, 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 or labeling of the post and comments in the dataset and then after used for designing a training model for Afaan Oromoo Sentiment Analysis based guideline we have prepared for this study. The following tasks performed to prepare the dataset for labeling:

- All none Afaan Oromoo and non-textual posts and comments can be removed.
- Blank values and whitespace all Null values automatically removed from the data.
- Combine the data of each page into one dataset that was initially collected separately with an independent text file.

All the above preparation of the dataset considers the nature and behaviors of the Afaan Oromoo language to keep the context of each text in a dataset for the labeling processes. The number of collected and preprocessed data that is posts and comments described in chapter five.

3.3.3 Dataset Annotation

Labeling is a series of actions for adding information to the collected data or document at some level. For this research study, the dataset labeling process needs to label collected posts and comments from Facebook pages to build a dataset for Afaan Oromoo Sentiment Analysis. For this study, we use the sequential labeling method for each collected data from those Facebook pages. This sequential technique helps to label all the filtered posts and comments on each page. The labeling has been carried on based on the instruction guideline provided by the researcher and other participants. The guideline has been prepared by referring to some related scientific works[55], [56], [57] and discussion with participants.

3.3.3.1 The Labeling process

The ultimate goal of labeling or annotating is to label the raw data into three polarities. Those polarities are positive, negative, and neutral classes. This process, mainly carried off by the researcher, and also there were three additional annotators who perform annotation or labeling. The number of participants of labeling or annotating was limited because of a lack of resources, mainly budget and time need for more annotators to participate in the process. Those participants were chosen based on their interests and skill to perform the task and Afaan Oromoo language skills. All the annotators are postgraduate students at Adama Science and Technology University. The annotators were instructed to annotate a random subset of an equal number of posts and comments from the dataset. Those participants were given instruction to annotate the same number of equal instances of posts and comments to measure the inter-rater agreement, which shows the agreement level their annotation decisions match. Due to the challenging nature of the manual annotation process of Sentiment analysis 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 of guideline that provided for this research paper by expert and scientific community into the annotation guidelines provided for labeling posts and comment into three different classes, as defined in Appendix A. The number dataset annotation with their respective number of polarities described in chapter 5.

3.4 Sentiment classification

Since we are using conventional machine learning and neural network technics the **Discourse patterns** can be suitable for sentiment classification techniques. According to this pattern, sentiment can be exactly annotated or labeled based on the real emotion rather than counting or

measuring polarity terms in the sentences or phrases. This way the Coherence relations interact with negation in interesting ways, because the model will be effective based on the pattern train. This pattern considers wider text, rather than considering the weight of words.

After we have extracted statements (words, phrases, and sentences) from a text, with or without having used a pruning method for sentences, the next step is to aggregate the semantic orientation, or evaluative value, of those individual words.

3.4.1 Preprocessing

In the preprocessing components, we are performing the process of cleaning the input texts to further analysis. Obviously, Social media texts usually contain many unnecessary or less useful texts such as hashtags, HTML scripts, special characters, URLs, and others. To get a suitable format of dataset the noise and unnecessary texts are removed in the tokenization component and prepare texts for use. This preprocessing process component has two subcomponents (the detailed discussion of these subcomponents are discussed under chapter four). The following methods used in the prepossessing processes:

- ✓ Removing inappropriate or less useful characters, punctuations, symbols, blank value and whitespace, URL, and emojis.
- ✓ Removing elongation from the collected data post and comment.
- ✓ Performing Tokenization to get the individual word or tokens in text.
- ✓ Performing text normalization.

3.4.2 Feature Extraction Methods

Feature extraction is the process of identifying the features to be used for learning. The word description and behaviors of the patterns are experienced in feature extraction. However, for the categorization purpose within, it is necessary to extract the features to be used. Feature extraction can sometimes be referred to as feature selection methods, which are based on state-of-the-art text mining; techniques applied for reducing redundant features, and dimensionality. The methods used in selecting a subset of appropriate features that would help in identifying positive, negative, and neutral polarities from the provided dataset and can be used in the modeling of the sentiment analysis. For this research work, we used TF-IDF and word2vec feature extraction technique because they are better and popular feature extraction methods used for sentiment analysis, text mining, and text classification.

N-Gram: N-grams are one of the most used techniques of natural language processing specifically on textual data like text classification and sentiment analysis. N-gram is a word prediction model using probabilistic methods to predict the next word after observing N-1 words. The most common N-grams approach consists of combining sequential words into lists with the 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 to them as unigram, bigram, and trigram, respectively. This study uses a word N-gram method to create an N-gram of dataset features. These three n-grams are used because when the number of N increases, the model performance remains constants.

TF-IDF: This Term is stands for Term Frequency-Inverse Document Frequency, which is a well-recognized method to evaluate the importance of the specified word in the given document[37]. The TF-IDF contains two matrices. The first one is containing the inverse document(TF) which can be calculate the frequency of a word in the whole corpus of documents and the second can calculate the term frequency of each word in each document[39][40]. The formulae to calculate both of them are as follows:

$$\mathbf{TF(word, Doc)} = \frac{\boldsymbol{The\ Frequency\ of\ the\ word\ in\ the\ Doc.}}{\boldsymbol{No.\ of\ word\ in\ the\ Doc.}} \qquad \textit{Equation 3. 1}$$

$$\mathbf{IDF(word)} = \mathbf{log_e}\left(1 + \frac{\boldsymbol{No.of\ Docs.}}{\boldsymbol{No.of\ DocsWithwords.}}\right) \qquad \textit{Equation 3. 2}$$

So TFIDF can be calculated as:

$$\mathbf{TF - IDF} = \mathbf{(word, doc)} * \mathbf{(word)} \qquad \textit{Equation 3. 3}$$

For Sentiment analysis and other textual data research's TF-IDF is a better way to convert the textual data into a Vector Space Model (VSM). Suppose there is a document that contains 500 words and out of these 500 words the term "*ummata*" appears 40 times, then Term Frequency will be 40/500=0.08 and suppose there are 100,000 documents and out of these only 1000 documents contains the term "ummata". Then IDF (*ummata*) = 100,000/1000=100, and TF-IDF (*ummata*) will be 0.08*100= 8.

Word2vec: The Word2Vec model was proposed by Mikolov. With the enhanced feature of vector, representation is capable of having the syntax and semantic meaning of natural language [36]. it takes a large corpus of text as input and produces a vector space that will be used when training a

model for Afaan Oromoo Sentiment Analysis. The models word2vec are capable of having the semantic similarity between words based on the circumstances in which these words occur. The Word2vec involving by projecting words into an n-dimensional space, and giving words with similar contexts similar places in this space. **Word2vec** is a two-layer neural network model that operates the texts by "vectorizing" words. It takes input in a text dataset and it gives the output as a set of vectors: feature vectors that represent words in that corpus. While **Word2vec** is not a deep neural network, it turns text into a numerical form that deep neural networks can understand[38]. Simply the model gives relationships between the words in the text. So it helps to get a feature model not only from the labeled dataset but also from the totally collected textual data from Facebook pages for this study.

3.4.3 Classification Algorithm

3.4.3.1 Machine Learning

Sentiment analysis is predominantly it is a kind of text classification; we applied several technics of supervised machine learning approaches[26]. The objective of sentiment analysis in Afaan Oromoo is to predict the category of user's opinion into its polarities (positive, negative, and neutral). To meet this aim we use some of the appropriate technics such as Support Vector Machine (SVM) multinomial Naive Bayes, logistic regression, Random Forest, and some other Neural Network practical methods[42].

We selected the algorithms because of their good classification performance, and their popularity in solving Sentiment Analysis and text classifications on previous research work in related other languages and English languages [49],[58]. The following three machine learning algorithms are used to build Sentiment Analysis models.

Logistic Regression: Logistic regression is an easy and simple to realize text classification algorithm, and Logistic regression can be easily inferred to multiple classes.

Support Vector Machine (SVM): An SVM does the classification tasks by building hyperplanes in a multidimensional space that separates cases of different class labels. Hyperplanes are used as decision boundaries that help classify the data points[37], [59]

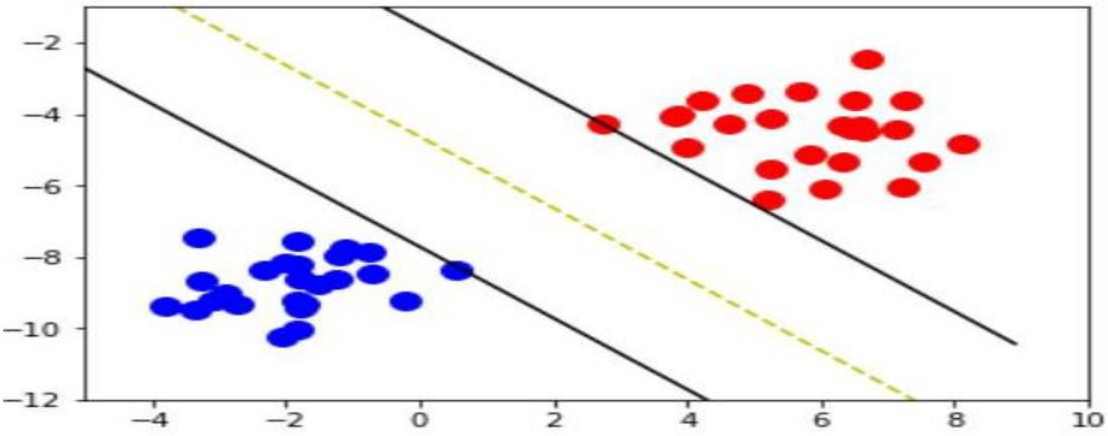

Figure 3. 3 SVM Hyperplane[60].

Originally the SVM technics used when the data has exactly two classes, like binary classification problems, but in this article, we'll focus on a multi-class support vector machine because in this study we are labeling our dataset into three classes, which is positive, negative, and neutrals. We applied the One-vs-Rest strategy of SVM technics to support multiclass classification.

$$w * x + b = 0 \qquad \textit{Equation 3. 4}$$

Where:

w is 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,

$$f_{w,}(x) = g\ (wx + b) \qquad \textit{equation 3. 5}$$

On this one we consider f(*x*) >0 for points on one side of the hyperplane, and f(*x*) <0 for points on the other.

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[61]. 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[39].

Naïve Bayes (NB): Naive Bayes Classifier: the one most suitable for word counts is the multinomial variant: 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[58].

$$(c|x) = (x|c) / (c)(x) \qquad \textit{Equation 3. 6}$$

Where, P(c|x) is the posterior probability of class c given to data x, P($x|c$) is a 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.

3.4.3.2 Neural network classifications

Long Short-Term Memory LSTM: Which is used for learning long-distance dependency between word sequences in short texts. After each word represented by its corresponding vector trained by the Word2Vec model, the arrangement of words terms {T_1, …, T_n} is input to LSTM one by one in a sequence.

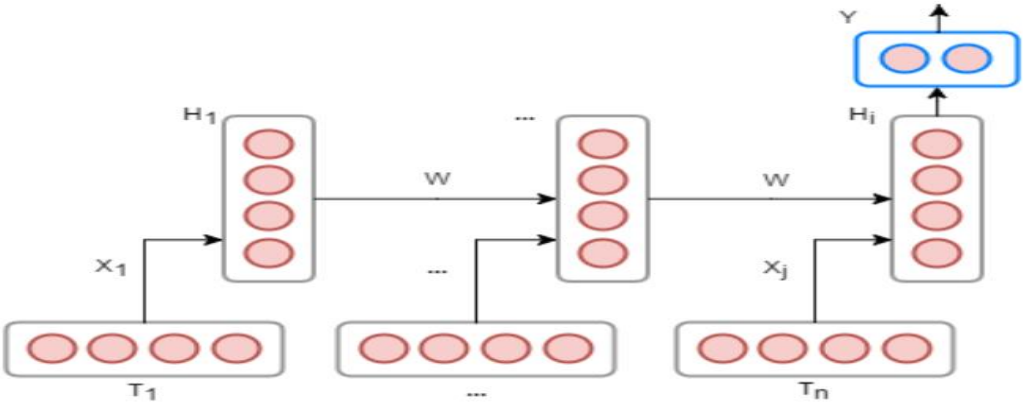

Figure 3. 4 How LSTM works [62].

As shown in figure 8 each term T_i is first changed to the corresponding input vector x_i by using Word2Vec feature extracting model and input into LSTM separately[30]. At each time *j*, the output *W* of the hidden layer H_j will be spread back to the hidden layer together with the next input x_{j+1} at the next point of time *j+1*. At the end, the final output Wn will be given to the output layer[34].

To suits the sequential input of LSTM, we first convert the Dataset into three-dimensional matrix M(X, Y, Z), where X is the dimension of the Word2Vec feature extraction model, Y is the number of words in the Dataset, and Z is the number of Dataset. To reduce the very long training time, we adopt a single hidden layer neural network. The number of neurons in the input layer similar to the dimension of the Word2Vec feature extraction model and the number of neurons in the output layer is the number of polarities, which is 3 in this case. By gradient-based

backpropagation through time, we will adjust a load of edges in the hidden layer at each point of our time. We will obtain the sentiment classification model after several epochs of training.

Convolutional Neural Network CNN: in natural language processing recently, there is a convolutional layer to make a piece of words can be considered together[63]. In this study, we propose an approach to parsing a provided dataset of Afaan Oromoo to understand the situation in the real world based on a CNN model to do the sentiment analysis[63]. We used google collab to run LSTM and CNN code online.

3.5 Evaluation of the system

The target is assessing the performance of the sentiment through several significant positions in sentiment analysis. This measurement is very tight related to accuracy evaluating for sentiments. The performance evaluation plays a vital role in accuracy measurement through a sentiment analysis at all levels. In this study, an evaluation was conducted on multiple phases of the development of the model for sentiment analysis.

3.5.1 Inter annotator agreement

The most important thing of data reliability is to assess the quality of manually annotated Dataset. To correctly annotate our dataset there must be an inter-annotator agreement. This agreement helps to decide whether a certain observation belongs the one or to another category. This study uses Cohen's kappa to measure the agreement level between annotators. Cohen's kappa defines chance as the statistical independence of the use of labeling categories by the annotators. It maintains that chance labeling is ruled by prior distributions that are specific to each annotator (annotator-bias) [64]. To label the dataset for this study all annotators were given the same instruction with the related quantity of collected data to be labeled. To measure the kappa of the agreement between the annotator all the data would be received from annotators. Kappa value has the range between 0 to1, which can be interpreted as the following table.

Table 3. 2 Inter annotator agreement kappa value standards

Agreement interpretation	Kappa values
Less than chance agreement	Less than 0.0
Slight agreement	Between 0.0 and 0.2
Fair agreement	Between 0.2 and 0.4

Moderate agreement	Between 0.4 and 0.6
Good agreement	Between 0.6.and 0.8
Very Good agreement	Between 0.8.and 1.0
Perfect agreement	1.0

3.5.2 Evaluation Metrics

This study presents several metrics for AMLSA of Afaan Oromoo performance measurement. These measurements were applied to different sentiment analysis techniques. The performance criteria consist of two types of performance. These perspectives include F-measure, accuracy, Recall, precision, and confusion metrics [44][65]. This solution can introduce an alternative solution for measuring sentiment accuracy and provide logical meaning in less time and higher accuracy. The first perspective is F-Measure that is a measure of a test's accuracy and depends on both precision and recall. It is one of algorithmic efficiency to examine and evaluate resource usage. Let say a group of reviews has positive sentiment (belongs to class positive), reviews have neutral sentiment (belongs to neutral) and reviews have negative sentiment (belongs to class negative). Then, in their relation to class positive: **TP**: True positive, **TN**: True Negative, **FP**: False Positive, and **FN**: False Negative. Where each of them can be defined as:

- ✓ True Positives TP: when both actual class and predict class are True
- ✓ True Negatives TN: when both actual class and predict class are False
- ✓ False Positives FP: when actual class is False and predicted class true
- ✓ False Negative FN: when actual class is true and predicted class false

Precision: It calculates the amount of correctly assigned sentences in relation to all automatically classified sentences for a given sentiment. Precision can be calculated as follows

$$\text{Precision} = \frac{TP}{TP+FP} \qquad \textit{Equation 3. 7}$$

For example, a high value for *precision* show that a very high number of sentences that are assigned to the sentiment by the algorithm are classified correctly. In contrast, a low value of *precision* show that a high number of sentences are not classified correctly.

Recall: it calculates the amount of correctly assigned sentences in relation to all sentences classified in the real world data for a given sentiment.

$$\text{Recall} = \frac{TP}{TP+FN} \qquad \textit{Equation 3. 8}$$

A high value for *recall* for positive sentences show that most of the truly positive posts are also classified correctly as being positive. A low value for *recall* show that the share of the automatically and correctly classified sentences for a sentiment in relation to all sentences of this sentiment is low.

Accuracy: It indicate the rate at which the method predicts results correctly. It can be calculate as total number of the model correct prediction divide by all number of data instances used for the model as follows:

$$\text{Accuracy} = \frac{TP+TN}{Total\ instsances\ (TP+FN+TN+FP)} \qquad \textit{Equation 3. 9}$$

F-Measure: some time it known as F1-score or F-score. The F measure is a harmonic mean of Precision and Recall. It can be formulated as:

$$\text{F-Score} = 2*\left(\frac{Recall*precision}{Rescall+Precision}\right) \qquad \textit{Equation 3. 10}$$

Confusion matrix: Confusion matrix also known as contingency table is generally used in supervised machine learning approach for allowing visualization of performance of algorithm shown in Table 3.2 From classification point of view, True Positive (TP), True Negative (TN), True Neutral (TN), False Positive (FP), False Negative (FP) False Neutral (FN) are used for comparison label of the classes.

Table 3. 3 Confusion Matrix

		Predicted Polarities		
		Positive	Negative	Neutral
Actual polarities	Positive	True Positive	False Negative	False Neutral
	Negative	False Positives	True Negative	False Neutral

	Neutral	False Positives	False Negative	True Neutral

CHAPTER FOUR

PROPOSED SOLUTIONS

4.1 Introduction

The main objective of this study is to provide a tool that helps to analyses Sentiment in Afaan Oromoo. Under this chapter, the proposed solution of an automatic Afaan Oromoo sentiment Analysis is discussed in detail. This research study provides a new dataset that has been collected from selected Facebook pages. Those datasets can be used to come up with the new proposed solution Automated Machine Learning based sentiment Analysis of Afaan Oromoo.

4.2 The architecture of the System

The proposed automated Machine Learning based sentiment analysis (AMLSA) for Afaan Oromoo system, has four main components. These are preprocessor, Feature extraction, model building, and model testing and evaluation. Each component of the system with its function is discussed in detail in this Section. The general architecture of (AMLSA) for the Afaan Oromoo system is shown in Figure 9. As we can observe from this architecture the system begins with accepting the collected data the perform the necessary preprocesses like removing a special character, removing numbers, removing Afaan Oromoo stop words, removing emoji's, removing abbreviation and miss-spelled. After preprocess performed annotate or label the data, and then process feature extraction, and then apply the selected machine learning and neural network algorithms, finally analyze the model with incoming new data and then predict polarities whether it is positive, negative, and neutral.

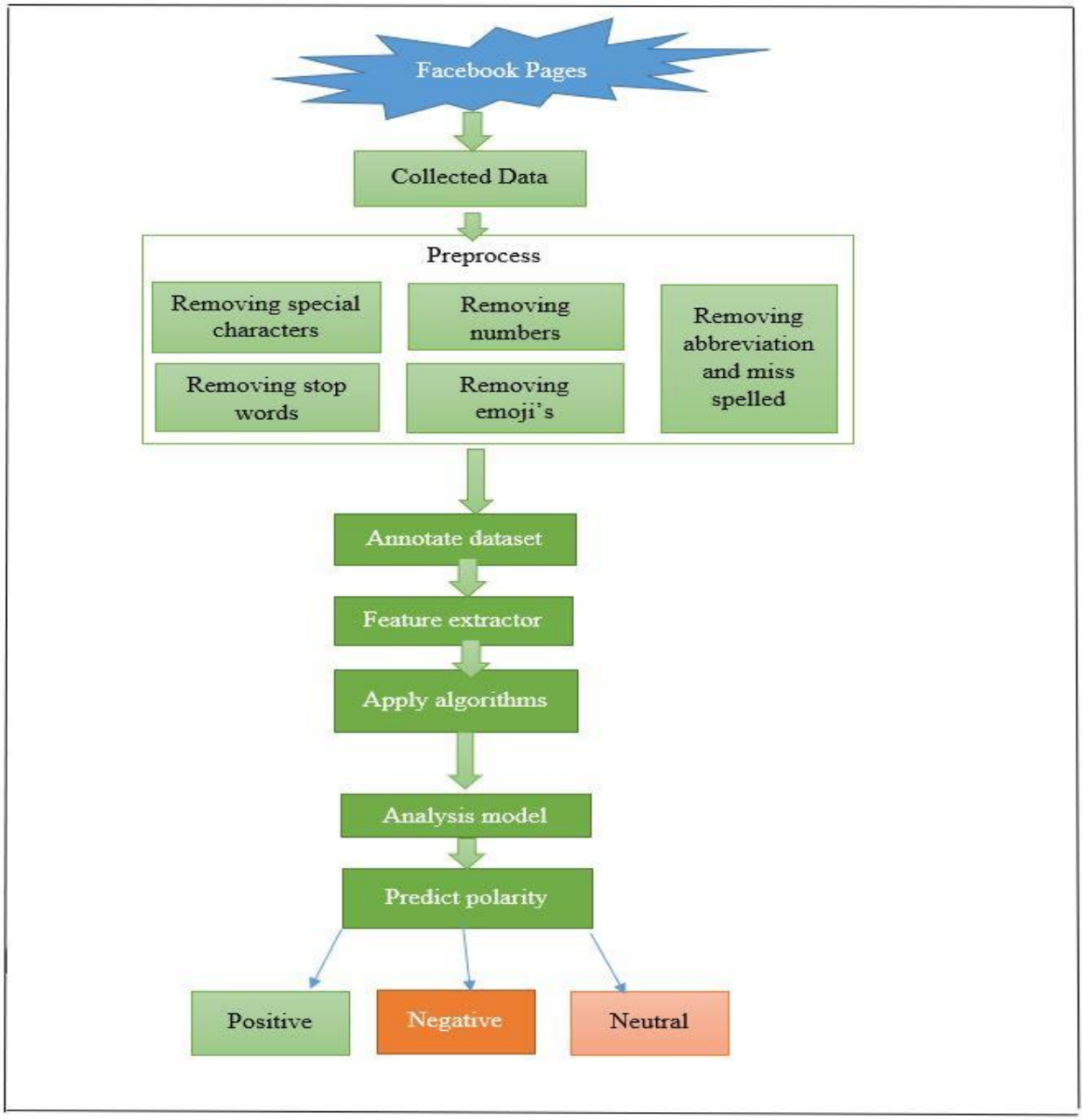

Figure 4. 1 General Architecture of the System

4.3 Preprocessing Component

The preprocessing component is mainly concerned with the process of cleaning the input texts to the greater extent of analysis. Social media texts usually contain several less useful characters such as hashtags (#), HTML scripts, special characters, URL, and others. The noisy or less useful texts are removed while tokenization component and prepare the texts into more useful formats. This component contains two subcomponents: tokenization and normalization.

4.3.1 Afaan Oromoo Text Preprocessing

Under this subcomponent, the preprocessing of the Afaan Oromoo dataset for the training and testing detection model was performed. This preprocessing is based on the Latin alphabet characters and basic text preprocess techniques such as removing(cleaning) punctuations and special characters), normalization, and tokenization for the reason that Afaan Oromoo writing system uses the Latin alphabet characters.

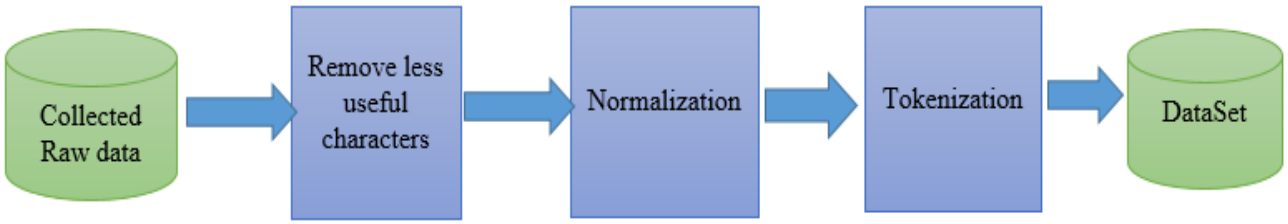

Figure 4. 2 Dataset preprocessing steps

4.3.2 Removing less useful (irrelevant) characters and Emoji's

Since the dataset is collected from social media specifically from Facebook, it may contain texts and some other less useful (irrelevant) characters such as special characters, punctuation marks, symbols, emojis, and symbols. So it needs to be removed those irrelevant characters.

Pseudocode to remove irrelevant characters

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 text contain number = [0-9] then

Remove number

If a text contains emoji = [...EMOJI...] then

Remove emoji

If a text contains extra white space, then

Trim the text

3. Return clean_text;

End:

4.3.3 Tokenization

To split the statements into a separate part of words or tokens by using space between words, tokens, or punctuation mark, tokenization takes place. This is a process to take place after the cleaning and normalization tasks. In the feature extraction process, tokenization is important because the meaning of a statement can depend on the relationship between word arrangements, this helps to get meaningful data. Tokenization method help to grasp appropriate feature from the provided dataset. In the tokenization process, input texts are tokenized into a sequence of tokens by detecting word boundaries from the written texts. In many languages, white space and punctuation marks are used as boundary markers, the same is true for Afaan Oromoo language that the punctuation marks are used to demarcate or separate words, sentences, etc. into a stream of characters. Punctuation marks like period (.), comma (,), and semicolon (;) are also included to demarcate words.

NLTK is a python library used for tokenizing input texts into lists of words. In this study, tokenization is the first step in the preprocessing component. In this step, input texts are tokenized into the stream of characters using white spaces and punctuation which helps to convert into a list of words.

For example: - when the sentence "*mana bareeda ijaare*" tokenized, the result will be

["mana", "bareeda", "ijaare",].

4.3.4 Removal of Stopwords

In preprocessing step removing stop words is very crucial to reduce the running speed. However, in Afaan Oromoo some stopwords change the polarity of the sentiment. For example *"qotiyyoo akka gaaritti qotu bite"* and *"qotiyyoo akka gaaritti* ***hin*** *qotne bite"* here two statements are opposite to each other due to the stop word "hin" only. Therefore removing all Afaan Oromoo stop words are not appropriate. For this reason we removed the stop words carefully, based on Dr. Debela Tesfayes' Afaan Oromoo stop word lists [66].

4.4 Feature Extraction

A feature refers to the information that could be extracted from any data sample in Machine Learning. A feature uniquely depicts the properties that the data possessed. The data used in

machine learning comprises features designed onto a high dimensional feature space. These high dimensional features must be represented onto a lower number of small dimensional variables that maintain information about the data as much as possible. Feature extraction is one of the dimensionality reduction technics[40]. The proposed feature extractions component of the sentiment analysis system performs the extraction of important features of a dataset. As depicted in figure 11 the extraction system takes the input of prepossessed and tokenized dataset words and performs extractions.

In this study, several techniques of feature extraction are used TF-IDF and word2vec. the following figure shows the steps involved in extracting features using both the techniques. Finally, the extracted features can be used to train the models of sentiment classification and to predict the polarities of sentiment whether it is positive, negative, and neutral.

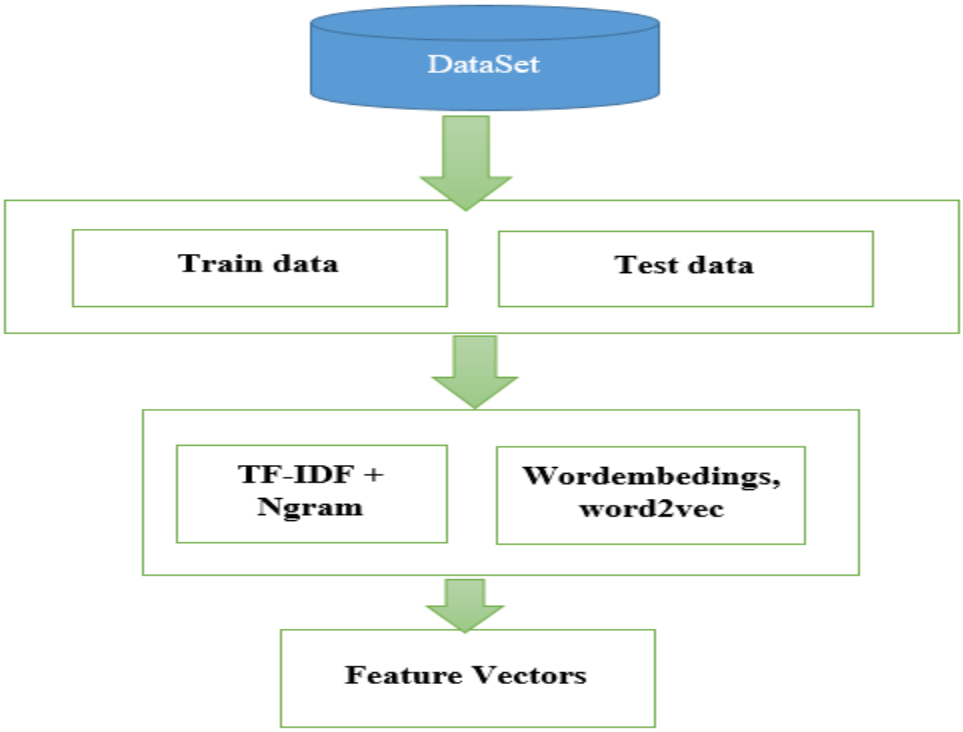

Figure 4. 3 feature extraction Flow diagram

4.4.1 TF-IDF Feature Extraction

As we mentioned in chapter three TF-IDF feature extraction is tried out 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 provided feature model gives the classifiers of the frequency and the importance of the word in the dataset as a feature vector for training.

TF: Term Frequency counts the number of times a particular word or term t occurred in the specified document d. Frequency increases when the term has occurred multiple times.TF is calculated by taking the ratio of the frequency of term t in document d to a number of terms in that particular document d.

IDF: TF counts only the frequency of the word or term t. Some words like stop words may found repeatedly but may not be useful. Hence Inverse Document Frequency (IDF) is used to measure term's importance. IDF gives more importance to the rarely occurring terms in document d.

4.4.2 Word embedding

Word2vec feature extraction is one of the word embedding technics 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 are used for building the model. This method is used due to the absence of the standard Afaan Oromoo word2vec model, and it is recommended that to build domain-related models for better results. The word2vec model containing a vector space and a similarity of all the words in post and comment. These models are 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 method performances. Because one of these study points are to select better feature extraction methods for Sentiment Analysis.

4.5 Models Building

Classifiers are trained on training data sets with the features obtained from the feature extraction techniques TF-IDF and Doc2vec and with the corresponding output labels from the dataset. The test data is tested with the trained classifiers to predict the sentiments and the accuracy is measured for test data. We have chosen the Classifiers from both machine learning and neural network. The selected classifiers are Logistic Regression (LR), Support vector machine (SVM), Random Forest (RF), multinomial Nave Bayes (MNB), convolutional neural Network (CNN), and Long Short-Term Memory (LSTM). The figure 4.4 shows the model buildings

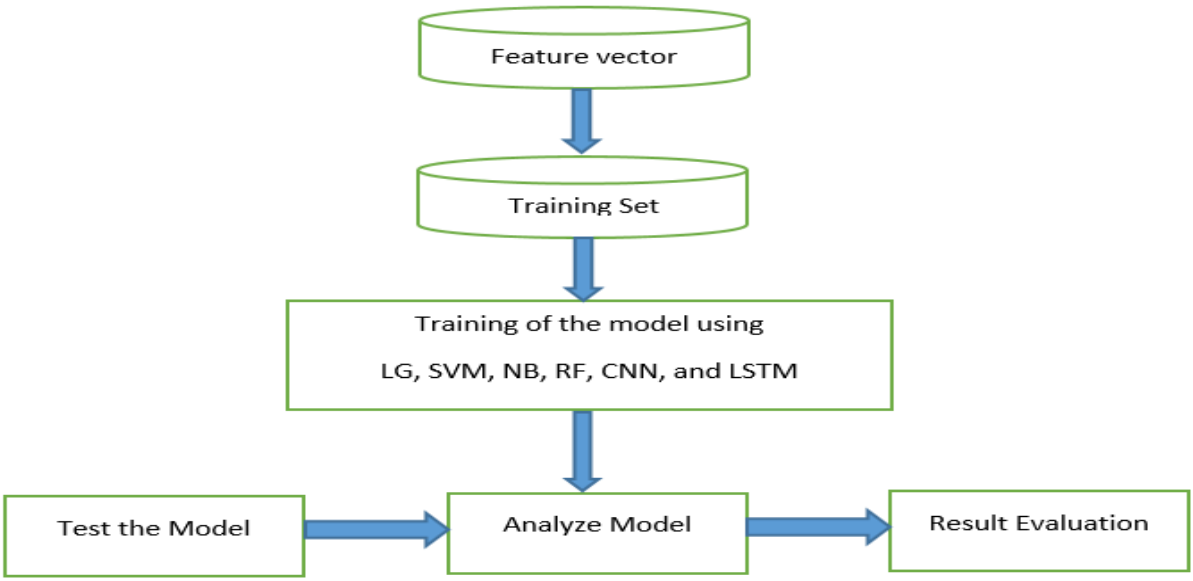

Figure 4. 4 Model building Diagram

4.5.1 Machine Learning Model Building

The proposed machine learning model aims to work with machine learning technics like support vector machine (SVM), logistic regression, random forest, multinomial Naïve Bayes for this study. These technics applied after feature extraction has been identified the feature vector known. The process of modeling is to determine patterns in the training set of the dataset that represents the collected data with their features to the target polarities by using those learning algorithms. The outcome of these modeling is a trained model that can be used for Afaan Oromoo sentiment Analysis, and making predictions on new input data.

Logistic regression: it assesses the relationship between the categorical dependent variable and one or more independent variables by approximating probabilities using a logistic function known as sigmoid function.

Leaner support vector classifier (SVC) of **support vector machine** (SVM) used on this study because SVC support multi-class classification on a textual dataset.

The other proposed algorithm is **multinomial Naïve Bays** (MNB) which is a probabilistic machine learning model that's used for classification. It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of

any other feature. MNB classifier is a specific instance of the classifier which uses multiple distributions for each of the features in the dataset.

The other proposed algorithm is **Random Forest** (RF) classifier, which can be expressed as the collection of tree structured classifiers[61]. The random forest algorithm aggregates multiple algorithms of the same type. That is multiple decision trees, resulting in a forest of trees, hence the name "Random Forest". RF is a Meta estimator that fits a lot of decision tree classifiers on a subsample of the dataset.

4.5.2 Neural Network Model Building

Long Short-Term Memory LSTM: The other model for sentiment classification built in this stage used for this study is LSTM. Which is used for learning long-distance dependency between word sequences in short texts. The model is trained with the training dataset and then is tested and evaluated in the next phases.

Convolutional Neural network CNN: in this model, a text as a sequence is passed to the model of CNN. It has a convolutional layer to extract information by a larger piece of text from feature extracted vector, so in this study we also additionally work for sentiment analysis with convolutional neural network, and we design a simple convolutional neural network model and test it on the standard benchmark provided for this study.

4.6 Models Evaluation and Testing

The role of evaluating and testing is to describe the evaluation metrics of the designed system and followed by its test results. To evaluate and test the machine learning models ability to learn the trained Afaan Oromoo sentiment dataset, we need to check the accurate estimation of the generalization error of the trained model on the provided dataset. In this study, the experiment can conducted based on the sentiment polarity classes and evaluates each evaluation metrics corresponding to each sentiment polarity classes. The sentiment polarity classes can classified into three categories: positive, negative, and neutral classes. To realize this we use different performance evaluation strategies that evaluate the performance of our proposed conventional machine learning and neural network algorithms used in our approach. Moreover, we used metrics such as F-measure (F-score), confusion matrix, accuracy precision and recall,

4.7 World cloud

A world cloud is a collection of randomly arranged words where the size of each word properly related in size to its frequency in the dataset. It give us an idea of what words represent in the dataset of each class but, it do not clearly point out accurate information especially when comparing each class. Just to have an idea of what our final training data looks like, we decided to use word cloud to visualize each polarities with word clouds. Wordcloud show the frequent terms in the dataset, wordcloud graphical representations of word frequency that give greater prominence to words that appear more frequently in a source text. The maximum frequent the word in the visual the more common the word was in the document[67].

Word clouds are an simple to use and not expensive alternetives for picturing the text data. One of the demanding situation of interpreting word clouds is that the visualization conserned on frequency of terms in the dataset, not necessarily their importance. Word clouds will not exactely reflect the content of text if somewhat different terms or words may used for the same idea (for example, ‘gadhee’, ‘hamaa’, ‘badaa’). They also may not allow context, so the meaning of individual words may be changed. Due to these limitations, word clouds are best suited for visualsation of exploratory qualitative analysis[67].

CHAPTER FIVE

EXPERIMENT

5.1 Introduction

In this chapter, we discuss in detail the implementation of the proposed solution for Afaan Oromoo sentiment Analysis by using the methodology discussed in chapter three, and the proposed solution described in chapter four also used. This study follows an experimental approach to determine the best experimental result of Neural Network and the Machine Learning algorithm and features extraction for final models' comparison.

5.2 The Development Environment and tools

To implement Sentiment Analysis in Afaan Oromoo various tools and packages used. We preferably use Python because it is simple yet powerful programing language with excellent functionality for processing machine learning and neural networks. It is highly readable, allows data and methods to encapsulate and contains extensive library including components for graphical programing, numerical programing and web connectivity. This study uses python programing language for implementing and experimenting with each proposed solution in chapter 3 and 4, starting from the data preprocessing to the model building phase. In addition to this, we also used to evaluate the implemented proposed classifiers model.

Sentiment analysis in Afaan Oromoo is 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.

The Tools and package used are described in the table 5.1 below.

Table 5. 1 Tools and packages used

No.	Tools and packages	Descriptions
1.	Python	Python is an object-Oriented, interpreted, high-level programming language with dynamically rich in semantics. It is Powerful programming language to develop a machine learning application to process linguistic data, easy to learn.
2.	Anaconda Navigator	Anaconda Navigator is a GUI that contains the **Anaconda** Individual edition, which makes it easy to launch applications and

		manage packages and environments without using command-line commands.
3.	Google Colab	**Google Colab** is a free cloud service which help us to use free GPU. This can improve efficiency of Python programming language to develop deep learning applications using popular libraries such as TensorFlow, Keras, etc.
4.	Jupyter notebooks	**Jupyter notebooks** is An open-source web application that help us to make and contribution of 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. It has been incorporated with anaconda navigator and we have developed our code on it.
5.	Microsoft Excel	We **Used Microsoft Excel** for data preparation tasks in cleaning filtering and sorting the gather data. Most probably used for manage the labeling or annotation activities.
6.	Microsoft Word	**Microsoft Word** it's a word processor, is arguably the most popular word processor on the planet we used it for Document preparation
7.	Microsoft power point	**Microsoft PowerPoint** is a presentation program, we used it For presentation.
Packages used		
8.	Scikit-learn	**Scikit-learn** provides bunch of built-in machine learning algorithms and models 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.
9.	nltk	**NLTK** is a platform for building Python programs to work with human language data, specifically for textual data. We used it for preprocessing and tokenization.

10.	NumPy	**NumPy** is a library for the Python programming language, adding help for large, multi-dimensional arrays and matrices, along with a large collection of 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.
11.	pandas	**Pandas** is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the **DataFrame**. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.This study uses it for data reading, manipulation, writing and handling the data frame.
12.	RegEx (Re)	A **regular expression** or RE is an excellent tool for text analysis or nlp, it defines 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 text.
13.	gensim	**Gensim** (Generate Similar) is Python library for topic modeling document indexing and similarity retrieval with large dataset. This study uses it for the constricting word2vec model.
14.	matplotlib	**Matplotlib** is plotting library for the **Python** programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits. In this study we used it for data and results visualization
15.	tensorflow	**tensorflow** is a symbolic math library based on dataflow and differentiable programming. We used it for CNN and LSTM neural Networking
16.	Keras	**Keras** is one of python libraries which high-level neural network that runs on top of **TensorFlow**.

5.3 Dataset Description

For this study, different datasets collected from public Facebook pages that uses Afaan Oromoo language to share their idea to the public. The selected Facebook page are Fana broadcasting corporate, Oromia broadcasting corporate, Oromia media network, and BBC Afaan Oromoo Facebook page. Those Facebook page were selected because of their data quality only, those media frequently posts their idea carefully having less typos error and miss spell, and they have large number of followers. Then most of their posts on these pages from September 2019 to April 2020 collected, and also comment under each post has been collected sentiment sentences have been manually collected from those pages to keep quality of data. Subsequently, filter the Afaan Oromoo post and comments by removing all non-textual data. We have collected annotated 11941 unique words, and this collected dataset is annotated into 6670 statements of sentiments. Statements are consists of sentences, words, and phrases. These domains were collected because of absence of any prepared available written sentiment datasets in Afaan Oromoo.

After collecting the sentiment data, it has been labeled into three class of polarities: positive, negative and neutral classes. Implementation of the Preprocessing. During implementation of our Dataset, we are using python programing language discussed under subtopic 5.2 and some selected modules. Before implementation of any step we need to import the different modules. The following python libraries were used for conventional machine learning classifiers.

```
import pandas as pd
import numpy as np
from nltk import word_tokenize
from nltk.stem import PorterStemmer
from nltk.corpus import stopwords
import re
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, f1_score, confusion_matrix
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression, SGDClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import LinearSVC
from sklearn.pipeline import Pipeline
import pickle
import pandas as pd
import numpy as np
from nltk.stem import PorterStemmer
from sklearn.model_selection import train_test_split
```

Figure 5.1 important modules for conventional machine learning techniques

After importing whole necessary libraries the next step is load the dataset file to the disk. The loaded dataset using panda used throughout the whole implementation of this study. The figure 5.2 is a code to load data set .

```
data = pd.read_csv('dataset/sentimentdata.csv',header=0,encoding = 'unicode_escape')
```

Figure 5. 2 Code to read dataset

After dataset has been loaded, the next step is to clean it. Cleaning is the process which consist in removing unimportant or disturbing elements for the next stage of analysis. In order to allow for only significant information, in general a clean sentences should not include URLs, hashtags (i.e.#gammadeera) or mentions name of entities(i.e. @voa) punctuation marks, special character, symbol, emojis, numbers, and so on. Furthermore, tabs and line breaks should be replaced with a blank space.

```
def preprocess_and_tokenize(data):
    #remove html markup
    data = re.sub("(<.*?>)", "", data)
    #remove urls
    data = re.sub(r'http\S+', '', data)
    #remove hashtags and @names
    data= re.sub(r"(#[\d\w\.]+)", '', data)
    data= re.sub(r"(@[\d\w\.]+)", '', data)
    #remove punctuation and non-ascii digits
    data = re.sub("(\\W|\\d)", " ", data)
    #remove whitespace
    data = data.strip()
    # tokenization with nltk
    data = word_tokenize(data)
    # stemming with nltk
    porter = PorterStemmer()
    stem_data = [porter.stem(word) for word in data]
    return stem_data
```

Figure 5.3 preprocess and tokenization

We used nltk python module to tokenize the dataset, to come up with tokenized data of our dataset. The tokenized data can be used for feature extraction.

5.4Feature extraction

Feature extraction helps to provide words in their vectored form that can be used for applying selected machine learning algorithms. The python Scikitlearn module for TF-IDF and python Gensim module for word2vec can used for feature extraction purpose in this study. Each feature extraction technics will use the preprocessed and tokenized form of the statements in the data set.

Vector form of the statement is needed, because machine-learning models function with the numbers or vectors for instead of words to train the models.

TF-IDF Implementation: TF-IDF uses *TfidfVectorizer* class of sci-kit learns package. This class changes statements (vector terms) of the dataset to a matrix of TFIDF features vectors. For TF-IDF mode, TfidfVectorizer function class is instantiated and pass the text in the dataset to *fit_transform ()* method of the class.

```
vect = TfidfVectorizer(tokenizer=preprocess_and_tokenize, sublinear_tf=True, norm='l2', ngram_range=(0, 1))
# fit on our complete corpus
vect.fit_transform(data.sentiment)
# transform testing and training datasets to vectors
X_train_vect = vect.transform(X_train)
X_test_vect = vect.transform(X_test)
```

Figure 5.4TF-IDF code

Word2Vec implementation: Word2Vec vectors are generated for each statements in dataset by traversing through the X_train dataset. By simply using the model on each word of the dataset, those words gave the word embedding vectors. For this study, we used a python Gensim module, which is used to implement different embedding methods.

Word embedding: When applying word embedding encoding to words, we come up with sparse (containing many zeros) vectors of high dimensionality task-specific word embedding with the Embedding layer of **Keras**. The code to word embedding matrix is as bellow.

```
def create_embedding_matrix(filepath, word_index, embedding_dim):
    vocab_size = len(word_index) + 1  # Adding again 1 because of reserved 0 index
    embedding_matrix = np.zeros((vocab_size, embedding_dim))
    with open(filepath) as f:
        for line in f:
            word, *vector = line.split()
            if word in word_index:
                idx = word_index[word]
                embedding_matrix[idx] = np.array(
                    vector, dtype=np.float32)[:embedding_dim]
    return embedding_matrix
```

Figure 5.5 Word embedding feature extraction

5.5 Models Implementations

After the feature's extraction process has been done, we used both machine learning and neural network techniques to build the model.

5.5.1 Machine learning model

To use machine learning technics like LG, SVM, MNB, and RF we need to load the feature extracted vectors of data by using x_train_vect and X_test_vect.

Logistic Regression: To implement logistic regression we used LogisticRegression() fuction.

```
log = LogisticRegression(solver='lbfgs', multi_class='auto', max_iter=200)
log.fit(X_train_vect, y_train)
ylog_pred = log.predict(X_test_vect)
```

Figure 5.6Logistic Regression model

Support vector machine: we used *LinearSVC()* classifier of the sklearn package to building the SVM model. Because the model supports both dense and sparse, input and the multiclass support is handled according to a one-vs-the-rest (OVR) scheme. The tolerance for stopping criteria parameter used is (tol=1e-0.5).

```
svc = LinearSVC(tol=1e-05)
svc.fit(X_train_vect, y_train)
ysvm_pred = svc.predict(X_test_vect)
```

Figure 5.7 SVM model

Random forest: we used RandomForestClassifier() function of the sklearn that ensemble the package to train and test the classifier.

```
rf = RandomForestClassifier(n_estimators=50)
rf.fit(X_train_vect, y_train)
yrf_pred = rf.predict(X_test_vect)
```

Figure 5.8 Random Forest Model

Naïve Bayes: To implement Naïve Bayes we *MultinomialNB*() classifier of sklearn , because it suitable for classification of discrete and multi-class data.

```
nb = MultinomialNB()
nb.fit(X_train_vect, y_train)
ynb_pred = nb.predict(X_test_vect)
```

Figure 5.9 Naive Bayes Model

5.5.2 Neural Network models:

In addition to technics listed under this subtitle, we also applied two neural network model CNN and LSTM. The keras and sklearn python module used for neural network implementation. To implement we used the following libraries.

```
import pandas as pd
import numpy as np
# text preprocessing
from nltk.tokenize import word_tokenize
import re
# plots and metrics
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, f1_score, confusion_matrix
# preparing input to our model
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.utils import to_categorical
# keras layers
from keras.models import Sequential
from keras.layers import Embedding, Bidirectional, LSTM, GRU, Dense
```

Figure 5.10 Libraries for neural network

Convolutional Neural Network: The parameters of neural network configuration achieved by try and error and fine-tuning process, because the number of epoch to be iterate will not found exactly at once. To remove unnecessary bias from the network we used maximum dropout (0.2). The parameters of CNN architecture configuration summarized as table below.

Table 5.2 CNN architecture configuration

Trained parameters name	Parameters amount
Dropout	0.2
Epoch	10
Activation	Softmax
Batch-size	64
Embedding dimension	100
Pooling	Globalmaxpooling
Optimizer	Adam

Long Short Term Memory: The other neural network classifiers techniques we used is LSTM, the following table summarizes the architecture we applied for the LSTM model by try and error and fine-tuning.

Table 5. 3 LSTM architecture configuration

Trained parameters name	Parameters amount
Dropout	0.2, recurrent_dropout=0.2
Epoch	7
Activation	Softmax
Batch-size	64
Embedding dimension	300
Pooling	Globalmaxpooling
Optimizer	Adam

5.6 Word cloud

As we described in chapter four word cloud give us an idea of what words represent in the dataset of each class but, it do not clearly point out accurate information especially when comparing each class. Just to have an idea of what our final training data looks like, we decided to use word cloud to visualize each polarities with word clouds. Word Clouds for three of the polarities are shown as follows. Figure 5.11 shows frequently used "Positive" words in the data set, by using the following codes.

```
pos_sentiment = train[train.polarity == 'positive']
pos_string = []
for t in pos_sentiment.sentiment:
    pos_string.append(t)
pos_string = pd.Series(pos_string).str.cat(sep=' ')
wordcloud = WordCloud(width=1600, height=800,max_font_size=200,
background_color='white').generate(pos_string)
plt.figure(figsize=(12,10))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```

Figure 5.11 Positive Sentiments Word Cloud

Figure 5.12 shows frequently used “Newtral” sentiments in the data set, by using the following codes.

```
plt.style.use('fivethirtyeight')
neut_sentiment = train[train.polarity == 'neutral']
pos_string = []
for t in neut_sentiment.sentiment:
    pos_string.append(t)
pos_string = pd.Series(pos_string).str.cat(sep=' ')
wordcloud = WordCloud(width=1600, height=800,max_font_size=200,
colormap='gray',

background_color='white').generate(pos_string)
plt.figure(figsize=(12,10))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```

Figure 5.12 Neutral Sentiments Word cloud

Figure 5.13 shows frequently used "Negative" sentiments in the data set, by using the following codes.

```
neg_sentiment = train[train.polarity == 'negative']
neg_string = []
for t in neg_sentiment.sentiment:
    neg_string.append(t)
neg_string = pd.Series(neg_string).str.cat(sep=' ')
from wordcloud import WordCloud
wordcloud = WordCloud(width=1600, height=800,max_font_size=200,
colormap='magma', background_color='white').generate(neg_string)
plt.figure(figsize=(12,10))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```

Figure 5.13 Negative Sentiment Word Cloud.

CHAPTER SIX

RESULT AND DISCUSSION

6.1 Introduction

In this chapter, we deeply discuss the result of the experiments of the proposed solution for Afaan Oromoo sentiment analysis using machine learning. We used six classifier for this study namely Support Vector Machine (SVM), Random Forest (RF), Multinomial Naïve Bayes (MNB), Logistic Regression (LR), Convolutional Neural Network (CNN), and Long-short Term Memory (LSTM). The result of each classifiers discussed in detail under this chapter.

6.2 Dataset Labeling Result

We have collected 6670 statements of dataset; the statements of collected dataset can be words, phrases, and sentences. These collected dataset classified to four participants of annotators, each participants were given 1500 different datasets, and 670 data were given to all member in common for inter annotator agreement.

Four different annotators have annotated the collected data; the annotators follow the annotation guideline provided for this research work. The detail of dataset annotation was discussed in chapter three subtopic 3.3. The total annotated dataset expressed in the following tables.

Table 6.1 Unique annotated dataset

Annotators	positive	Negative	Neutral	Total unique annotated data
Annotator 1	467	453	580	1500
Annotator 2	567	483	450	1500
Annotator 3	519	487	494	1500
Annotator 4	531	466	503	1500
Total	2084	1889	2027	6000

Table 6.2 common annotated dataset

Annotators	positive	Negative	Neutral	Total common annotated data
Annotator 1	186	321	163	670

Annotator 2	189	213	268	670
Annotator 3	169	193	308	670
Annotator 4	204	227	239	670
Total annotated in common	670			

Table 6.3 Total annotated dataset

Polarity of the class	Total
positive	2270
Neutral	2210
Negative	2190
Total Dataset	6670

Inter annotator agreement has been calculated between the annotator 1 and annotator 2 at the first round, between 2 and3 at the second round. Between 3and 4 at the third round. Finally, the average rate of all annotator has been taken. Inter annotator agreement has been calculated based on Kohen kappa standards. The first result is 0.75, the second result is 0.90, the third result is 0.84, and the average result of annotators is 0.81. According to kappa value interpretation standards discussed in chapter three, it indicates that there is **very good** agreement between annotators.

The kappa results show that the dataset contains few ambiguities between annotators that need or to be improved until there is excellent or perfect agreement reached, because the same statements might differently labeled while having a guideline that specifies how to annotate the three most ambiguous polarity class of Positive, negative and neutral.

6.3 Test Results

In this research work, to test or evaluate the performance of the system and its effectiveness on sentiment polarity classifications, we conducted the experiment using the labeled or annotated collected sentiment datasets. The study model′s performance evaluation metrics, as discussed in chapters three and four. These metrics are Precision (P), Recall(R), F-score (F1), and accuracy. We applied these metrics on different machine Learning approach namely Naïve Bayes, Support

Vector machine, logistic Regression, and Random Forest. The feature extraction vectors used for those classification approaches are TF-IDF in combination with Unigram, Bigram, and Trigram.

Multinomial Naïve Bayes: the performance evaluation result of Naïve Bayes classification approach are described it the table 6.4.

Table 6. 4 Performance of Naive Bayes classification for three different features

Naïve Bayes metrics	Features		
	TF-IDF + Unigram	TF-IDF + Bigram	TF-IDF + Trigram
Precision	78.62	80.43	80.42
Recall	78.32	80.22	79.97
F1-score	78.26	80.08	79.80
Accuracy	78.32	80.32	80.12

The result of Multinomial Naïve Bayes classification performance shows that, the feature of TF-IDF in combination with Bigram has highest result of all involved metrics as compared to the combination of TF-IDF with Unigram and Trigram. The f-score of TF-IDF with combination of bigram is 80.08. Whereas, TF-IDF with combination of Unigram is 78.26 and TF-IDF with combination of Trigram is 78.80. We compared F1- score metrics selected to compare the features because it is more useful than other metrics when the dataset contains uneven class distribution. The confusion matrix for Naïve Bayes classification of three different features are shown on fig bellows.

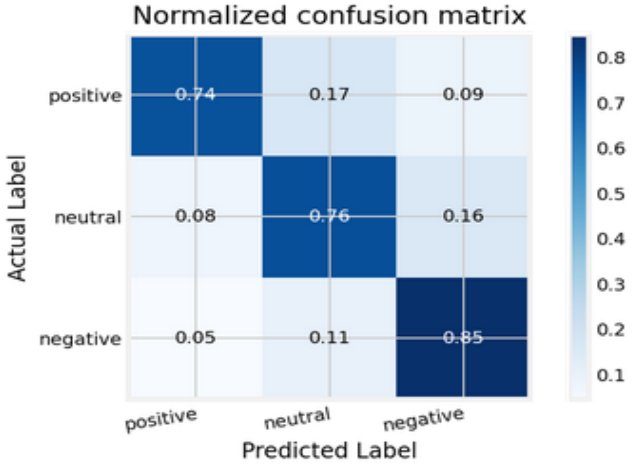

Figure 6.1 confusion matrix of Naive Bayes of TF-IDF +Unigram

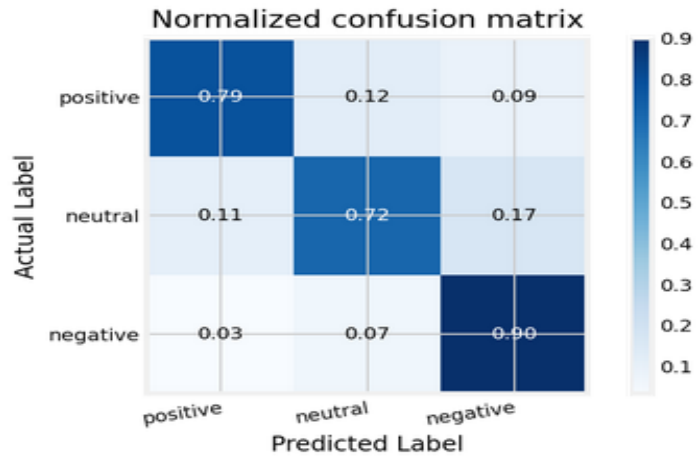

Figure 6.2 confusion matrix of Naive Bayes of TF-IDF +Bigram

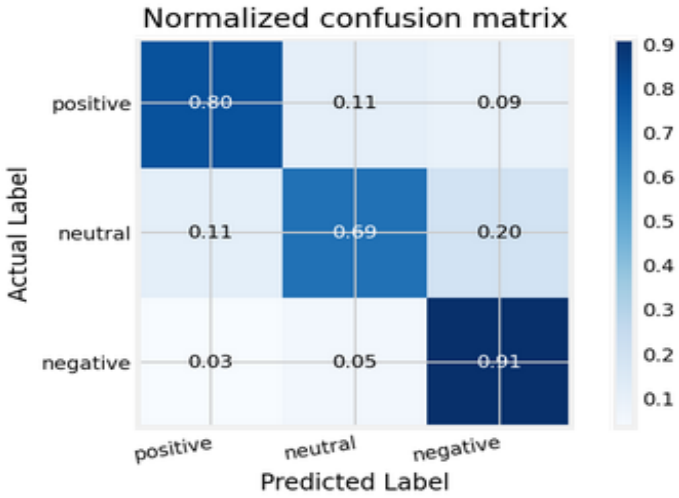

Figure 6.3 confusion matrix of Naive Bayes of TF-IDF +Trigram

Here we have three confusion matrix for the multinomial Naïve Bayes, let elaborate the second one that is the feature of TF-IDF in combination with Bigram. As depicted on the diagram the Positive polarity classified correctly with 79%, and misclassified with 12% and 9%. The Neutral polarity classified correctly with 72% and misclassified with 11% and 17%. The Negative polarity classified correctly with 90% and misclassified with 3% and 7%.

Random Forest: the performance evaluation result of Random Forest classification approach are described it the table 6.5.

Table 6.5 Performance of Random Forest classification for three different features

Random Forest metrics	Features		
	TF-IDF + Unigram	TF-IDF + Bigram	TF-IDF + Trigram
Precision	80.59	81.34	80.60
Recall	80.32	80.86	80.45
F1-score	80.30	80.98	80.47
Accuracy	80.32	81.02	80.62

The result of Random Forest classification performance shows that, the feature of TF-IDF in combination with Bigram has highest result of all involved metrics as compared to the combination of TF-IDF with Unigram and Trigram. The f-score of TF-IDF with combination of bigram is 80.98%. Whereas, TF-IDF with combination of Unigram is 80.30% and TF-IDF with combination of Trigram is 80.47%.. The confusion matrix for Random Forest classification of three different features are shown on fig bellows.

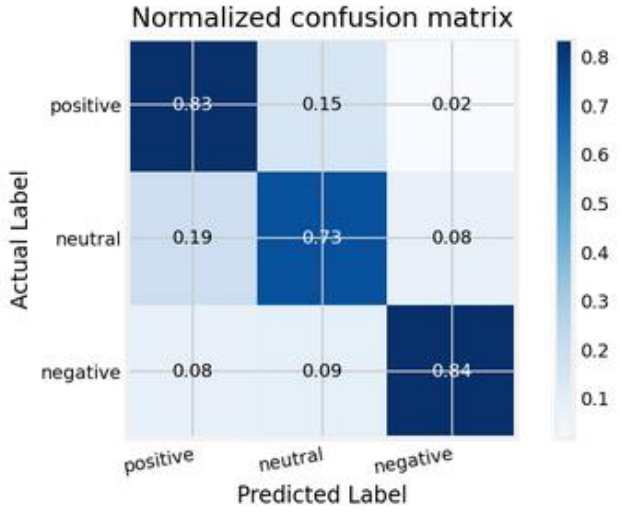

Figure 6.4 confusion matrix of Random Forest of TF-IDF +Unigram

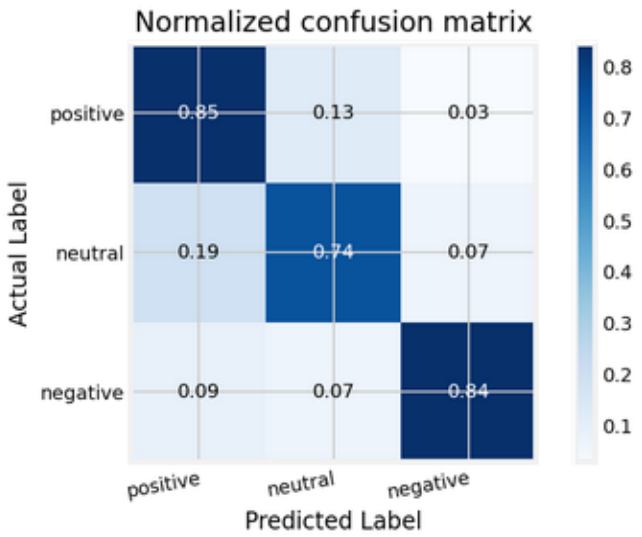

Figure 6.5 confusion matrix of Random Forest of TF-IDF +Bigram

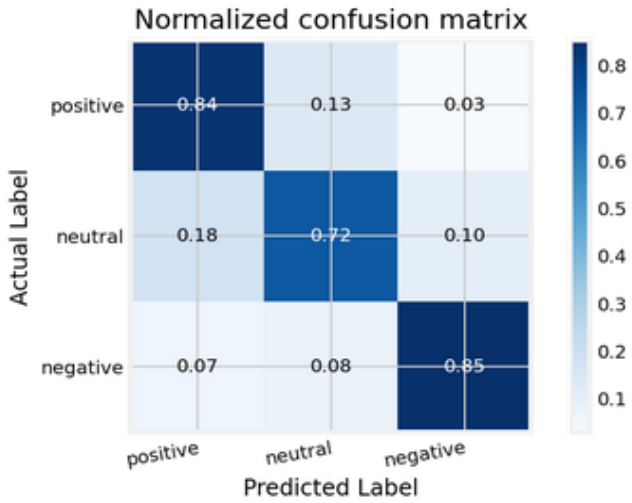

Figure 6.6 confusion matrix of Random Forest of TF-IDF + Trigram

From confusion matrix of three Random Forest, let elaborate the second one that is the feature of TF-IDF in combination with Bigram because the second confusion matrix has highest accuracy as compared to the other features. As depicted on the diagram the Positive polarity classified correctly with 84%, and misclassified with 13% and 3%. The Neutral polarity classified correctly with 72% and misclassified with 18% and 10%. The Negative polarity classified correctly with 85% and misclassified with 7% and 8%.

Logistic Regression: the performance evaluation result of Logistic Regression classification approach are described it the table 6.6

Table 6.6 Performance of Logistic Regression classification for three different features

Logistic Regression metrics	Features		
	TF-IDF + Unigram	TF-IDF + Bigram	TF-IDF + Trigram
Precision	79.14	81.70	82.47
Recall	79.32	81.68	82.40
F1-score	79.09	81.68	82.43
Accuracy	79.32	81.82	82.52

According to the result of Logistic Regression classification performance, the feature of TF-IDF in combination with Trigram has highest result of all involved metrics as compared to the combination of TF-IDF with Unigram and Bigram. The f-score of TF-IDF with combination of Trigram is 82.43%. Whereas, TF-IDF with combination of Unigram is 79.09% and TF-IDF with combination of Bigram is 81.68%.. The confusion matrix for Logistic Regression classification of three different features are shown on fig belows.

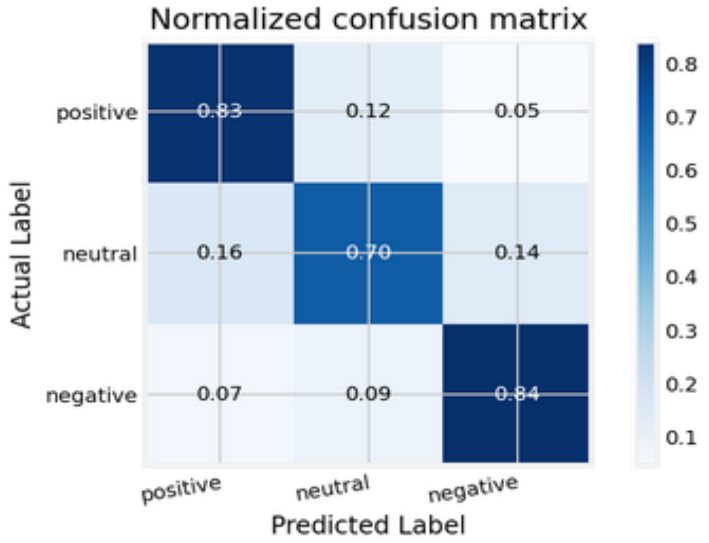

Figure 6.7 confusion matrix of Logistic Regression of TF-IDF +Unigram

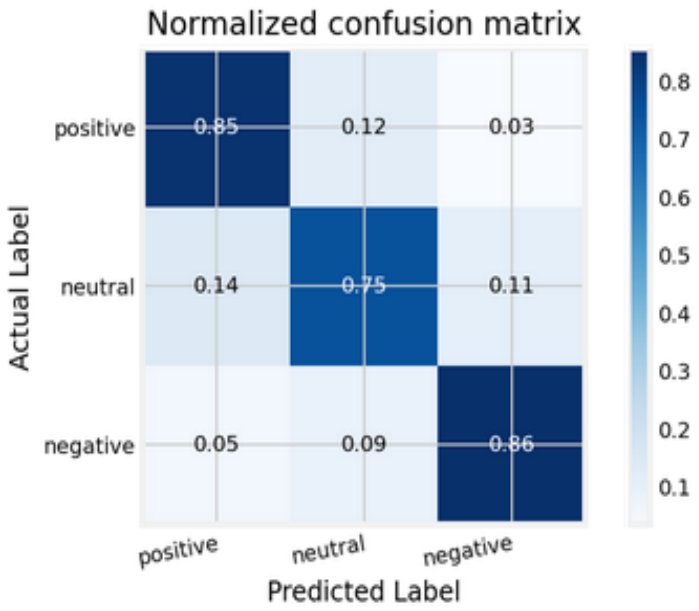

Figure 6.8 confusion matrix of Logistic Regression of TF-IDF +Bigram

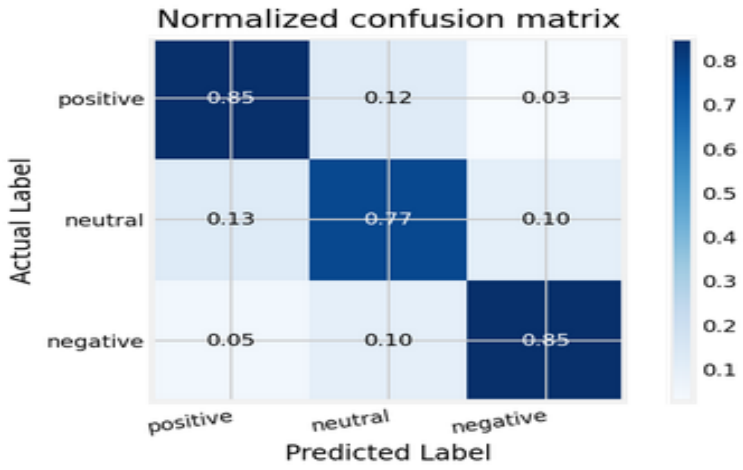

Figure 6. 9 confusion matrix of Logistic Regression of TF-IDF +Unigram

The performance of Logistic regression increase with the feature of TF-IDF in combination with Unigram, Bigram, and Trigram in the respective order. Therefore, the third confusion has the highest performance. As depicted on the diagram the Positive polarity classified correctly with 85%, and misclassified with 12% and 3%. The Neutral polarity classified correctly with 77% and misclassified with 13% and 10%. The Negative polarity classified correctly with 85% and misclassified with 5% and 10%.

Support Vector Machine: the performance evaluation result of SVM classification approach are described it the table 6.7.

Table 6. 7 Performance of Support Vector Machine classification for three different features

SVM metrics	Features		
	TF-IDF + Unigram	TF-IDF + Bigram	TF-IDF + Trigram
Precision	83.35	84.69	84.50
Recall	83.34	84.65	84.47
F1-score	83.30	84.65	84.48
Accuracy	83.52	84.82	84.62

According to the result of Support Vector Machine classification performance, the feature of TF-IDF in combination with Bigram has highest result of all involved metrics as compared to the combination of TF-IDF with Unigram and Trigram. The f-score of TF-IDF with combination of Bigram is 84.65%. Whereas, TF-IDF with combination of Unigram is 83.30% and TF-IDF with combination of Trigram is 84.48%.. The confusion matrix for Support Vector Machine of three different features are shown on the following figure.

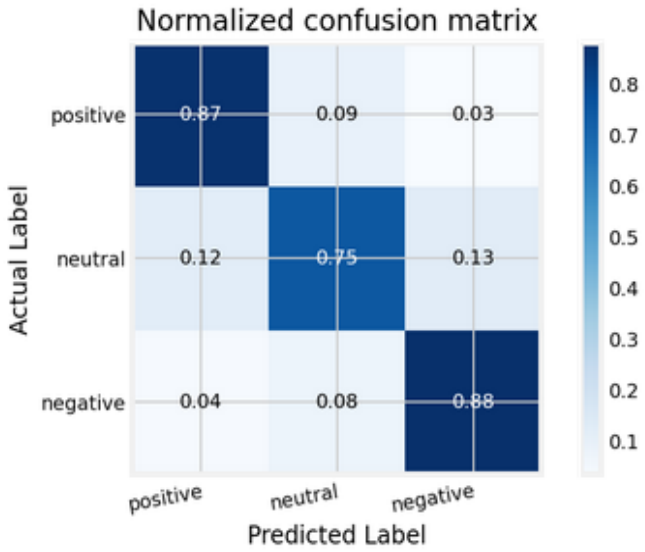

Figure 6.10 confusion matrix of SVM of TF-IDF +Unigram

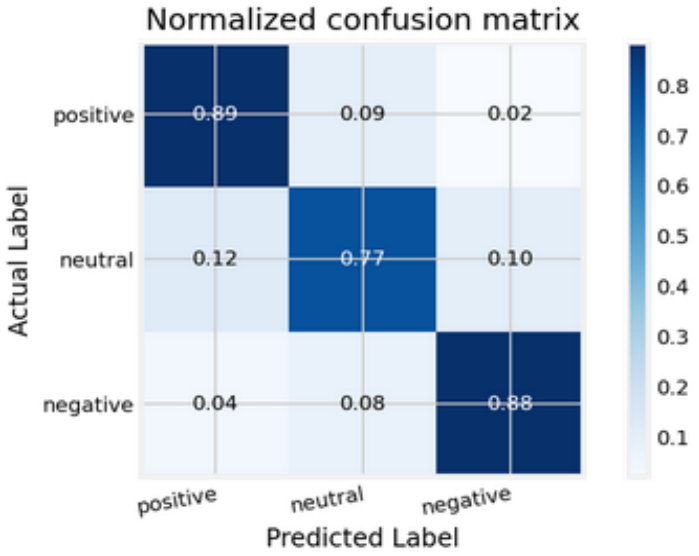

Figure 6. 11 confusion matrix of SVM of TF-IDF +Bigram

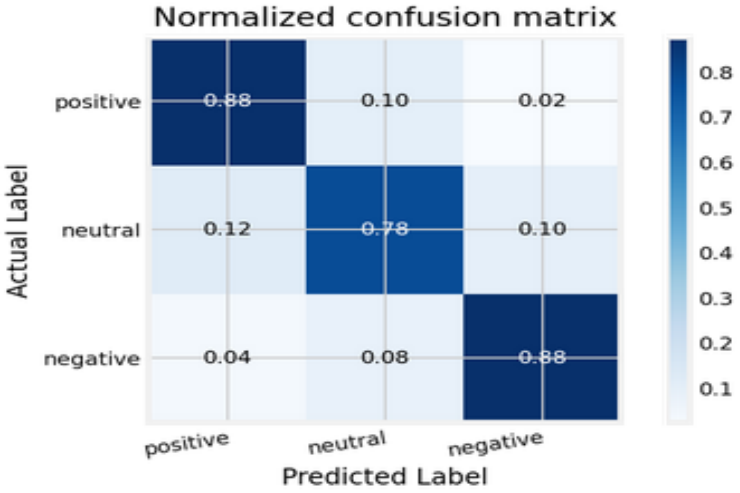

Figure 6.12 confusion matrix of SVM of TF-IDF +Trigram

From confusion matrix of three Support Vector machine, let elaborate the second one that is the feature of TF-IDF in combination with Bigram because the second confusion matrix has highest accuracy as compared to the other features. As depicted on the diagram the Positive polarity classified correctly with 88%, and misclassified with 10% and 2%. The Neutral polarity classified correctly with 78% and misclassified with 12% and 10%. The Negative polarity classified correctly with 88% and misclassified with 4% and 8%.

Convolutional Neural Network: CNN is one of the model we used in this research work, this model come up with result of each polarity namely positive, negative, and neutral polarities. The evaluation metrics applied involved for performance evaluation of this models are as follows:

Table 6. 8 CNN performance Evaluation of each polarities

polarities	Evaluation Metrics		
	Precision (%)	Recall (%)	F-score (%)
Positive	77	61	68
Neutral	78	71	74
Negative	67	87	76

The overall performance evaluation summarized as follows:

Table 6.9 CNN performance evaluation of overall dataset

Evaluation Metrics	Percentage (%)
Precision	74.07
Recall	72.91
F-score	72.64
Accuracy	72.91

The proposed CNN mode achieved accuracy of 72.92%, F-score 72.64%. The strength of this model is its capacity to handle the contextually of new entry because its dataset was labeled or annotated contextually, and it can easily recognize the pattern of data's.

The confusion matrix of the CNN model is as follows

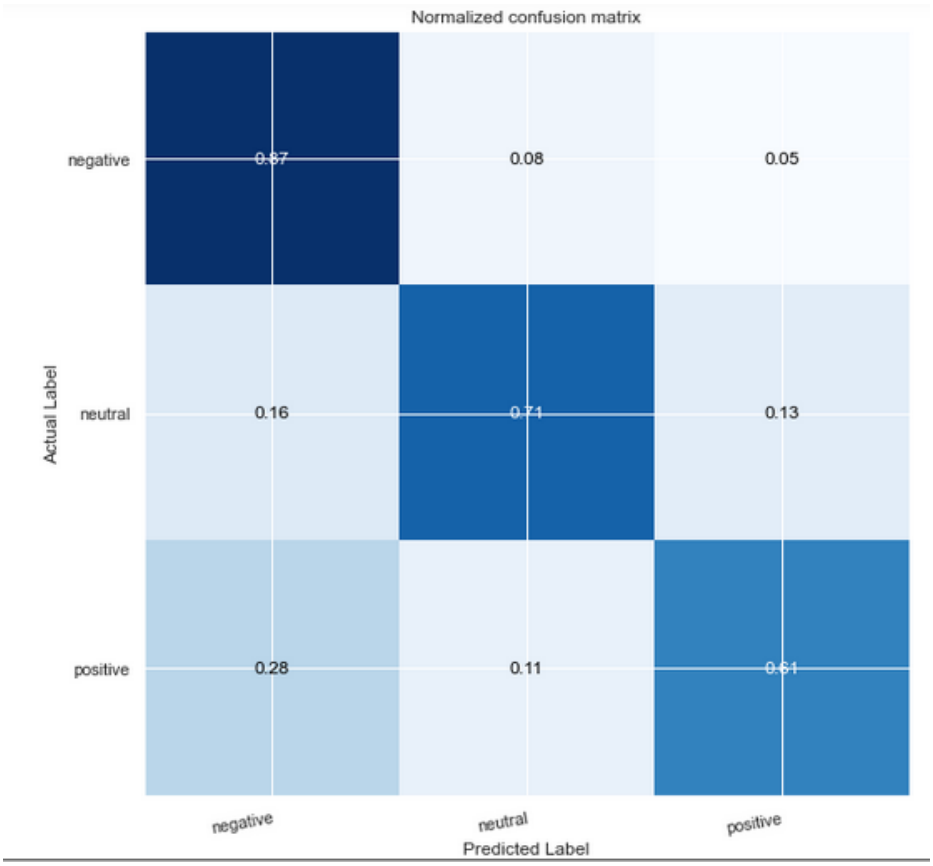

Figure 6.13 Normalized confusion matrix of CNN model

From confusion matrix of CNN model, we can observe that the Positive polarity classified correctly with 61%, and misclassified with 28% and 11%. The Neutral polarity classified correctly with 71% and misclassified with 16% and 13%. The Negative polarity classified correctly with 87% and misclassified with 8% and 5%. The Advantage of this model over the conventional machine learning model is the capability to handle the context of the statements because neural network handle the pattern of the statements based on their contexts.

Long Short Term Memory: The other neural network techniques we used for this study is LSTM. This model achieve the accuracy of 85.01% based on the architecture discussed in chapter five, the loss value of this model is 0.47. the following figure shows the accuracy and loss of both train and test value of LSTM model.

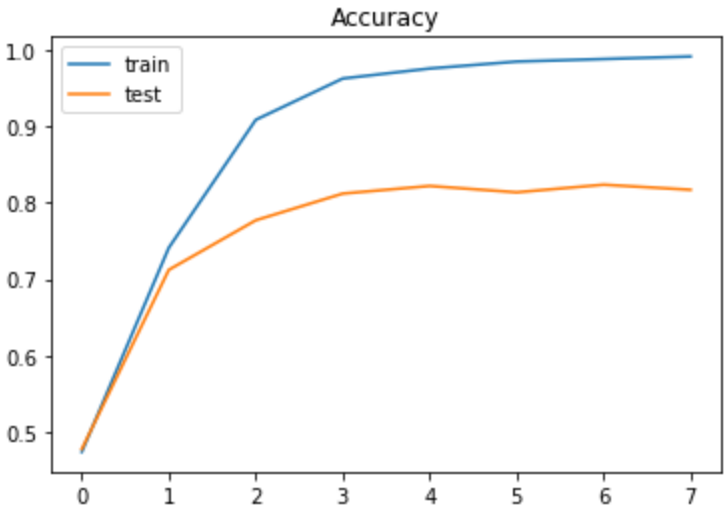

Figure 6. 14 Accuracy value diagram of LSTM classification

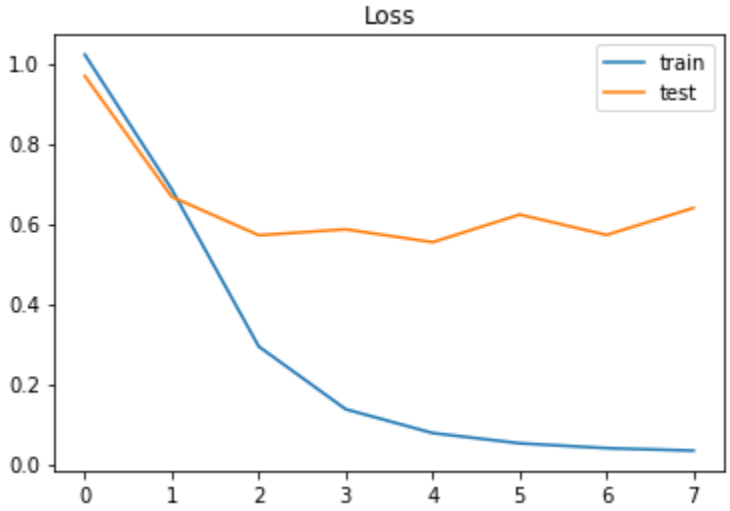

Figure 6.15 Loss value diagram of LSTM classification

6.4 Performance Evaluation with new entry data

We checked the performance of our model that built from LSTM by entering new data (data that not used for model buildings) we entered twenty-five statements and twenty two of them classified correctly and three of them classified wrongly. The LSTM classifier selected due to highest result of accuracy of this classifiers out of all classifiers used for this study. From this, we can perceive that it classified 88% correctly. The data we inputted and the result predicted is as follows on the figure.

```
        Your Sentiment :Obbo Taaggasaa Caafoon gitasaanii Yuugaandaa Ribqaa Kaadaagaa waliin mari'atan
predicted Polarity [[8.9006186e-05 9.9923491e-01 6.7612220e-04]] NEUTRAL
        Your Sentiment :Af yaa'iin Mana Maree Bakka Bu'oota Ummataa FDRI obbo Taaggasaa Caafoon gitasaanii Yuugaandaa Ribqa
a Kaadaagaa waliin mari'atan
predicted Polarity [[3.6877038e-05 9.9989367e-01 6.9459311e-05]] NEUTRAL
        Your Sentiment :Af yaa'iiwwan lameen hariiroo ummataa caalmaan cimsuuf kan hojjetan ta'uu mari'ataniiru
predicted Polarity [[1.5073191e-04 3.5138283e-02 9.6471101e-01]] POSITIVE
        Your Sentiment :Muummee ministiraa Dooktar Abiy Ahimad kaleessa Noorweey Ooslootti argamuun badhaasa Noobeelii Nage
enyaa fudhachuun isaanii ni yaadatama
predicted Polarity [[5.7264388e-04 1.5655244e-02 9.8377210e-01]] POSITIVE
        Your Sentiment :Itti aanaan kantiibaa magaalaa Finfinnee Injiinar Taakkalaa Uuman Faanaa Broodkaastiing Koorporeeti
tti akka himanitti, muummee ministiraatif booru simannaan ni taasifama
predicted Polarity [[3.5911979e-04 9.9922848e-01 4.1240721e-04]] NEUTRAL
        Your Sentiment :Simannichas Bulchiinsi Magaalaa Finfinnee fi Waajjirri Muummee Ministiraa qopheessaniiru
predicted Polarity [[0.00163368 0.9902445  0.00812187]] NEUTRAL
        Your Sentiment :Jiraattonni magaalaa Finfinnee fi godina addaa Oromiyaa naannawa Finfinnee simannicharratti akka hi
rmaatan waamichi dhiyaatera.
predicted Polarity [[0.00306289 0.9949635  0.00197369]] NEUTRAL
        Your Sentiment :Maqa oromotin nagaduun haadhabbaatu bilxigiinnaan hasimatu
predicted Polarity [[0.9100177  0.08267332 0.00730903]] NEGATIVE
        Your Sentiment :lubbu keynaa kenine isa simana Abichun keyna nagadhan qofa nu hadhufu
predicted Polarity [[0.2362693  0.7454117  0.01831903]] NEUTRAL
        Your Sentiment :Sirriidha simanaan godhamuu qaba,garuu eegumsi cimaan Abbichuudhaaf godhamuu qaba
predicted Polarity [[8.5014489e-04 9.9794680e-01 1.2030943e-03]] NEUTRAL
        Your Sentiment :Baga Gammaddan,Baga gammanne Ilmaan Oromoo hundinuu
predicted Polarity [[5.0432285e-05 7.7474699e-04 9.9917477e-01]] POSITIVE
        Your Sentiment :Bagaa gammadee Abbichuu
predicted Polarity [[0.00439511 0.10238515 0.89321977]] POSITIVE
        Your Sentiment :gaaffii ummata Oromoo akka ofii barbaadutti osoo hin taanee akka ummani Oromoo barbaadutti deebisi
predicted Polarity [[1.8542741e-03 9.9783105e-01 3.1467495e-04]] NEUTRAL
        Your Sentiment :Goota baddaa oromiyaa kan baga nuuf dhalatte
predicted Polarity [[1.6720686e-04 6.3757123e-03 9.9345714e-01]] POSITIVE
        Your Sentiment :Nuti Oromoon Nama Qumaara Taphatu Wajjin Qumaara Taphachu Him Feenu
predicted Polarity [[0.00542937 0.9396619  0.05490876]] NEUTRAL
        Your Sentiment :ABO dabalatee paartileen morkattootaa waliin hojjechuuf waliigalan
predicted Polarity [[7.069519e-04 9.987006e-01 5.923812e-04]] NEUTRAL
        Your Sentiment :Paartileen walitti dhufiinsa taasisan maqaa waloo ni baafatu jedhameera
predicted Polarity [[0.00157782 0.9080812  0.09034107]] NEUTRAL
        Your Sentiment :Waggaa 3 dura magaalaa Adaamaatti Riqichi ijaarrame jedhamee guyyaa jala bultii Eebba isaatti dhaba
me turuun isaa kan yaadatamuudha
predicted Polarity [[0.3644725  0.15332434 0.48220313]] POSITIVE
        Your Sentiment :Yakkoota itti yaadamee, qorannoo fi qindoominaan raawwatamuuf ture hambisuun danda'ameera
predicted Polarity [[0.99666816 0.0022871  0.00104467]] NEGATIVE
        Your Sentiment :Sadaasa dhumarraa eegalee barattoonni Yuunvarsiitii Dambi Doolloo butamanii ugguramaniiru
predicted Polarity [[0.14034724 0.8557701  0.00388265]] NEUTRAL
        Your Sentiment :Jilli ministira Raayyaa Ittisa Biyyaa FDRI obbo Lammaa Magarsaan duurfamu daawwannaaf Faransaay jir
a
predicted Polarity [[1.3700679e-04 9.9973398e-01 1.2897751e-04]] NEUTRAL
        Your Sentiment :Daawwannaa kanaan itti gaafatamtoota Waajjira Preezdaantii Faransaay, Ministira Raayyaa Ittisa Biyy
aa, ministir deettaa Ministeera Dhimma Alaa fi Itaamaajoorii biyyattii waliin dhimmoota waloo irratti ni mari'atu jedhameer
a.
predicted Polarity [[7.2559451e-05 9.9990916e-01 1.8257793e-05]] NEUTRAL
        Your Sentiment :Dhaabbileen barnootaa dhaloota biyya ijaaru uumuuf hojjechuu qabu
predicted Polarity [[2.4670262e-05 1.2992333e-03 9.9867612e-01]] POSITIVE
        Your Sentiment :Ministeerri Aadaa fi Turiizimii dhaabbilee barnootaa fi qooda fudhattoota waliin ga'ee dhaabbileen
barnootaa ijaarsa biyyaa keessatti gumaachuu qaban irratti marii gaggeessaa jiru
predicted Polarity [[2.5053229e-04 9.9942493e-01 3.2448352e-04]] NEUTRAL
        Your Sentiment :Sirni barnootaa biyyattii xinsammuu dhaloota roga gaariin ijaaruu irratti xiyyeeffatee hojjechuu ak
ka qabus ibsameera
predicted Polarity [[6.6825916e-04 7.2169756e-03 9.9211472e-01]] POSITIVE
```

Figure 6. 16 Evaluation of the model with real time data

6.5 Discussions

Sentiment analysis is a process of intelligently extracting information from user's opinions, to know feelings or emotion of intended group or individuals. Currently it is difficult to handle the people feelings or sentiment due to bulkiness nature of data over social media. The main goal of this study is to design and develop an automated machine learning-based Sentiment Analysis for Afaan Oromoo. To meet this objective we accomplished several tasks. First, we gather data from public Facebook page. Then, the collected data has been labeled or annotated based on guideline prepared for this study. The next step is preprocessing the data to filter out less useful data and to come up with quality data. During preprocessing, we applied several task like removing Afaan Oromoo stop words, hash tags numbering system, emojis, and trim off spaces. After preprocessing of data, we applied several feature extraction techniques to vectorize our datasets. For feature extraction, we used techniques like TF-IDF in collaboration with unigram, bigram, and trigram we applied word embedding of word2vec. Based on performance evaluation result TF-IDF in collaboration with bigram achieved highest accuracy. After our dataset vectorized we applied several algorithms. We used four conventional machine learning techniques like SVM, MNB, LR, RF, and two Neural Network techniques like CNN and LSTM.

Based on the performance evaluation LSTM achieves highest accuracy. To evaluate the performance of each techniques, the we used several performance evaluation metrics such as Accuracy, F-score, Precession, and Recall. The MNB technique achieved accuracy 80.12%, F-score 79.80%, precision 80.42%, and recall 79.97% by using trigram with TF-IDF feature extraction techniques. The LR technique achieved accuracy 82.52%, F-score 82.43%, precision 82.47%, and recall 82.40% by using trigram with TF-IDF feature extraction techniques. The RF techniques achieved accuracy 80.62%, F-score 80.47%, precision 80.60%, and recall 80.45% by using trigram with TF-IDF feature extraction techniques. The SVM techniques achieved accuracy 84.62%, F-score 84.48%, precision 84.50%, and recall 84.47% by using trigram with TF-IDF feature extraction techniques. From the neural network techniques, CNN achieved accuracy of 72.91% by using word2vec feature extraction techniques, and LSTM achieved accuracy of 85.01%. The summary of this performance evaluation is described in table 17

To check the performance of our model we manually fed twenty-five different statements to our system and it classifies twenty-two correctly and classifies three statements incorrectly. These means 88% present of our newly entered data classified correctly.

Where P: Precision, R: Recall, F1: F-score, and Ac: Accuracy

Table 6.10 Summary of Performance Evaluation

Techniques	Feature extractions													
	TF-IDF + unigram in present (%)				TF-IDF + bigram present (%)				TF-IDF + trigram present (%)				Word 2vec	Word embedding
	P	R	F1	Ac	P	R	F1	Ac	P	R	F1	Ac	Ac	Ac
SVM	83.35	83.34	88.30	88.52	84.69	84.65	84.65	84.82	84.50	84.47	84.48	84.62	-	-
LR	79.14	79.32	79.09	79.32	81.70	81.68	81.68	81.82	82.47	82.40	82.43	82.52	-	-
RF	80.59	80.32	80.30	80.32	81.34	80.86	80.98	81.02	80.60	80.45	80.47	80.62	-	-
MNB	78.62	78.32	78.26	78.82	80.43	80.22	80.08	80.32	80.42	79.99	79.80	80.12	-	-
CNN	-				-				-				72.92	-
LSTM	-				-				-				-	85.01

CHAPTER SEVEN

CONCLUSION, RECOMMENDATION AND FUTURE WORK

7.1 Introduction

This chapter described the appropriate conclusions from the experimental results, a conclusion based on the investigations, contributions and forward recommendations to those who want study in the area of Sentiment Analysis in Afaan Oromoo and any other necessary for the study.

7.2 Conclusion

Sentiment analysis is a process of extracting information from user's opinions. Many users spent their time on social media for different purposes; this leads them to share their opinion on the website, blogs, microblogs, web forums, and social networks. Therefore, it is difficult to manage this large data for decision-making. In sentiment analysis identifying the polarity (positive, negative and neutral) of data helps for decision-making. Large companies, organization, political parties', government organs use sentiment polarities for decision making. That is why sentiment analysis attractive research area for the researchers.

This Study proposed to design, develop, and implement a Sentiment analysis on social media using Machine Learning techniques. The study undertook to design, develop, implement and compares different techniques under conventional machine learning and neural network techniques. For this study, data collected from selected popular Facebook pages. Total of 6670 statements of data collected and annotated into 2270 positive, 2210 neutral, and 2190 negative based guideline prepared for this study. The collected data set preprocessed and cleared. Several techniques applied for feature extraction namely TF-IDF in collaboration with n-gram, word2vec of word embedding. On this study, TF-IDF in collaboration with bigram achieved better performance of all other feature extraction techniques.

The researcher applied six different classification techniques, four of them are conventional machine learning techniques and two of them are neural network techniques, conventional machine learning techniques are Multinomial Naïve Bayes (MNB), logistic Regression (LR), support vector machine (SVM), and Random Forest (RF). The two neural network techniques are Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) techniques.

To evaluate the performance of each techniques, the researcher used several performance evaluation metrics such as Accuracy, F-score, Precession, and Recall. The feature extraction

techniques used for conventional machine learning techniques are unigram with TF-IDF, Bigram with TF-IDF, and trigram with TF-IDF, and neural network techniques used word embedding of word2vec methods. The MNB technique achieved accuracy 80.12%, F-score 79.80%, precision 80.42%, and recall 79.97% by using trigram with TF-IDF feature extraction techniques. The LR technique achieved accuracy 82.52%, F-score 82.43%, precision 82.47%, and recall 82.40% by using trigram with TF-IDF feature extraction techniques. The RF techniques achieved accuracy 80.62%, F-score 80.47%, precision 80.60%, and recall 80.45% by using trigram with TF-IDF feature extraction techniques. The SVM techniques achieved accuracy 84.62%, F-score 84.48%, precision 84.50%, and recall 84.47% by using trigram with TF-IDF feature extraction techniques. From the neural network techniques, CNN achieved accuracy of 72.91% by using word2vec feature extraction techniques, and LSTM achieved accuracy of 85.01%.

The result shows that the LSTM techniques achieved the highest accuracy of all the techniques used from both conventional machine learning and Neural Network techniques used. Therefore, the researcher decided to use the LSTM technique to deploy the models by using Django web development.

7.3 Contributions

- The researcher attempted to propose, design, develop, implement, and compares conventional Machine Learning and Neural Network techniques, and text feature extraction methods for Sentiment Analysis for specifically Afaan Oromoo.
- The researcher tried to design, to predict the real-time polarities of the sentiment and the tests set to train and test the model simultaneously.
- The researcher tried to come up with the data set of multiple class that contains Positive, Negative, and Neutral classes, whereas the former researchers conducted on binary class.
- The researcher believe that the model helps to build other datasets for further improvements regarding sentiment analysis for Afaan Oromoo.
- The researchers set out guidelines to label the Afaan Oromoo datasets for textual sentiment analysis

7.4 Recommendation and Future work

After we achieved the stetted objectives of the study, future studies has to plans for further exploit its capabilities and make full use of its potential have been laid out below. The researcher would like to recommend the following for future works.

- Obviously, the social media contains erroneous contents we recommend the sentiment analysis system to integrate with miss spelled handling system in Afaan Oromoo.
- Identify other possible used language in Ethiopian like Amharic, Tegrians, Somali, etc. to make multilingual system.
- The acronyms and slang has to expand to their original the acronyms and slang to their original words form utilizing the acronym dictionary.
- Future research should also focus other deep learning techniques by increasing the numbers of the training and testing datasets.
- Sentiment expressed not only textual, therefore other mechanism like emojis, audio, video has to be incorporated for future study.
- In order to achieve better performance we would like to recommend more classification polarities of the sentiments like weak positive, strong positive, weak negative, and strong negative

Reference

[1] Y. Getachew, "DEEP LEARNING APPROACH FOR AMHARIC SENTIMENT ANALYSIS," no. March, 2019.

[2] M. Learn, "https://monkeylearn.com/sentiment-analysis," *[online accessed]*, 2020. .

[3] M. Tadesse, "Trilingual Sentiment Analysis on Social Media," Addis Abeba University, 2018.

[4] M. Farhadloo, "Statistical Methods for Aspect Level Sentiment Analysis," 2015.

[5] D. M. El-Din, "Analyzing Scientific Papers Based on Sentiment Analysis," no. January, p. 121, 2016.

[6] A. Alsaeedi and M. Z. Khan, "A study on sentiment analysis techniques of Twitter data," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 2, pp. 361–374, 2019.

[7] E. Aydogan and M. Ali Akcayol, "A comprehensive survey for sentiment analysis tasks using machine learning techniques," *Proc. 2016 Int. Symp. Innov. Intell. Syst. Appl. INISTA 2016*, 2016.

[8] J. Abate, M. Meshesha, and A. Ababa, "Unsupervised Opinion Mining Approach for Afaan Oromoo Sentiments."

[9] M. Z. Asghar, A. Khan, S. Ahmad, M. Qasim, and A. Khan, "Lexicon-enhanced sentiment analysis framework using rule-based classification scheme," pp. 1–22, 2017.

[10] M. Oljira, "Sentiment Analysis of Afaan Oromoo Social media content using machine learning approaches," Ambo University, 2020.

[11] "https://en.wiktionary.org/wiki/sentiment_analysis." .

[12] "https://en.wikipedia.org/wiki/Sentiment_analysis," *online accessed 9/12/2020*, 2020. .

[13] D. Jurafsky, "Sentiment Analysis," *Stanford*, pp. 1–81, 2017.

[14] D. R. Kawade and D. K. S. Oza, "Sentiment Analysis: Machine Learning Approach," *Int. J. Eng. Technol.*, vol. 9, no. 3, pp. 2183–2186, 2017.

[15] N. Rathee, N. Joshi, and J. Kaur, "Sentiment Analysis Using Machine Learning Techniques on Python," *Proc. 2nd Int. Conf. Intell. Comput. Control Syst. ICICCS 2018*, vol. 7, no. V, pp. 779–785, 2019.

[16] S. J. Sachdeva, R. Abhishek, and D. A. V.K, "Comparing Machine Learning Techniques for Sentiment Analysis," *Ijarcce*, vol. 8, no. 4, pp. 67–71, 2019.

[17] S. Priyanka and M. Sivakumar, “Big Data Processing in Sentiment and Opinion Mining for Detecting Student Depression in E-Learning Using Rich Facebook Dataset Collection,” vol. 5, no. 4, pp. 1208–1216, 2015.

[18] N. Kausikaa and V. Uma, “Sentiment Analysis of English and Tamil Tweets using Path Length Similarity based Word Sense Disambiguation,” vol. 18, no. 3, pp. 82–89, 2016.

[19] G. Beigi, X. Hu, R. Maciejewski, and H. Liu, “An Overview of Sentiment Analysis in Social Media and its Applications in Disaster Relief.”

[20] P. F. Bakken, T. A. Bratlie, C. Marco, and J. A. Gulla, “Political news sentiment analysis for under-resourced languages,” *COLING 2016 - 26th Int. Conf. Comput. Linguist. Proc. COLING 2016 Tech. Pap.*, pp. 2989–2996, 2016.

[21] W. Medhat, A. Hassan, and H. Korashy, “Sentiment analysis algorithms and applications: A survey,” *Ain Shams Eng. J.*, vol. 5, no. 4, pp. 1093–1113, 2014.

[22] D. M. E. D. M. Hussein, “A survey on sentiment analysis challenges,” *J. King Saud Univ. - Eng. Sci.*, vol. 30, no. 4, pp. 330–338, 2018.

[23] S. Kautish and R. Kaur, *Sentiment Analysis- From Theory to Practice*. 2017.

[24] A. Jurek, M. D. Mulvenna, and Y. Bi, “Improved lexicon-based sentiment analysis for social media analytics,” *Secur. Inform.*, vol. 4, no. 1, 2015.

[25] M. Z. Asghar, F. M. Kundi, A. Khan, and S. Ahmad, “Lexicon-Based Sentiment Analysis in the Social Web,” *J. Basic. Appl. Sci. Res*, vol. 4, no. 6, pp. 238–248, 2014.

[26] E. Boiy and M. F. Moens, “A machine learning approach to sentiment analysis in multilingual web texts,” *Inf. Retr. Boston.*, vol. 12, no. 5, pp. 526–558, 2009.

[27] N. Mehra, S. Khandelwal, and P. Patel, “Sentiment Identification Using Maximum Entropy Analysis of Movie Reviews,” p. 7, 2002.

[28] G. Vaitheeswaran, “Combining Lexicon and Machine Learning Method to Enhance the Accuracy of Sentiment Analysis on Big Data,” vol. 7, no. 1, pp. 306–311, 2016.

[29] L. M. Rojas-Barahona, “Deep learning for sentiment analysis,” *Lang. Linguist. Compass*, vol. 10, no. 12, pp. 701–719, 2016.

[30] S. Minaee, E. Azimi, and A. Abdolrashidi, “Deep-Sentiment: Sentiment Analysis Using Ensemble of CNN and Bi-LSTM Models,” 2019.

[31] J. Deriu and M. Cieliebak, “Sentiment analysis using convolutional neural networks with multi-task training and distant supervision on Italian tweets,” *CEUR Workshop Proc.*, vol.

1749, 2016.

[32] A. Mehta, “A Complete Guide to Types of Neural Networks,” *[Online] https://www.digitalvidya.com/blog/types-of-neural-networks/.)*. .

[33] H. Liang, X. Sun, Y. Sun, and Y. Gao, “Text feature extraction based on deep learning: a review,” *Eurasip J. Wirel. Commun. Netw.*, vol. 2017, no. 1, pp. 1–12, 2017.

[34] Q. H. Vo, H. T. Nguyen, B. Le, and M. Le Nguyen, “Multi-channel LSTM-CNN model for Vietnamese sentiment analysis,” *Proc. - 2017 9th Int. Conf. Knowl. Syst. Eng. KSE 2017*, vol. 2017-Janua, pp. 24–29, 2017.

[35] G. Biricik, B. Diri, and A. C. Sönmez, “Abstract feature extraction for text classification,” *Turkish J. Electr. Eng. Comput. Sci.*, vol. 20, no. SUPPL.1, pp. 1137–1159, 2012.

[36] R. P. Nawangsari, R. Kusumaningrum, and A. Wibowo, “Word2vec for Indonesian sentiment analysis towards hotel reviews: An evaluation study,” *Procedia Comput. Sci.*, vol. 157, pp. 360–366, 2019.

[37] R. Ahuja, A. Chug, S. Kohli, S. Gupta, and P. Ahuja, “The impact of features extraction on the sentiment analysis,” *Procedia Comput. Sci.*, vol. 152, pp. 341–348, 2019.

[38] O. B. Deho, W. A. Agangiba, F. L. Aryeh, and J. A. Ansah, “Sentiment analysis with word embedding,” *IEEE Int. Conf. Adapt. Sci. Technol. ICAST*, vol. 2018-Augus, no. March, pp. 1–4, 2018.

[39] B. Das and S. Chakraborty, “An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation,” 2018.

[40] M. Avinash and E. Sivasankar, “A study of feature extraction techniques for sentiment analysis,” *Adv. Intell. Syst. Comput.*, vol. 814, pp. 475–486, 2019.

[41] P. B. Awachate and V. P. Kshirsagar, “Improved Twitter Sentiment Analysis Using N Gram Feature Selection and Combinations,” *Int. J. Adv. Res. Comput. Commun. Eng. ISO*, vol. 3297, no. 9, pp. 154–157, 2007.

[42] D. Gamal, M. Alfonse, E.-S. M. El-Horbaty, and A.-B. M. Salem, “Analysis of Machine Learning Algorithms for Opinion Mining in Different Domains,” *Mach. Learn. Knowl. Extr.*, vol. 1, no. 1, pp. 224–234, 2018.

[43] Z. Wei, D. Miao, J. H. Chauchat, R. Zhao, and W. Li, “N-grams based feature selection and text representation for chinese text classification,” *Int. J. Comput. Intell. Syst.*, vol. 2, no. 4, pp. 365–374, 2009.

[44] M. Felgueiras, F. Batista, and J. P. Carvalho, *Creating Classification Models from Crunchbase*, vol. 2. Springer International Publishing, 2020.

[45] "machine-learning-word-embedding-sentiment-classification-using-keras," *[online] https://towardsdatascience.com/*, 2020. .

[46] "Word embedding," *[online accessed] https://medium.com/@hari4om/word-embedding-d816f643140*, 2020. .

[47] S. Al-Saqqa and A. Awajan, "The Use of Word2vec Model in Sentiment Analysis: A Survey," *ACM Int. Conf. Proceeding Ser.*, pp. 39–43, 2019.

[48] M. T. G/Medhin, "Trilingual Sentiment Analysis on Social Media Mebrahtu Tadesse G / Medhin A Thesis Submitted to the Department of Computer Science in," 2018.

[49] A. Hasan, S. Moin, A. Karim, and S. Shamshirband, "Machine Learning-Based Sentiment Analysis for Twitter Accounts," *Math. Comput. Appl.*, vol. 23, no. 1, p. 11, 2018.

[50] A. Gupta, J. Pruthi, and N. Sahu, "Sentiment Analysis of Tweets using Machine Learning Approach," *Int. J. Comput. Sci. Mob. Comput.*, vol. 6, no. 4, pp. 444–458, 2017.

[51] Wondwossen Mulugeta, "A Machine Learning Approach to Multi-Scale Sentiment Analysis of Amharic Online Posts," pp. 80–87.

[52] W. Tegegne, "The Development of Written Afan Oromo and the Appropriateness of Qubee , Latin Script , for Afan Oromo Writing," *Hist. Res. Lett.*, vol. 28, no. 1976, pp. 8–14, 2016.

[53] B. Ibrahim, "The Origin of Afaan Oromo : Mother Language," vol. 15, no. 12, 2015.

[54] "Social Media Stats Ethiopia | StatCounter Global Stats," *[Online] available (https://gs.statcounter.com/social-media-stats#monthly-201908-202002)*. .

[55] V. Bobicev and M. Sokolova, "Inter-annotator agreement in sentiment analysis: Machine learning perspective," in *International Conference Recent Advances in Natural Language Processing, RANLP*, 2017, vol. 2017-September, pp. 97–102.

[56] M. M. S. Missen *et al.*, "OpinionML-Opinion markup language for sentiment representation," *Symmetry (Basel).*, vol. 11, no. 4, pp. 1–37, 2019.

[57] S. Mohammad, "A Practical Guide to Sentiment Annotation: Challenges and Solutions," pp. 174–179, 2016.

[58] C. Troussas, M. Virvou, K. J. Espinosa, K. Llaguno, and J. Caro, "Sentiment analysis of Facebook statuses using Naive Bayes Classifier for language learning," *IISA 2013 - 4th Int. Conf. Information, Intell. Syst. Appl.*, pp. 198–205, 2013.

[59] T. Mullen and N. Collier, “Sentiment analysis using support vector machines with diverse information sources,” *Natl. Inst. Informatics*, vol. 1, pp. 25-es, 2004.

[60] Https://rpubs.com/masmit11/533879, “Support Vector Machine,” *[online accessed]*, 2020. .

[61] N. Bahrawi, “Sentiment Analysis Using Random Forest Algorithm-Online Social Media Based,” *J. Inf. Technol. Its Util.*, vol. 2, no. 2, p. 29, 2019.

[62] J. H. Wang, T. W. Liu, X. Luo, and L. Wang, “An LSTM approach to short text sentiment classification with word embeddings,” *Proc. 30th Conf. Comput. Linguist. Speech Process. ROCLING 2018*, pp. 214–223, 2018.

[63] S. Liao, J. Wang, R. Yu, K. Sato, and Z. Cheng, “CNN for situations understanding based on sentiment analysis of twitter data,” *Procedia Comput. Sci.*, vol. 111, no. 2015, pp. 376–381, 2017.

[64] J. Y. Antoine, J. Villaneau, and A. Lefeuvre, “Weighted Krippendorff’s alpha is a more reliable metrics for multicoders ordinal annotations: Experimental studies on emotion, opinion and coreference annotation,” *14th Conf. Eur. Chapter Assoc. Comput. Linguist. 2014, EACL 2014*, pp. 550–559, 2014.

[65] S. M. John and K. Kartheeban, “Sentiment Scoring and Performance Metrics Examination of Various Supervised Classifiers,” *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, no. 2S2, pp. 1120–1126, 2019.

[66] D. Tesfaye, “Designing a Stemmer for Afaan Oromo Text : A Hybrid Approach,” ADDIS ABABA UNIVERSITY FACULTY OF INFORMATICS, 2010.

[67] Https://www.betterevaluation.org/en/evaluation-options/wordcloud, “WordCloud,” *online accessed 9/12/2020*, 2020. .

Appendix A: Guide line for the annotation

Define clear rules

A good technique to label text is defining clear rules of what should receive which polarity. Once we provide some list of rules, be consistent. If we classify profanity statement as negative, it is not expected to label the other half of the dataset as positive if they contain inappropriate profanity. But this won't always True. Depending on the problem, even sarcastic statement can be a problem and a sign of negativity. So, the next rule of to be applied for labelling text is to label the easiest examples first. The obvious polarity (positive, negative and neutral) examples should be labelled earlier as soon as possible, and the hardest ones should have labeled to the next the very hard statement has to be labeled last, when you have a better comprehension of the problem.

The emotional states are needed to label the sentiments with the opinions. It is obvious that in Sentiment Analysis different emotional states are needed for different kinds of situations and domains, therefore, their use cannot be forced by defining a fixed set of states.

The persons participated in the labeling or annotations were given a briefing on purpose of annotations along with demonstrations. Each participant was given the following set of annotation guidelines, mostly focusing on cases with possible conflicts.

Before sentiment polarity classification the annotators needs to be identify the following aspects.

- **Polarity identification of the statements:** Annotating or labeling statements are supposed to labeled, based on the statement as positive or negative if it shows a positive (or negative) orientation for a given target. The statement shall labeled as neutral if there is no description that indicate both negative and positive terms.
- **Identification of Sentence Topic:** the annotators should be careful about the selected topic and must choose the nearest chain because sentiments may be shows vary based on their topic and context.
- **Identify Informal Expressions** An idiomatic or sarcastic resource may be experienced, in the idioms or informal expressions present in the data it is a very necessary to be careful because words that found in an sarcastic statement idiomatic expression do not necessarily reflect their actual semantics.

- **Labeling Opinion Targets:** the entity that giving opinion holder of the opinion if exists can be tracked, while the entity which is being opinionated or target of the opinion can be checked in object of the statement, because target determine the polarity of the sentiments.

General guideline

While labeling the sentiments, the annotators should consider the following rule in common as the.

- Positive: the statement can be considered as positive if it shows an explicit or implicit clue in the text suggesting that the text is in a positive state like happy, admiring, relaxed, forgiving, etc.
- Negative: the statement can be considered as negative if it shows an explicit or implicit clue in the text suggesting that the text is in a negative like sad, angry, anxious, violent, etc.
- Neutral: the statement can be considered as negative if it shows there is no explicit or implicit indicator of the speaker's emotional state.

If the statement structure is complex and contains both negative and positive sentiment polarities at the same time the statement can be considered as Positive or Negative based the pattern of Afaan Oromoo grammatical rule.

Sample example for general guidelines:

- Jireenyi magaalaa baayyee gaarii dha

  This statement has **positive** polarity.

- Jireenyi magaalaa baayyee sodaachisaa dha

  This statement has **negative** polarity.

- Uummata baayyetu magaalaa jiraata

  This statement has **Neutral** polarity.

- Vaariyasiin haaraan lubbuuf sodaachisu magaalota birootti babal'achaa jira

  This statement has **negative** polarity.

- Qorannoon gara garaa dhukkuba haaraa vaayirasii koronaa fayyisuun akka danda'amu jedhamuun isaa nama gammachiiseera

This statement has **positive** polarity

- Intalli tokko waan barbaaddu jaalalleen isee itti himnaan iseenis akkuma dhageessen maatii gurbichaatti himte isaatti akka himte ibsiti

  This statement has **Neutral** polarity

Appendix B: Afaan Oromoo stop words

The following lists are some of the Stop words in Afaan Oromoo

kan	**saniif**
sun	**tanaaf**
ani	**tanaafi**
ini	**tanaafuu**
isaan	**waan**
iseen	**itumallee**
isaa	**otumallee**
akka	**ta`ullee**
kan	**tawullee**
ofii	**tahullee**
yoom	**ituullee**
kun	**otuullee**
koo	**ennaa**
kee	**henna**
ammo	**innaa**
garuu	**hoggaa**
yookaan	**oggaa**
yookiin	**hogguu**
akkasumas	**waggaa**
booda	**yoo**
booddee	**hoo**
eegana	**yeroo**
eegasii	**yommuu**
erga	**yammuu**
eega	**yemmuu**
kanaaf	**yommii**
kanaafi	**simmoo**
kanaafuu	**oo**
tanaaf	**woo**

tanaafi
tanaafuu
malee
silaa
yookinimoo
fi
immoo
moo
illee
akka
jechuu
jechuun
jechaan
otuu
otoo
utuu
osoo
odoo
ituu

hoo
akkasumas
akkam
akka
ituu
odoo
silaa
yeroo
hanga
erga

akkum
akkuma
booda
booddee
dura
erga
eega
kanaaf
kanaafi

Appendix C: Sample codes

Sample code for data labeling inter-annotator agreement for annotator one and two

```
from sklearn.metrics import cohen_kappa_score
import pandas as pd
rater1 = pd.read_csv('E:/OBN corpus/annotator1.csv', header=0,encoding = 'unicode_escape')
rater2 = pd.read_csv('E:/OBN corpus/annotator2.csv', header=0,encoding = 'unicode_escape')
#iterate for each dimension

dimensions=rater1.columns

for dim in dimensions:
    dim_codes1 = rater1[dim]
    dim_codes2 = rater2[dim]
    print('Agreement Result:',dim)
    score = cohen_kappa_score(dim_codes1,dim_codes2)
    print(' ',score)
```

Sample Source code for data preprocessing

```
# Data Cleaner and tokenizer
def tokenizeText(text):
    text = text.strip().replace("\n", " ").replace("\r", " ")
    text = text.lower()
    tokens = parser(text)
    lemmas = []
    for tok in tokens:
        lemmas.append(tok.lemma_.lower().strip() if tok.lemma_ != "-PRON-" else tok.lower_)
    tokens = lemmas
    # reomve stop words and special charaters
    tokens = [tok for tok in tokens if tok.lower() not in STOPLIST]
    tokens = [tok for tok in tokens if tok not in SYMBOLS]
    tokens = [tok for tok in tokens if len(tok) >= 3]
    # remove remaining tokens that are not alphabetic
    tokens = [tok for tok in tokens if tok.isalpha()]
    tokens = [tok for tok in tokens if tok != '']
    # stemming of words
       tokens = list(set(tokens))
    #return tokens
    return str(' '.join(tokens[:]))
```

Sample source code for word cloud

```
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
%matplotlib inline
%config InlineBackend.figure_format = 'retina'
from wordcloud import WordCloud, STOPWORDS
neg_sentiment = train[train.polarity == 'negative']
neg_string = []
for t in neg_sentiment.sentiment:
    neg_string.append(t)
neg_string = pd.Series(neg_string).str.cat(sep=' ')
from wordcloud import WordCloud
wordcloud = WordCloud(width=1600, height=800,max_font_size=200, colormap='magma',
                      background_color='white').generate(neg_string)
plt.figure(figsize=(12,10))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```

Sample source code for vectorising or feature extraction

```
vect = TfidfVectorizer(tokenizer=preprocess_and_tokenize, sublinear_tf=True, norm='12', ngram_range=(0, 1))
# fit on our complete Afaan Oromoo Dataset
vect.fit_transform(data.sentiment)

# transform testing and training datasets to vectors
X_train_vect = vect.transform(X_train)
X_test_vect = vect.transform(X_test)
```

Sample source code for classifiers

```
nb = MultinomialNB()
nb.fit(X_train_vect, y_train)
ynb_pred = nb.predict(X_test_vect)
from sklearn.metrics import accuracy_score, f1_score, confusion_matrix, precision_score, recall_score
print("Accuracy: {:.2f}%".format(accuracy_score(y_test, ynb_pred) * 100))
print("\nF1 Score: {:.2f}".format(f1_score(y_test, ynb_pred, average='macro') * 100))
print("\nCOnfusion Matrix:\n", confusion_matrix(y_test, ynb_pred))
print("Recall add by me: {:.2f}%".format(recall_score(y_test, ynb_pred,average='macro')*100))
print("presision add by me: {:.2f}%".format(precision_score(y_test, ynb_pred,average='macro')*100))
# Plot normalized confusion matrix
plot_confusion_matrix(y_test, ynb_pred, classes=class_names, normalize=True, title='Normalized confusion matrix')
plt.show()
```

Sample source code for confusion matrix

```
def plot_confusion_matrix(y_true, y_pred, classes,
                          normalize=False, title=None, cmap=plt.cm.Blues):
    if not title:
        if normalize:
            title = 'Normalized confusion matrix'
        else:
            title = 'Confusion matrix, without normalization'
    cm = confusion_matrix(y_true, y_pred)
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')
    print(cm)
    fig, ax = plt.subplots()
    im = ax.imshow(cm, interpolation='nearest', cmap=cmap)
    ax.figure.colorbar(im, ax=ax)
    ax.set(xticks=np.arange(cm.shape[1]), yticks=np.arange(cm.shape[0]), xticklabels=classes, yticklabels=classes,
           title=title, ylabel='Actual Label',   xlabel='Predicted Label')
    plt.setp(ax.get_xticklabels(), rotation=10, ha="right",     rotation_mode="anchor")
    fmt = '.2f' if normalize else 'd'
    thresh = cm.max() / 2.
    for i in range(cm.shape[0]):
        for j in range(cm.shape[1]):
            ax.text(j, i, format(cm[i, j], fmt),
                    ha="center", va="center",
                    color="white" if cm[i, j] > thresh else "black")
    fig.tight_layout()
    plt.xlim(-0.5, 2.5)
    plt.ylim(2.5, -0.5)
    return ax
```