A MACHINE LEARNING APPROACH TO MULTI-SCALE SENTIMENT ANALYSIS OF TIGRIGNA ONLINE POSTS

M.Sc. THESIS

HAGAZI SAMUEL

SEPTEMBER 2021
HARAMAYA UNIVERSITY, ETHIOPIA

A Machine Learning Approach to Multi-Scale Sentiment Analysis of Tigrigna Online Posts

A Thesis Submitted to the Department of Computer Science,
Post Graduate Program Directorate
HARAMAYA UNIVERSITY

In Partial Fulfillment of the Requirements for the Degree of
MASTER OF SCIENCE IN COMPUTER SCIENCE

Hagazi Samuel

September 2021
Haramaya University, Ethiopia

HARAMAYA UNIVERSITY
POST GRADUATE PROGRAM DIRECTORATE

We here by certify that we have read and evaluated this Thesis entitled **"A Machine Learning Approach to Multi-Scale Sentiment Analysis of Tigrigna Online Posts"** prepared under our guidance by **Hagazi Samuel**. We recommend that it be submitted as fulfilling the thesis requirement

Yaregal Assabie (Ph.D.)		17/06/2021
Major Advisor	Signature	Date

Akubazgi Gebremariam(MSc)		17/06/2021
Co-Advisor	Signature	Date

As a member of the Board of Examiners of the MSc Thesis Open Defense Examination, we certify that we have read and evaluated the Thesis prepared by **Hagazi Samuel** and examined the candidate, I recommend that the thesis be accepted as fulfilling the Thesis requirements for the degree of Master of Science in Computer Science.

Tilahun Shiferaw (Assistant Professor)		
Chairman	Signature	Date

Million Meshesha (Ph.D.)		
Internal Examiner	Signature	Date

Dereje Teferi (Ph.D.)		
External Examiner	Signature	Date

Final approval and acceptance of the Thesis is contingent upon the submission of its final copy to the Council of Post Graduate Directorate (CPGD) through the candidate`s department or school graduate committee (DGC or SGC).

DECLARATION

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all sources of materials for the thesis have been duly acknowledged.

Hagazi Samuel

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

Yaregal Assabie (Ph.D.)

Haramaya, Ethiopia
September 2021

DEDICATION

ድ ኣደኣይይ ድ ዓደይይ

To-my beloved Mother and Country

ACKNOWLEDGMENT

First of all, thanks to the Almighty GOD and St. Virgin Mary for everything. Next, I would like to express my deep gratitude to my research advisor Yaregal Assabie (Ph.D.) for his guidance, constructive comments and suggestions. I would also like to thank Wondwossen Mulugeta (Ph.D.) for his kind willingness to give me insightful comments and suggestions whenever I asked him. I am thankful to my co-advisor Akubazgi Gebremariam (MSc) for his help and inspiration. I am also grateful to my love Yemsrach Tadele (Lwamey), whose love and encouragement was crucial to the completion of the study. Finally, I would like to show respect and gratitude to my family and friends who have supported me in every way.

ABSTRACT

With the rapid growth of web technologies, individuals and organizations are increasingly using public opinions in blogs, forums, review sites, social networks, etc. for expressing their views and opinions. These reviews are very useful for service providers, manufactures and organizations in making informed decisions and improving their service. However, the huge volume of reviews on the social media grows so rapidly and becoming increasingly difficult for users to analyze and extract relevant information. Therefore, an automated sentiment analysis is needed.

In this research, we presented a multiscale sentence-level sentiment analysis for Tigrigna online posts using a supervised machine learning approach. The multiscale Tigrigna sentiment analysis model classifies a given sentence into five predefined classes: very positive (2), positive (1), neutral (0), negative (-1) and very negative (-2). We have used three supervised machine-learning algorithms: Naïve Bayes (NB), Maximum Entropy (MaxEnt) and Support Vector Machine (SVM) with unigram, bigram, trigram and hybrid of unigram and bigram variants of N-gram as a feature. The proposed model contains different components like preprocessing (tokenization, normalization, stop word removal), morphological analysis (lemmatizing), feature extraction, training a machine learning algorithms, classification and evaluation of the result using evaluation metrics.

For conducting the experiments, 1500 Tigrigna sentences are collected from different sources. Due to the morphological complexity of the language, preprocessing techniques have been applied in order to clean noisy data and reduce sparseness and dimensionality of the dataset. After preprocessing, the dataset is lemmatized, before it is given to training phase of the experiment. The experimental results show the SVM algorithm with unigram language model outperforms all algorithms with 71% accuracy. In conclusion, despite the language morphological complexity and lack of effective morphological analysis tools, the achieved experimental results are promising. However, we are convinced that the results could improve further with a larger, pre-annotated and cleaned corpus.

Keywords: Tigrigna Language; N-gram model; Multi-scale Sentiment Analysis; Maximum Entropy; Support Vector Machine; Naive Bayes

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF ACRONYMS AND ABBREVATIONS

CSAE	Central Statistical Agency for Ethiopia
CSV	Comma Separated Values
DSR	Design Science Research
DW	Dmtsi Weyane
FBC	Fana Broadcasting Corporate
MaxEnt	Maximum Entropy
NB	Naive Bayes
NLP	Natural Language Processing
NLTK	Natural Language Tool Kit
SVM	Support Vector Machine
TigTv	Tigrai TV
VOA	Voice of America

CHAPTER 1
INTRODUCTION

1.1 Background

The rapid growth of web technologies such as blogs, forums, and various other types of social networks such as Facebook, Twitter, YouTube, LinkedIn, etc., provides people a medium to find useful information in a factual and opinion forms, to share their views, experiences, ideas, and opinions and influence each other by providing sentiment. The web technologies; which have billions of users over all the globe, provide a platform, to comment in discussion forums and other social medias about other people's posts, newly launched services, products. Individuals and organizations are increasingly using public opinions in blogs, forums, wikis, review sites, social networks, tweets and so on for their decision-making. This has changed the manner in which people communicate and influence social, political and economic behavior of other people and organizations. According to Pang & Lee (2008) ,an opinion is a person's view, judgments, appraisal, ideas and thoughts formed in the mind towards entities, individuals, issues, events, topics, a particular matter and their attributes. Huge amount of text sentiment or opinion is stored on the social media in the form of tweets, status updates, blog posts, comments, Facebook posts, reviews etc. This yields in an enormous amount of unstructured information, which requires systematic and efficient categorization (Pang et al., 2002). Therefore, it is important that doing analysis on that sentiment to extract and identify the opinions of people regarding company, products and people and provide a relevant information for organizations which is crucial in making the right decision.

Sentiment analysis is the multidisciplinary field of study; that deals with identifying, extracting and analyzing people's sentiments, attitudes, emotions and opinions; about different entities such as products, services, individuals, companies, organizations, events and topics. It includes multiple fields such as Natural Language Processing (NLP), Text Analysis, Computational Linguistic (CL) s, Information Retrieval (IR), Machine Learning (ML) and Artificial Intelligence (AI) (Liu, 2012a). It determines the subjectivity whether the expression is subjective or objective and uses automated tools to detect the subjective expressions of opinion holders. 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 others specific topics with their attributes in order to extract subjective information in a given text unlike factual information, opinions and sentiments are subjective (Abdul-mageed et al., 2013).

According to Liu (2012) and Saberi & Saad(2017) sentiment analysis can be done at three levels: document level, sentence level or feature level. The document level sentiment classification classifies the whole opinion expressed by the opinion holder as positive, negative and neutral. In sentence level sentiment analysis, there are two major tasks: the subjectivity classification and sentiment classification. The subjectivity classification deals with classifying every sentence in the given document in to objective sentences which express factual information and subjective sentences which express opinions. The sentiment classification is concerned with polarity classification such as positive, negative or neutral classification. At the feature level, instead of looking at document, sentence or paragraph it directly looks at the opinion itself. In this research, our focus is on sentence level sentiment classification of Tigrigna texts.

There are two main techniques for sentiment classification namely Machine Learning and Lexicon based approach(Liu, 2012). There are three types of machine learning techniques; supervised learning, semi-supervised learning and unsupervised learning. In supervised leaning, a collection of labeled sentiment examples is provided to the model for training to decide the polarity. In contrast to supervised learning, unsupervised is unlabeled and they do not offer with the correct aims at all and therefore depend on clustering. The second sentiment analysis method, lexicon-based approach uses dictionaries of words footnoted with their semantic orientations. It has two techniques dictionary-based approach, and corpus-based approach.

Corpus-based approaches find co-occurrence patterns of words to determine the sentiments of words or phrases (Hatzivassiloglou & McKeown, 1997; Turney, 2001). Dictionary-based approaches use synonyms and antonyms in WordNet to determine word sentiments based on a set of seed opinion words. Due to the increasing of social media users in both Ethiopia and Eritrea, large volume of Tigrigna texts is available in social media, which include Facebook, Twitter and YouTube. As result, this study deals with sentiment analysis to, automatically analyze sentiments written in Tigrigna using supervised machine learning approach.

1.2 Statement of the problem

The growth of Internet technologies and the availability of smart phones provide the benefit for people to actively express their feelings and emotion on, products, news, governmental policy, governmental service and political campaigns issue through social media platforms. Similarly, they can also provide sentiments, feedbacks and feelings on business companies, product companies and public services through social media platforms. People makes commentaries about a certain subject or talks about their personal experiences and invite readers to provide their own comments about products, services, organizations, events etc. Opinions are so important that whenever we need to make a decision, we want to hear other's opinions. This is not only true for individuals but also for organizations. People give comments about the goods and bad features of the products, services on social media networks and different websites. Getting these feedbacks is very helpful and informative for the users, services providers, manufacturers, producers etc.

With the fast-growing development of the web and social media engagement, the number of documents expressing opinions becomes more and more important (Tromp, 2011).As a result, governmental and non-governmental organizations are moving towards using online social media platforms for collecting feedbacks, opinions about their products, services etc.However,due to the rapid growth of opinionated documents, a large number of reviews and posts exists on the web, the need for finding relevant sources, extract related sentences with opinions, summarizing and organizing them to a useful form is becoming difficult for companies or service providers to get the sentiment of the customer easily and at the same time, it also makes it difficult for a potential customer to read them for making an informed decision on whether to purchase the product or not. The feedback collecting methods are tedious, time and resource consuming but important for the organization. Therefore, it is highly significant to have an automatic sentiment analysis tool for efficiently grasping opinions in a short time.

Social media users can write and express their sentiments, feedbacks and feelings by using different languages like Tigrigna, Amharic, Arabic and English on topics posted on social media. Extracting this information and classifying it according to their degree of polarity by using machine-learning approach is the main interest of this research work. There are different sentiment analysis researches conducted for Amharic (Selama Gebremeskel, 2010; Tulu Tilahun, 2013;Abreham Getachew, 2014;Wondwossen

Philemon & Wondwossen Mulugeta, 2014) and English ((Mourad & Darwish, 2013);(Salunkhe & Deshmukh, 2017);Pang & Lee, 2008). Tigrigna is closely related to Amharic and distantly related to Arabic, but there are important morphological structure and language behavior differences. As a result, we cannot apply these stated research studies directly. There is a work done using rule based approach for Tigrigna by Nabyom Shishay(2018) and also another rule based Trilingual sentiment analysis research conducted for Amharic, English and Tigrigna languages (Mebrahtu Tadesse, 2018).

The research works conducted for Tigrigna by (Mebrahtu Tadesse, 2018; Nabyom Shishay, 2018) used manually prepared lexica of Tigrigna sentiment terms to identify and assign polarity values. Even though a rule-based approach is considered as a valuable alternative for underdeveloped and under resourced languages like Tigrigna, systems developed using this approach are not easy to scale-up and unavailability of huge lexical domain like sentiword makes classification using lexical knowledge difficult. Besides, machine learning performs with less human intervention and have the ability to improve overtime(Wondwossen Philemon & Wondwossen Mulugeta, 2014) (Pang et al., 2002).As far as the researcher's knowledge is concerned, there is no multi-scale sentiment analysis research conducted for Tigrigna online texts using a machine learning approach. In this research work, we attempted to design and develop machine learning based multi-scale sentiment analysis of Tigrigna texts, as it is worth of conducting the study taking in to account the rapid growth of Tigrigna texts on social media. The main aim of this research is therefore, to develop a sentence-level sentiment classification model of Tigrigna posts which classifies the online texts into five classes namely very positive, positive, neutral, negative and very negative. Only explicit and regular types of sentiments with n-gram models are used for feature selection and extraction by annotating the training and testing data manually, and this will improve current activities of opinion mining by adapting machine learning based sentiment analysis. At the end of this study, the research aims to answer the following research questions (RQ) from the obtained experimental result.

RQ1. What are the important features that can be extracted from opinionated Tigrigna texts that have the greatest influence on sentiment analysis?

RQ2. Which machine learning techniques perform better classification for Tigrigna sentiment analysis?

RQ3. What is the effect of stemming on the Tigrigna language sentiment analysis?

1.3 Objectives of the study

1.3.1 General Objective

The general objective of this research is to design and develop a multi-scale sentiment analysis model of Tigrigna online texts using a machine learning approach.

1.3.2 Specific Objectives

The specific objectives of the research are to:

- Review available related literature on sentiment analysis, techniques of sentiment analysis.
- Analyze the general structure and morphology of Tigrigna sentences related to identifying statements, which express sentiments.
- Collect the dataset and prepare the needed corpus
- Design the general architecture of sentiment analysis system.
- Preprocess, lemmatize and stemming of the dataset
- Develop the model for Tigrigna sentiment analysis system.
- Train the machine learning algorithms using the training dataset
- Evaluate the effectiveness of the proposed model using precision, recall and F-measure.

1.4 Scope and Limitation of the Study

Scope of the study

Tigrigna multiscale sentiment analysis deals with classifying extracted multiscale sentiments from Tigrigna social media texts (posts, comments, feedbacks) written with Geez script at a sentence level using a supervised machine learning approach by considering only regular and explicit sentiment types into a predefined five polarity classes namely: very positive, positive, neutral, negative and very negative.

Delimitation of the study

Tigrigna multiscale sentiment analysis does not work for feature or aspect level sentiment analysis and does not consider implicit and comparative sentiment types. Idiomatic expressions, sarcasm, slang of words, and sentiments expressed through an image/picture, an audio, video and other emotional symbols are out of scope of this study. It does not also cover sentiment analysis tasks like subjective or objective classification.

1.5 Significance of the Study

The research made efforts to provide the following domain benefits:

- The model can help in decision making on activities like to design new policy and strategies, new product launches, politics, movies, music, etc.
- The model can provide structured and summarized sentiment information about public opinion.
- Business and organizations can use the system to reduce the money spent to find consumer's sentiment and opinions.
- The model can be used to answer opinion questions and reactions towards some events such as the 6th Ethiopian national election on June, 2021.
- The model can be used as an advertisement means when one praises a product using the system.
- Can be used for other researchers' who want to work on Tigrigna sentiment analysis with specific domain(s).

1.6 Organization of the Thesis

This research paper is organized in six chapters. Chapter 1 presents introduction, statement of the problem, objectives, significance, scope and limitation of the study. Chapter 2 discusses an overview of sentiment analysis and the different techniques used in sentiment analysis researches., the general steps in sentiment analysis are discussed. Linguistic behavior of Tigrigna and its challenges in sentiment analysis and related works of the research are also discussed in this chapter. Chapter 3 presents the employed research methodology. Chapter 4 discusses the general architecture and design of multi-scale sentiment analysis model for Tigrigna texts. Chapter 5 shows experiment conducted and performance evaluation of the model with analysis of results. Chapter 6 presents conclusions, contributions and recommendations of the study.

CHAPTER 2
LITERATURE REVIEW

2.1 Overview

In this chapter, numerous works carried out on sentiment analysis is reviewed to; identify the gap and findings of the research. It is also reviewed to have deep understanding about concepts, techniques and methods of sentiment analysis, background literature on general sentiment analysis, types of sentiment analysis, components of sentiment analysis, levels of sentiment analysis, steps of sentiment analysis, classification approach of sentiment analysis metrics and evaluation methods, supervised ,unsupervised and semi supervised machine learning approaches, lexicon based approaches ,Tigrigna language and related works have been reviewed.

2.2 Sentiment Analysis

Textual information in the world can be broadly classified into two main categories, facts and opinions. Facts are objective statements about entities and events in the world in other hand opinions are subjective statements that reflect people's sentiments or perceptions about entities and events that are not necessarily based on a fact. People have lots of different opinions and in many cases people can have differing opinions on the same issue(Tromp, 2011).Sentiment is a belief, thought or judgment prompted by a feeling, which may use to express emotions, opinions, attitudes, views, outlooks, approaches, experiences etc. It may be expressed in a form of text, speech, tweets, news, posts, etc. For example, some people may have the opinion that "life begins at birth and that abortion should be illegal or restricted only to rare situations such as when the life of the mother is in danger". Other people have the opinion that "abortion is a woman's health issue and that women should have the freedom to choose whether to abort a child or not". Berhe can have the opinion that reading is *boring*; while Rahel can have the opinion that reading is *fun*.

The web contains a bulk of opinions about products, politicians, events and more which are expressed in newsgroup posts, review sites, social networking sites etc. Much of the existing research on text information processing has been focused on mining and retrieval of factual information, e.g., information retrieval, Web search, and many other text mining and natural language processing tasks. Little work has been done on the processing of opinions until recently. Yet, opinions are very important whenever one needs to make genuine decision. This is not only true for individuals but also true for

organizations (Tripathy et al., 2015).Unlike factual information, opinions and sentiments have an important characteristic, namely, they are subjective. It is thus important to examine a collection of opinions from many people rather than only a single opinion from one person because such an opinion represents only the subjective view of that single person, which is usually not sufficient for application. Due to a large collection of opinions on the web, an automated sentiment analysis is thus needed(Pang & Lee, 2008).

Sentiment analysis, also called opinion mining, is the field of study in natural language processing (NLP) that analyzes people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes towards some particular real-world entity. It represents a large problem space and widely used in various fields to analyze people's sentiments on various application domains can be movies, products, politics, hotels and others specific topics (Liu, 2012a).

Interpretation of opinions could be challenging for humans, as binary distinction of opinions as only positive or negative may not be sufficient. The scale values show the strength of positivity or negativity of a text as rank. It provides a quick indication of the tone of a text and provides a more refined analysis, which is important for several real life applications such as comparison of several opinions and for giving ranks to different opinions(Pang & Lee, 2005).

Classification of human's opinions is challenging because of the polarity of the people's sentiment. Basic sentiment analysis allows to determine or measure the polarity of sentiment. In other words, it involves classifying opinions in text into categories like positive, negative, or neutral. Sentiment analysis is not simple task as it includes the study of NLP that process grammatical issues in order to identify the opinionated terms in the language(Turney, 2001). The multi-scale property of sentiment indicates the extent to which an intensity of certain emotions such as love, joy, surprise, anger, sadness, fear etc. gives a better insight to sentiment analysis. To handle both factual and emotional evaluations it is significant to apply multi-scale sentiment analysis to the opinions of the sentence(Wondwossen Philemon & Wondwossen Mulugeta, 2014).

2.3 Types of Sentiment

Liu (2012) divided sentiment into two types, regular and comparative sentiments. The regular sentiment is used to express desired or undesired states or situations (e.g.

wonderful (ግሩም/grum), poor (ደኽም/dKum), pleasing (ትሕትና/tHtna), delightful (መሓሽ/meHegosi), etc.) and is often referred to simply as opinions in this research. It includes two main sub-types. These are **direct opinion**, which is expressed directly on an entity or one of its aspects (e.g., "This drug is very good."). In contrast, **indirect opinion** is expressed on an entity based on its effect on some other entities (e.g., "After taking this drug, my blood pressure rises.").

The comparative sentiment is used to express comparative or superlative opinions (e.g. *better, worse, best,* etc.) This states the similarity or difference relationship among two or more entities and /or a preference of the opinion holder based on some mutual aspects of the entities. For example, a typical regular opinion sentence is "*The voice quality of this phone is amazing*," and a typical comparative opinion sentence is "*The voice quality of Samsung is better than that of Techno*." This comparative sentence does not say that any phone's voice quality is good or bad, but simply states a relative ordering in terms of voice quality of the two smart phones. The comparative opinion is usually expressed using the comparative or superlative form of an adjective or adverbs, although not always.

In addition to this classification, sentiment can be classified depending on how they are stated in text, explicit opinion and implicit or implied opinion(Jindal & Liu, 2006).Explicit sentiment is a subjective statement and gives a regular or comparative opinion, e.g. ("Pepsi tastes great," and "Pepsi tastes better than Coke"). As it is shown in the example, explicit sentiment is clearly and unambiguously expressed or stated. In contrary to explicit sentiment, implicit sentiment is an objective statement that infers a regular or comparative sentiment. These often describe a desirable or undesirable fact, e.g., ("The battery life of Techno phones is longer than Tana phones"). An explicit sentiment is easy to identify and classify than implicit one. Therefore, it is important to consider both explicit and implicit sentiment during sentiment analysis tasks.

2.4 Components of Sentiment Analysis

The sentiment analysis consists of three basic components (Liu, 2012b; Wiebe & Mihalcea, 2005): opinion holder, object and opinion. ***Opinion holder*** is a person or an organization that expresses the opinion on a particular object, event, topic or services. It is also called an opinion source. ***Object*** is an entity that can be a product, a person, an event, an organization or a topic, which opinion holders have expressed an opinion on that object. ***An opinion*** is the view, attitude, opinion, sentiment or appraisals on a

particular object from an opinion holder. The opinion expressed in the object can be strong positive, positive, neutral, negative or strong negative sentiments or can be a numeric rating score that expresses the strength or intensity of the sentiments. During sentiment analysis, the (very) positive, (very) negative and neutral are called the semantic orientations or polarities(Abdul-Mageed et al., 2011). Example: Gebre said that the printer is slow. Here Gebre is the opinion holder, printer is the object and slow is the actual opinion or sentiment towards the Printer.

2.5 Sentiment Analysis Levels

As Liu (2012) stated, sentiment analysis can be conducted at different levels of granularity with different levels of detail. It can take place in three levels: Document level sentiment analysis, Sentence level sentiment analysis and Entity or Aspect level sentiment analysis.

2.5.1 Document Level Sentiment Analysis

The task at this level is to classify whether the whole review document expresses a positive or negative sentiment. For example, in a given movie review or product review, the system determines whether the review expresses an overall positive ,neutral or negative polarity about the movie or the product. In some cases, a neutral class is also considered. This level of analysis assumes that each document expresses an opinion on a single entity (e.g., a single movie or single product). Hence, it is not applicable to documents, which evaluate or compare multiple entities (Pang et al., 2002).

2.5.2 Sentence Level Sentiment Analysis

The task at this level is to determine whether each sentence expresses positive, negative or neutral sentiments. In this level, besides sentiment classification, subjectivity classification is another problem. Since, subjectivity classification aims to determine whether the sentence is subjective or objective. A subjective sentence expresses a subjective view and determines either the sentence is positive or negative sentiment, whereas, an objective sentence expresses a factual information from a sentence (Abdul-Mageed et al., 2011).Supported by this observation, the type of granularity that is preferable for social media posts is the sentence level sentiment analysis and this research work deals with the sentence level sentiment analysis because the corpus is composed of social media texts including posts, comments and feedbacks.

2.5.3 Aspect Level Sentiment Analysis

Aspect level directly looks at the opinion itself instead of looking on the language constructs (documents, paragraph, sentences, clause or phrases). It is based on the concept that an opinion consists of a sentiment either positive or negative and a target of opinion (Somprasertsri & Lalitrojwong, 2010). During aspect level sentiment analysis, identifying an opinion target helps to understand the sentiment analysis problem better. For example, “although the service is not great, I still love this hotel”, this sentence contains positive polarity, but we cannot say that the sentence is entirely positive. In fact, the sentence is positive about the hotel (emphasized), but negative to the service (not emphasized). In different applications, opinion targets are represented by entities and/or their different aspects because the aim of this level of sentiment analysis is to discover the sentiments on the specific object entities and/or their aspects. At the end, it provides the structured summary of sentiments about entities and their aspects from unstructured texts(Liu, 2012a).

2.6 Major Subtasks of Sentiment Analysis

Sentiment analysis involves the main subtasks of sentiment identification (identifying the opinionative words or phrases), feature extraction (identifying the features required for the analysis), sentiment classification (classifying the polarity as positive, negative or neutral), and summarization of result (Siqueira & Barros, 2010).Web users use various blogs, forums, social networking sites, and review sites to express their personal feelings, opinions, emotions, and feedback in relation to a particular product, service, individual, organization ,or event. The sentiment analysis process begins with data collection, which involves collecting data from opinion sources related to domains under study.

2.6.1 Sentiment Identification

Once the collected data is preprocessed, sentiment identification is performed. Sentiment identification also known as subjectivity detection is the task of identifying objective and subjective sentences. Objective sentences are those which do not exhibit any sentiment (Chaturvedi et al., 2018). So, it is desired for a sentiment analysis engine to find and separate the objective sentences for further analysis, e.g., polarity detection or sentiment classification.

2.6.2 Feature Extraction

The feature extraction subtask, which follows opinion identification, is a subtask of sentiment classification task that aims to identify the important words in the inputted text document. Examples of text features include part-of-speech (POS), term presence and frequency, negations, and opinion words and phrase (Yadollahi et al., 2017). Feature extraction subtask will reduce the amount of data that needs to be considered in the subsequent sentiment classification subtask. As the extracted features will be fed to the classifier, it is vital to extract the correct features which will contribute to the success of the machine learning technique that will be used in the sentiment classification subtask that will follow.in some researches the features such as term presence, opinion words and phrases (Liu, 2010), positions of words, part-of-speech, syntax and negation are used to limit the classification challenge (Pang et al., 2002). The explanation of the term presence, opinion words and phrases, n-grams, part-of-speech, syntax and negation are given in the following.

Term Frequency and Presence: Describe the text as a vector representing the number of times individual terms have been repeated. Term presence improves the sentiment classification task (Pang & Lee, 2008).In the term based features, document representation emphasizing term presence contain 1 if term appears in the document at least once, 0 otherwise.

Opinion Words and Phrases: These words are regularly used to express positive or negative sentiments. For example, beautiful, wonderful, good, and amazing are positive opinion words, and bad, poor, and terrible are negative opinion words. Although many opinion words are adjectives and adverbs, nouns (e.g., rubbish, junk, and crap) and verbs (e.g., hate and like) can also indicate opinions. Not only single terms, there are also sentiment phrases and idioms, e.g., cost someone an arm and a leg. Opinion words and phrases are instrumental to sentiment analysis for obvious reasons.

Word positions: Term positions are also significant in representing the document for sentiment analysis. The position of terms decides, and sometimes reverses, the polarity of the sentence. So, position information is sometimes encoded into the feature vector(Pang & Lee, 2008).

Part-of-Speech: Adjectives are good pointers of sentiment in text. For example, (Turney, 2001) uses part-of-speech patterns, most containing an adjective or an adverb, for sentiment detection.

Syntax: Syntax information may contain vital text features such as negation, intensifiers, and diminishers, which use for sentiment analysis(Liu, 2012a).

Negations: Negations reverse the polarity of the sentence by coming before the word and sometimes before and after a word.

2.6.3 Sentiment Classification

Sentiment classification is a fundamental task which refers to the task of automatically categorizing or classifying a given opinionated piece of text into predefined classes such as, “positive”, “negative”, “neutral”,” very positive”, and “very negative. It mainly consists of two important tasks, sentiment polarity assignment and sentiment intensity assignment (Abbasi et al., 2011). Sentiment polarity assignment deals with analyzing, whether a text has a positive, negative, or neutral semantic orientation. For example, in the sentence, “Mary is very beautiful!” the word beautiful is adjective which shows the positive sentiment of the given sentence. Sentiment intensity assignment deals with analyzing, whether the positive or negative sentiments are mild or strong. Consider the two sentences “I don’t like you” and “I hate you very much”, both sentences are negative but the second sentence is more intense than the first. Sentiment intensity assignment is more important for multi-scale sentiment analysis as it considers many intensifiers of the language.

Sometimes it is crucial that considering more precise about the level of polarity of the opinion, so instead of just talking about positive, neutral, or negative opinions the multi-polarity levels like very positive, positive, neutral, negative and very negative are looked at, this is generally referred to as fine-grained sentiment analysis (Pang & Lee, 2005). It could be, for example, represented by 5-star rating in a review, e.g.: very positive = 5 stars and very negative = 1 star. Some systems also give different flavors of polarity by recognizing if the positive or negative sentiment is linked with a specific feeling, such as, anger, sadness, or worries (i.e., negative feelings) or happiness, love, or enthusiasm (i.e., positive feelings).

2.6.4 Sentiment Summarization

It is essential for the users when they look at the results to get a general understanding of peoples’ sentiments towards an item/product or a specific feature. The data need to be analyzed, before the results can be summarized and efficiently presented to the indented users in a concise understandable form (Tedmori & Awajan, 2019). The summarization of results subtask involves presenting the user with a summary of the

results of the analysis. This summary can range from a simple list of positive and negative evaluations to more advanced graph based summarization and summary models(Liu, 2010). In the case where the volume of data is large, presenting this data in a graphical manner can be more efficient than presenting it in basic tabular or numerical formats.

Hu & Liu (2004) used aspect-based summary when presenting the summary of opinion, by employing two numbers for each aspect; one shows the number of positive feelings towards this aspect and the other shows the number of negative feelings towards the same aspect. Zhuang et al.(2006) provide a statistical summary showing the sentiment distribution of each aspect along with the corresponding sentences for each aspect and sentiment.

2.7 Common classification Approaches

There are different approaches to the problem of sentiment classification. The most commonly applied techniques for sentiment classification are machine learning, lexicon based, and hybrid approaches. The subsequent sections discuss each approach in detail.

2.7.1 Machine Learning Approach

Machine learning is the sub-field of artificial intelligence that helps a computer to learn without explicitly programmed but learn from a given labeled example data or experience. The machine learning approach combines linguistic features and machine learning algorithms. Machine learning methods are categorized into supervised, unsupervised, and semi supervised methods.

2.7.1.1 Supervised Learning

Supervised methods depend on the availability of large annotated training data. The supervised machine learning methods provides a solution to the classification problem based on training algorithms that involves two steps: learning the model from a corpus of training data and classifying the unseen data based on the trained model(Singh & Shahid Husain, 2014).

Supervised sentiment classification model classifies sentiment sentence based on large labeled sets of data: training set and test set. The datasets are then provided to the model during the training and testing processes to produce a meaningful prediction output. The success of the learning process mainly depends on the selection and extraction of the specific set of features useful for a sentiment detection. Supervised classifier can be implemented with the following steps and components:

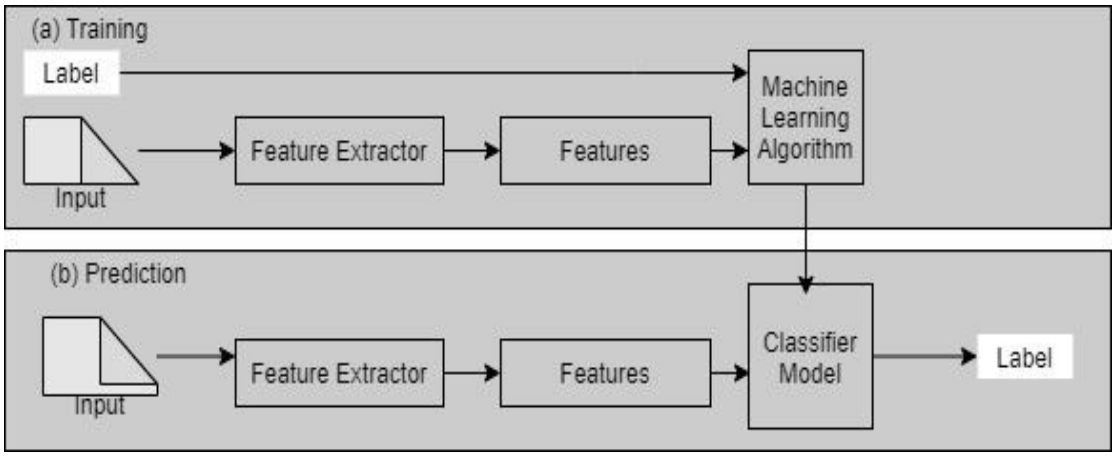

Figure 2-1 Components of supervised learning(Steven et al., 2009)

In the training process (a), the model learns to associate a particular input (i.e., a text) to the corresponding output (tag) based on the test samples used for training. The feature extractor transfers the text input into a feature vector. Pairs of feature vectors and tags (e.g., *very positive*, *positive*, *very negative*, *negative*, or *neutral*) are fed into the machine learning algorithm to generate a model. In the prediction process (b), the feature extractor is used to transform unseen text inputs into feature vectors. These feature vectors are then fed into the model, which generates predicted tags (again, *very positive*, *positive*, *very negative*, *negative*, or *neutral*)(Steven et al., 2009).

The machine learning algorithms like Naïve Bayes, Maximum Entropy and Support Vector Machine (SVM) have achieved very good classification performance in sentiment analysis(Pang et al., 2002). The classifiers are trained on label dataset having samples representing all classes. A test dataset is used to evaluate the performance of the classifiers for the given task. Let the set of sentences as $\{S = s_1,\ldots,s_n\}$, and set of classes labeled as $\{C = c_1,\ldots,c_n\}$, then the task is to classify sentences s_i in S with a label c_i in C. This task can be performed using supervised classifiers. The most commonly used sentiment analysis classifiers are discussed below.

i. Naïve Bayes

Naive Bayes is a simple probabilistic classifier based on Bayes' theorem, which is especially suitable for situations with high input dimensions. Naïve Bayes can learn the pattern of examining a set of documents that has been categorized (Kurt Junshean E., 2013). It compares the contents with the list of words to classify the documents to their right category or class. It assumes an underlying probabilistic model and allows us to capture uncertainty about the model in a principled way by determining the probabilities of the outcomes. Naïve bayes results from applying Bayes Theorem with

independent assumptions between the features and it assumes that the presence (or absence) of a particular class feature is unrelated to the presence (or absence) of any other feature. It is an approach to text classification that assigns the class $c^* = \text{argmax}\ P(c \mid d)$, to a given document d.

$$P(c \mid d) = \frac{P(c)P(d \mid c)}{P(d)} \qquad (2\text{-}1)$$

The *Naive Bayes* (NB) classifier uses the Bayes' rule, Where, $P(d)$ plays no role in selecting c^*. To estimate the term $P(d|c)$, Naive Bayes decomposes it by assuming the fi's are conditionally independent given d's class as in the following Naïve Bayes equation,

$$P_{NB}(c \mid d) = \frac{P(c)\left(\prod_{i=1}^{m} P(f_i \mid c)^{n_i(d)}\right)}{P(d)} \qquad (2\text{-}2)$$

Where, m is the no of features and fi is the feature vector. Consider a training method consisting of a relative-frequency estimation $P(c)$ and $P(fi \mid c)$.

The Naïve Bayes classifier is the simplest and most commonly used classifier. Naïve Bayes classification model computes the posterior probability of a class, based on the distribution of the words in the document. The model works with the n-gram feature extraction which ignores the position of the word in the document. Despite its simplicity, and its conditional independence assumption is obviously not valid in the real world, text classification based on Naive Bayes still tends to work very well. In fact, Naive Bayes is optimal for some problems with highly dependent features. Many researches have shown the Naive Bayes algorithm classify reasonably well.

ii. Maximum Entropy (MaxEnt)

Maximum Entropy (ME) classification is yet another technique, which has proven effective in a number of natural language processing applications.it always tries to maximize the entropy of the system by estimating the conditional distribution of the class label. The conditional distribution defined, as MaxEnt makes no independence assumptions for its features extracted from the given dataset, unlike Naive Bayes. Sometimes, it outperforms Naive Bayes at standard text classification(Berger et al., 1996). The fundamental principle of Maximum Entropy is that the distribution should be uniform. Besides, constraints for the model that characterize the class-specific

expectations for the distribution are derived from labeled training data. When using maximum Entropy, the first step is to identify a set of feature functions, which define a category. Its estimate of $P(c \mid d)$ takes the exponential form as in the, where, $Z(d)$ is a normalization function.

$$P_{ME}(c \mid d) = \frac{1}{Z(d)} \exp\left(\sum_i \lambda_{i,c} F_{i,c}(d,c)\right) \quad (2\text{-}3)$$

In the second step, it determines F_i, c is a *feature/class function* for feature *fi* and class *c*, as in ,

$$F_{i,c}(d,c') = \begin{cases} 1 & n_i(d) > 0 \text{ and } c' = c \\ 0 & \text{otherwise} \end{cases} \quad (2\text{-}4)$$

MaxEnt classifier (known as a conditional exponential classifier) is used to explain the relationship between one nominal dependent variable and one or more independent variables. MaxEnt measures the relationship between the dependent variable (our label/polarity class, what we want to predict) and the one or more independent variables (or features/sentences), by estimating probabilities using its underlying logistic function. These probabilities must then be transformed into binary values in order to actually, make a prediction. This is the task of the logistic function, also called the sigmoid function. The Sigmoid-Function is an S-shaped curve that can take any real-valued number and map it into a value between the range of 0 and 1, but never exactly at those limits. These values between 0 and 1 will then be transformed into either 0 or 1 using a threshold classifier. We want to maximize the likelihood that a random data point gets classified correctly, which is called Maximum Likelihood Estimation. Maximum Likelihood Estimation is a general approach to estimating parameters in statistical models. We can maximize the likelihood of using different methods like an optimization algorithm. Newton‘s Method is such an algorithm and can be used to find maximum (or minimum) of many different functions, including the likelihood function. Instead of Newton‘s Method, we could also use Gradient Descent(de Vries, 2017).

MaxEnt is much more robust to correlated features; compared to Naïve Bayes classifier; if two features f1 and f2 are perfectly correlated, regression will simply assign half the weight to w1 and half to w2. Thus, when there are many correlated features, logistic regression will assign a more accurate probability than naive Bayes. Nonetheless, these less accurate probabilities often result nonetheless in naive Bayes making the correct

classification decision. For example, a specific feature/class function can be triggered if and only if the bigram "still hate" appears and the sentence sentiment is assumed to be negative. Importantly, unlike Naive Bayes, maximum entropy does not make assumptions about the relationship between features, so it may work better when the conditional independence assumption is not met.

iii. Support Vector Machine

Support vector machines (SVMs) have been shown to be highly effective at traditional text categorization, generally outperforming Naive Bayes(Joachims, 1998). They are *large-margin*, rather than probabilistic, classifiers, in contrast to Naive Bayes and Maximum Entropy. In the two-category case, the basic idea behind the training procedure is to find a maximum margin hyperplane, represented by vector$\vec{w}$, that not only separates the document vectors in one class from those in the other, but also for which the separation, or *margin*, is as large as possible. This corresponds to a constrained optimization problem; letting $c_j \in \{1, -1\}$ (corresponding to positive and negative) be the correct class of document *dj*, the solution can be written as the following equation.

$$\vec{w} := \sum_j \alpha_j c_j \vec{d}_j, \quad \alpha_j \geq 0 \qquad (2\text{-}5)$$

Where, the α_j's (Lagrangian multipliers) are obtained by solving a dual optimization problem. Those $\vec{d}_j$ such that α_j is greater than zero are called *support vectors*, since they are the only document vectors contributing to $\vec{w}$ .Classification of test instances consists simply of determining which side of $\vec{w}$'s hyperplane they fall on.

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(Singh & Shahid Husain, 2014).It is a technique for the classification of both liner and non- liner data. Additionally, it is also an algorithm that uses a nonlinear mapping to transform the original training data into a higher dimension. The working principles of SVM are based on the concept of decision planes that defines the decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. An appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyper plane. Support vector machine have various characteristics such

as ability to handle large feature space, ability to prevent of over fitting and information dense in a given data set. Clearly, we can see that there exist multiple lines that offer a solution to the classification problem. But the problem here is that which of lines are better than the others. A line is not good classifier if it passes too close to either of the points because it will be noise sensitive and not accurately classify all set of points. Therefore, our goal should be to finding optimal hyper plane which classify all set of points at optimal margins between. Therefore, the optimal separating hyper plane maximizes the margin (distance from the hyper plane to set points) of the training data as shown in Figure 2-2.

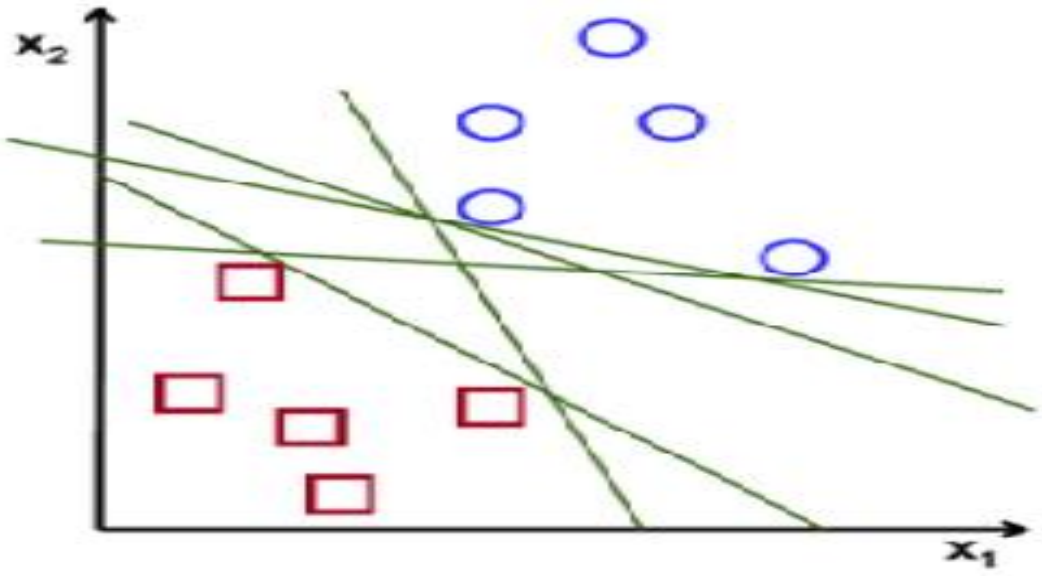

Figure 2-2 SVM algorithm working

(Pang et al., 2002) shows that using unigrams (a bag of individual words) as features in classification performed well with either NB or SVM. The studies also show that SVM is effective, accurate, and can work well with small amount of training data.

2.7.1.2 Unsupervised Learning

Unsupervised learning is a learning method in which a model is not trained with supervised data. The training is provided to the machine with the set of data without labelling, classifying or categorizing and the unsupervised algorithm act on the given data without any supervision. It does not consist of a category and they do not provide with the correct targets at all and therefore rely on clustering. When an annotated training data is difficult to find or not available at all, unsupervised machine learning methods can be an attractive alternative. It takes a text as an input and produces classification output without the need of annotated training data (Rothfels & Tibshirani,

2010). The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns. In unsupervised learning, we do not have predetermined result. The machine tries to find useful insights from the huge amount of data.

2.7.1.3 Semi supervised Learning

Semi supervised method can be used when only a small amount of labeled data is available. It generates an appropriate function or classifier in which both labeled and unlabeled examples are combined(Buche, 2013).

2.7.2 Lexicon Approach

Lexicon based approach uses dictionaries of words footnoted with their semantic orientations and classification is done by matching the features of a given text along sentiment lexicons whose sentiment values are determined prior to their use. It has three techniques manual approach, dictionary-based approach, and corpus-based approach (Taboada et al., 2011). The manual approach opinions are classified depending on the linguistic knowledge and the sentiment summary is calculated manually. In dictionary-based approach, the dictionary of different sentiment words and phrases with their associated orientations, intensification and strength are stored and sentiment score for each word and phrases is computed depending on sentiment word dictionary. On the other hand, corpus based approaches find co-occurrence patterns of words to determine the sentiments of words or phrases in a large corpus (Liu, 2010).

2.7.3 Hybrid Approach

In hybrid approach, lexicon based and machine learning based approaches are combined to benefit from their synergy effect that rises to hybrid approach. Researchers have proved that this combination gives improved performance in sentiment classification (Vaitheeswaran & Dr. L., 2016). In 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.

2.8 Tigrigna Language

Tigrinya (ትግርኛ, also spelled as Tigrinya) is a Semitic language, which belongs to Afro-Asiatic super family, spoken in the Tigrai national regional state of Ethiopia and Eritrea. It is also spoken by a large immigrant community in different countries around the

world such as Sudan, Saudi Arabia, Italy, Sweden, Norway, Germany, the United Kingdom, Canada and the United States. There are around 7 million Tigrigna speakers worldwide. It is an official language and medium of instruction in Tigrai and one of the official languages in Eritrea. According to the 2007 population and housing census of Ethiopia, there are over 4.3 million Tigrigna speakers in Tigray (CSA Ethiopia, 2007)and according to (Ominglot, 2021) there are 3.1 million Tigrigna speakers in Eritrea in 2016. Tigrigna is classified under the Ethio-Semitic sub-phylum and is grouped with Ge‘ez, Tigre, Amharic, Silte, Harari etc. where it shows the characteristic features of a Semitic language(John, 1996).The writing system,morphology,word classes, numbers and punctuation marks of the Tigrigna language are discussed in detail in the following subsections.

2.8.1 Tigrigna Writing System

Tigrigna is written with its own version of the Ge'ez script(also known as “Fidel|ፊደል”) and first appeared in writing during the 13th century in a text on the local laws for the district of Logosarda in southern Eritrea(Ominglot, 2021). The writing system is syllabary; every symbol represents a combination of consonant and vowel. It has 34 base symbols, 6 labialized velars and 7 vowels, which change the basic phoneme of base consonant into seven orders and labialized velars into five orders. Unlike Arabic and Hebrew, Ge'ez script are conveniently written in tabular format of columns from left to right side where the first column represents the base character and the others represent the derived characters that are derived from their vocal sounds(Mebrahtu Tadesse, 2018).The Ethiopic Script includes different letters which have the same sound in a language. The letters 'ሀ' (He) and ‘ኀ’ (He), letters 'ሰ' (se) and 'ሠ' (se), letters 'ጸ' (Tse) and ‘ፀ’ (Tse) are some examples. In Eritrea, the 'ጸ' series is used commonly while in Tigrai the ‘ፀ’ series is used(Osman & Mikami Yoshiki, 2012). Although the alphabets ‘ኀ’ and 'ሠ' are not common on formal and most Tigrigna writings, people use them interchangeably in the informal social media Tigrigna posts. Thus, a single Tigrinya word may exist in two different variations on many texts. For example, መረጻ/mereSa and መረፃ/mere'Sa are two variants of the same for the “election” word.

2.8.2 Tigrigna Word Classes

Word class also known as Parts of Speech (POS) is an appropriate class of word, where each given word of a written text has a unique class.(Daniel Teklu, 2008) classified Tigrigna words into two broad categories: open and closed in general and into five

classes in particular. In open classes new members are always added and are unlimited in number; whereas, members of the closed classes are relatively fixed and few in number. Nouns, Verbs and Adjectives are grouped in the open class category while adverbs and prepositions are grouped in the closed class category.

In Tigrigna, sentiments are expressed mainly using adjectives, nouns, verbs and sometimes although very rare, they are expressed using adverbs. Adverbs however have another important role, intensifying and diminishing a scale of a given sentiment, in expressing sentiments because they have a capability of modifying adjectives, verbs or another adverb. As a result, special attention is given to adjectives, adverbs, nouns and verbs because these word classes play significant roles in affecting the classification process of in Tigrigna sentiment analysis. Since most of the lexical category, words of Tigrigna are important for our work, and then we have described them in detail below.

Noun

Tigrigna noun is a word class used for labeling or referring a real and imaginary thing, such as persons for example, ሓየሎም/Hayelom, places ወጀራት/Wejerat, an idea for example, love), an entity for example, Lawyer, an object for example, Book, a situation for example, ዛሕሊ/zaHli (Cold), etc. Tigrigna nouns, like English nouns, are words used to name or identify a class of things, people, places, ideas, actions and phenomena.

Verb

Verb is a word that tells us the state of doing or being. Tigrigna verbs carry inflections of aspect and mood and hence are morphologically the most complex word class. Many words with other class words are derived primarily from verbs. There are two major approaches to identify verbs from other word categories: syntactical and morphological approach(Teklay Gebregabiher, 2010).

In the former case, verbs function as predicates in a simple sentence and they are found at the end of a sentence because Tigrigna has generally a subject, object, verb (SOV) word order. In the latter case, they reflect grammatical categories such as aspect, mood and agreement. For example, ተስፋይ እንጀራ በሊዑ/tesfay injera beli'U (Tesfay ate Enjera). In this sentence በሊዑ (he ate) is the verb, Tesfay, Enjera are subject and object of the given sentence respectively.

Adjective

Adjectives are words that modifies a noun and pronoun by providing descriptive or specific detail. Unlike adverbs, adjectives do not modify verbs, other adjectives, or adverbs. The numbers of Tigrigna Adjectives are too many and their numbers increase

from time to time. Some words categorized in Adjective are ሓፃር (short feminine), ሓፂር (Short masculine), ነዋሕ/newaH (Tall), ፀሊም/'Selim (Black), ቀያሕ/qeyaH (Red), ሰሓባይ/seHabay (Interesting*)*, *ፅቡቕ*/'SbuQ(Good*)*, *ቀዳማይ/qedamay (*First), ድኹም/dKum (Poor*)*, etc. Adjectives are words that describe or add extra information to a noun. Adjectives in Tigrigna usually precede the nouns that they modify or describe. For example, ሓፃር ጓል/Ha'Sar gWal (short girl), here the word ሓፃር/Ha'Sar(short) describes the noun ጓል/ gWal(girl). However, this does not mean that a word is an adjective just because it precedes a noun. For instance, in the sentence" እታ ጓል ቆንጆ እያ"/ita gWal qonjo eya (The girl is beautiful.), the word እታ/ita(the) precedes the noun ጓል/gWal(girl). Nevertheless, the word እታ/ita (the) is a demonstrative pronoun.

Adverb

An adverb is a word that modifies a verb, adjective, sentence or clauses and other adverbs. Modifiers of verbs or verb phrases usually express time, place, manner etc. Modifiers of adjectives and adverbs commonly express degree while adverbs functioning as sentence modifiers usually express the speakers" attitude regarding the event spoken(Teklay Gebregabiher, 2010).for example,ተወልደ ትማሊ ናብ ኣክሱም ከይዱ። /tewelde tmali nab aksum keydu(Tewelde went to Aksum Yesterday),here ትማሊ/tmali (yesterday) is adverb of time.

Preposition

Prepositions are small set of words, which have meanings only when they are attached with other words such as nouns, verbs, pronouns and adjectives. They can express relationship between person, thing, or event etc. and another. For example, ኣብ ውሽጢ ገዛ /ab wxTi geza (inside the house) and ሰናይ ምስ ኣዕርኽቱ/senay ms a'rKtu (Senay with his friends), ኣብ/ab(inside) and ምስ/ms (with) are the prepositions.

2.8.3 Tigrigna Morphology

Morphology is the branch of linguistics that deals with the internal structure of words and word formation, including affixation behavior, roots, and pattern properties(Daniel Teklu, 2008).Morpheme is the smallest unit of morphology that cannot be broken down further into meaningful parts. Morphemes in Tigrigna can be categorized as free and bound morphemes. Free morphemes are morphemes that can stand on their own to give meaning; whereas, bound morphemes are morphemes that cannot stand on their own to give as a word. For example, the Tigrigna word ከተማታት/ketematat (Cities) contains two morphemes i.e., ከተማ/ketema and ታት/tat. From this, the first word ከተማ/ketema(city) is free morpheme and the second word ታት/tat is bound morpheme

(Girma Berhe, 2006). A basic computational task of morphological language analysis is to use the process of derivation and inflection to deduce the root and grammatical attributes of the word according to the internal structure of the word (Yonas Fisseha, 2011).

Derivation deals with adding of morphemes in the stem word to generate new words, which changes the meaning, word class and lexical function of the given stem words.

Inflection is a morphological variation that never changes word class or the meaning of the stem word upon which the morphemes are attached but mark distinctions of case, gender, number, tense, person, mood, voice and comparison.

Tigrigna affix is a morpheme that is attached at the end, beginning or middle (inside) of the root to create the inflectional or derivational morphology of Tigrigna words. These words may be new in meaning and structure from their respective roots (Hailay Beyene, 2013).Tigrigna affixes can be classified in to four categories as prefixes that come at the beginning of the root, such as ን/n,ዝ/z,እንተ/inte,ም/m,ብም/bm... suffixes that come at the end of the root, such as ና/na,ታት/tat,ት/t,ነት/net,ን/n...infixes that come inside the root, such as ባ in ሰባበረ/sebabere,ላ in በላልዐ/belal'A,ታ in ሰታተየ/setateye ... and circumfixes such as ኣይ....ን in ኣይቀተለን/ayketelen that are attached before and after the base form at the same time.

Morphological analysis of a language plays vital role in several natural language applications such as sentiment analysis, text generation, machine translation, document retrieval, etc. Due to the reason that prepositions are morphologically unproductive and adverbs are few in number and less productive, the derivational and inflectional morphology of the language concentrates on the rest three-word classes (nouns, verbs and adjectives). For the purpose of investigating the derivation and inflection of Tigrigna language nouns ,verbs and adjectives, we have used Tigrigna grammar books by (Daniel Teklu, 2008) and (Amanuel Sahle, 1998).

2.8.3.1 Derivational Morphology

Noun Derivation

Tigrigna nouns are derived from other word classes by adding affixes and using compound words. In the case of compound words, the new noun is constructed from two separate words with and without a morpheme to bind. Example ቤት/bEt and ፅሕፈት/'SHfet gives ቤት ፅሕፈት/bEt 'SHfet(office), which means office, ቤት + አ + መንግስቲ, gives ቤተ መንግስቲ/bEtemengsti(palace) which means palace. Nouns can be constructed

from other nouns by adding affixes like -ነት/net and -ኛ/Na. Example, ምስክር+ነት gives ምስክርነት/mskrnet(testimony), ዓረብ+ኛ generates ዓረብኛ/'arebNa (Arabic). In addition, nouns can be derived from adjectives, roots, stems and verbs. For example, ሰብኣዊ/sebawi (adjective) + ነት gives ሰብኣዊነት/sebawinet a noun, which means humanity.

Adjective Derivation

Tigrigna adjectives can also be primary or derived, even though the number of primary adjectives is very small. Adjectives are derived from nouns, verbs and adjectives itself. Adding morphemes like -ኣዊ/awi, -ዊ/wi -ኣም/am, -ኣይ/ay, -ታይ/tay, -ኣኛ/ANato nouns such as ፍትሒ/ftHi(justice), ኣፍሪካ/afrika (Africa), ሰንኪ/ senki (Cause), ማእኸል/maiKel (Center), ነቕሰገ/ neQsege (name of a place), ተንኮል/ tenkol(trickery) that generates the following new adjectives ፍትሓዊ/ftHawi(just), ኣፍሪካዊ/afrikawi (African), ሰንካም/senkam, ማእኸላይ/maiKelay(central), ነቕሰገታይ/neQsegetay (person (Masculine) from neQsege) and ተንኮለኛ/ tenkoleNa(crafty) respectively. Similarly, it can be constructed from the root verbs. For instance, the root verb ንፅል/n'Sl(single) by infixing the vowel -ኣ/a-, it generates the adjective ንዓል/n'Sal(singled). Adjectives can also be constructed by compounding words such as እግሪ/igri (noun) +ሊሎ/lilo (noun) gives እግሪ ሊሎ(adjective) which means fast and ርጉፅ/rgu'S(adjective) + ከብዲ/kebdi(noun) generates ርጉፅ ከብዲ/rgu'S kebdi(adjective) which means liar.

Verb Derivation

Unlike nouns and adjectives, Tigrigna verbs can be derived only from verbal roots and stems. Example, ብ-ድ-ል, ብኣድኣል which provides በደል/bedel(offence) and it can also be constructed from verbal stems by adding affixes like ተ and ኣ to the stem verb. Almost all Tigrigna verbs are derived from root consonants. Traditionally a distinction is made between simple and derived verbs. Simple verbs are those verbs derived from roots by intercalating vowel patterns whereas derived verbs are considered as derivatives of simple verbs. The derivation process can be internal or external, and this is found in the form of agentive, causative, passive, repetitive and reciprocal verbs.

Causative verbs are derived by adding the derivational morpheme ኣ- to the verb stem as in the examples በፅሐ/be'SHe(arrive)-ኣበፅሐ/ab'SHe (cause to arrive), ቀተለ/qetele(kill)-ኣቕተለ/aqtele (cause to kill) and ወሰደ/wesede(take)-ኣውሰደ/awsede (cause to take). In most cases the 'ኣ- morpheme is used to form causative of intransitive verbs, transitive ones and verbs of state. Some exceptions are the verbs that begin with 'ኣ' always take the morpheme 'ኣ' but add the morpheme" እ" after the morpheme 'ኣ' to

form causative e.g., ኣሰረ/asere (arrest), ኣእሰረ/aisere (cause to arrest) and ኣገደ/agede(prohibit), ኣኣገደ/aAgede (cause to prohibit).

Passive verbs are derived using the derivational morpheme ተ. This derivational morpheme is realized as ተ before consonants and as ት before vowels. Moreover, in the imperfect, jussive and in derived nominal like verbal noun, the derivational morpheme ት-is used. In this case, it assimilates to the first consonant of the verb stem, and as a result, the first radical of the verb geminates. Some exceptions are intransitive verbs such as ፈሊሑ/feliHu (it boiled) that form their passive forms using the prefix ተ-as in ተፈሊሑ/tefeliHu (it was boiled). Such kind of verbs can derive their passive from their causative form ኣፍሊሑ/afliHu (he boiled).

Repetitive verbs indicate an action that is performed repeatedly. For tri-radical verbs, such verbs are formed by duplicating the second consonant of the root and using the ኣ- after the duplicated consonant as in ቀዳደደ/qedadede (he tears repeatedly), derived from the root ቅድድ. All verb types have the same reduplicative/ repetitive forms.

Reciprocal verbs are derived by prefixing the derivational morpheme ተ either to the derived form (that use the vowel a after the first radical) or to the reduplicative stem. For example, reciprocal forms of ተቃተሉ /teqatelu (killed each other) and ተቀታተሉ /teqetatelu (killed one another) are derived from the derived stem -ቃተሉ and reduplicative stem -ቀታተሉ respectively.

2.8.3.2 Inflectional Morphology

Inflection is a morphological variation that does not change the word class category and general meaning, but the grammatical function. Since Tigrigna, language is a highly inflectional language, given root of a Tigrigna word; it can be found in different forms.

Noun Inflection

Tigrigna nouns inflects for gender, person and number by adding affixes to the noun stem. Tigrigna nouns are either male or female by nature. Therefore, in order nouns to express possession, pluralism, nationality and gender, affixes such as -ኣ/a, -ታት/tat, -ኣት/at, -ኣን/an, - ኦት/Ot, -ውቲ/wti, -ቲ/ti, -ዊ/wi, -ና/na, etc. are used. Here are some examples to show the inflection of nouns.

Noun stem	Affix	Noun after affixation
ላሕሚ/laHmi(cow)	ኣ/a	ኣላሕም/alaHm(cows)
እምባ/imba(mountain)	ታት/tat	እምባታት/imbatat(mountains)
ሃገር/hager(country)	ኣት/at	ሃገራት/hagerat(countries)
መምህር/memhr(teacher)	ኣን/an	መምህራን/memhran(teachers)
ገዛ/geza(house)	ውቲ/wti	ገዛውቲ/gezawti(houses)
ከልቢ/kelbi(dog)	ና/na	ከልብና/kelbna(dogs)

Table 2-1 Noun Inflection

Tigrigna has two grammatical genders: masculine and feminine, and all nouns belong to either one or the other and inanimate objects does not have fixed gender.

Verb Inflection

Tigrigna verbs also found in different forms, such as perfective, imperfective, gerundive, jussive and imperative by employing affixes. Tigrigna verbs inflect for number, gender, person, tense, mood and aspects, and the result is an inflected word with affixes to the verb stem. Tigrigna verbs express two tenses, namely, perfect and imperfect. Perfect tense indicates completed actions whereas imperfect tense expresses uncompleted actions.

The morphological variation of the perfective verbs is generated by adding suffixes like ኧ/A, ና/na, ኡ/U, ኦም/Om, ኧን/An, ኩም/kum, ኣት/At, ኪ/ki, and ካ that indicates inflection for person, gender and numbers to the perfect verb stem. The following table illustrates inflection of perfective verbs.

Verb variation	Person	Number	Gender
ሰሚዐ/semi’A	First	Singular	masculine or feminine
ሰሚዕና/semi’na	First	Plural	masculine or feminine
ሰሚዐ/semi’U	Third	Singular	Masculine
ሰሚያም/semi’Om	Third	Plural	Masculine
ሰሚዐን/semi’An	Third	Plural	Feminine
ሰሚዕኩም/semi’kum	Second	Plural	Masculine
ሰሚዕኪ/semi’ki	Second	Singular	Feminine
ሰሚዕካ/semi’ka	Second	Singular	Masculine

Table 2-2 Inflection of perfective verbs

Imperfective verbs also formed by adding affixes on the verb stem and markers for gender, person and number. In imperative tense prefixes እ/i, ት/t, ይ/y, ን/n and the suffixes attached are ኡ/U, ኢ/I, and ኣ/a. To indicate negative verbs the morphemes ኣይ/ay, ኣይት/ayt, ኣይን/ayn are added as prefixes and ን/n, ኣን/an, ኡን/Un are added as suffixes.

Person	Singular	Plural	Gender
First	እሰምዕ/isem’	ንሰምዕ/nsem’	masculine or feminine
Second	ትሰምዕ/tsem’	ትሰምዑ/tsem’U	Masculine
Second	ትሰምዒ/tsem’I	ትሰምዓ/tsem’a	Feminine
Third	ይሰምዕ/ysem’	ይሰምዑ/ysem’U	Masculine
Third	ትሰምዕ/tsem’	ይሰምዓ/ysem’a	Feminine

Table 2-3 Inflection of imperfective verb

The gerundive form is inflected by adding suffixes at the end of the gerundive verb to indicate person, gender and number. For example, ፅሒፈ/’Shife(I wrote), ፅሕፍኻ/’SHfka (you wrote), ፅሕፍኺ/’SHfki(you wrote),ፅሕፉ/’SHfu(he wrote),ፅሒፋ/’SHifa(she wrote),ፅሕፍና/’SHfna(we wrote),ፅሕፍኹም/’SHfKum(you wrote),ፅሕፍኽን/’SHfKn(you wrote),ፅሕፎም/’SHfom(they wrote),ፅሕፈን/’SHfen(they wrote) can show how the gerundive verb ፅሕፍ/’SHf morphologically varies. The suffixes ኣ/A, ኻ/Ka, ኺ/Ki,ኡ/U, ና/na, ኹም/Kum, ኽን/Kn, ም/m and ን/n attached with the inflected stem verb.

Jussive and imperative verbs are sometimes called mood and jussive verbs are used to express a command for first and third persons whereas imperative verb is used to express second person in the singular and plural form. Jussive verbs sometimes called mood verbs are used to express a command for first and third persons, whereas imperative verbs used to express second person in singular and plural forms(Yonas Fisseha, 2011).

Adjective Inflection

Tigrigna adjectives are marked for number, gender, and degree and the result is the inflected word with affixes to the given adjective. Tigrigna adjectives can have singular masculine, singular feminine and the same adjective to express both masculine and feminine genders in the plural form. Adjectives can be inflected by adding affixes like -ኦ/O, -ት/t, -ቲ/ti, -ኣት/at, ዝ/z-, etc. The following table shows the inflected Tigrigna adjective words for number and gender.

Adjective stem	Affix	Adjective after affixation
ሸቃሊ/xeqali	ኦ/O	ሸቃሎ/xeqalo
መሃዚ/mehazi	ቲ/ti	መሃዝቲ/mehazti
ብሉፅ/blu'S	ኣት/at	ብሉፃት/blu'Sat
ሸረታይ/xretay	ኦት/Ot	ሸረቶት/xretot

Table 2-4 Adjective inflection

Adjectives are also inflected for degree. According to (Daniel Teklu, 2008) ,Tigrigna adjectives shows three degrees; positive degree, comparative degree and superlative degree. The following table shows example of adjective inflection for gender, number and degree.

No	Masculine	Feminine	Plural	Degree	
				Comparative	Superlative
1	ሓፂር/Ha'Sir	ሓፃር/Ha'Sar	ሓፀርቲ/Ha'Serti	ይሓፅር/yHa'Sar	ዝሓፀረ/zHa'Sere
2	ድኹም/dKum	ድኹምቲ/dKumti	ድኹማት/dKumat	ይደክም/ydekm	ዝደኸመ/zdeKeme
3	ቀዪሕ/qeyiH	ቀያሕ/qeyaH	ቀያሕቲ/qeyaHti	ይቀይሕ(ykeyH	ዝቀይሐ/zqeyehe

Table 2-5 Adjective inflection for degree

2.8.4 Tigrigna Punctuation Marks and Numbers

Punctuation marks are symbols useful to know word demarcation and to organize and clarify the meaning of writing in natural language processing. Tigrigna has a set of punctuation symbols, some of them are listed in the following table with their names and purposes.

Symbol	Name	Symbol and Name		
፡	ክልተ ነጥቢ		Space	Word separator (not in a modern use)
፤	ድርብ ሰረዝ	;	Semi colon	Sentence connector
፣	ነፀላ ሰረዝ	,	Comma	List separator
።	ኣርባዕተ ነጥቢ	.	Full stop	End of sentence
!	ትእምርተ ኣንክሮ	!	Exclamation Mark	End of an emphatic declaration
፧	ሕቶ ምልክት	?	Question mark	Question mark (not in a modern use)
፨				Paragraph separator(not in a modern use)

Table 2-6 Tigrigna Punctuation

Tigrigna uses the traditional set of numerals used in Ge'ez texts as shown in the following table. These numerals have been replaced by the Arabic numerals, that are, the same ones used in English because they are not suitable for arithmetic computation as they lack representation for zero and decimal points. Nevertheless, we may get them in different writings of Tigrigna such as calendar purposes.

፩	፪	፫	፬	፭	፮	፯	፰	፱	፲	፳	፴	፵	፶	፷	፸	፹	፺	፼
1	2	3	4	5	6	7	8	9	10	20	30	40	50	60	70	80	90	100	10,000

Table 2-7 Ge'ez Numbers

The numerals can be classified as ordinal numbers and cardinal numbers (John, 1996). The cardinal numbers are numbers like ሓደ /Hade(one), ክልተ/klte (two), ሰለስተ/seleste (three), ዓሰርተ/'aserte (ten), etc. and the corresponding ordinal numbers are ቀዳማይ/qedamay(first), ካልኣይ/kalay (second), ሳልሳይ/salsay(third), ዓስራይ/'asray(tenth), etc. There are also special numerals in Tigrigna that correspond to the English like ፍርቂ/frqi(half), ርብዒ/rb'I(quarter), ሲሶ/siso (one third) etc. The removal of punctuation marks increases the effectiveness and efficiency of natural language processing systems as morphological analyzer and stop word removal does (Mulubrhan Hailegebreal, 2017). They also employ the same punctuation markers with slight differences in the usage of some of the markers.

2.9 Challenges of Tigrigna sentiment analysis

(Assefa Gebrehiwot, 2011; Atalay Leul, 2014) notes that there are a number of challenges in Tigrigna language for text processing and classification tasks. The computational task of automatically performing the sentiment analysis task is thus faced with many challenges such as grammatical nuances, morphological complexity, implied meaning from facial expressions and body language, misspellings, ambiguity, and regional or cultural variations in the language. Generally, the challenges of the Tigrigna in sentiment analysis are discussed in the following subsections.

2.9.1 Redundancy of characters

In Tigrigna language, text could be written in a different way to represent the same sound. For instance, letters "ሀ" and "ኀ"; "ጸ" and "ፀ"; "ሰ" and "ሠ" have similar sounds. The use of various forms of characters for the same sound has a problem in the process of feature preparation for the classifier learning. As a result, the characters "ኀ", "ጸ" and "ሠ" are replaced by "ሀ", "ፀ" and "ሰ" characters respectively in the normalization phase of the preprocessing task. For example, the word peace can be written in both

ሰላም/selam and ሠላም/selam, however they represent the given word similarly. There is no clear guideline that tells where to use each character.

2.9.2 Spelling variation of the same word

Spelling variations of a word would unnecessarily increase the number of words representing a sentence(s) which could reduce the efficiency and accuracy of the classifiers. A word in Tigrigna could be translated by different persons using different spelling variation due to regional and dialectical variations. For instance, when "radio" word is translated into Tigrigna, it may be written as ረድዮ/redyo, ራድዮ/radyo or ሬድዮ/riedyo. All these words are used to mean the word "radio" in Tigrigna. The translation of English and other language borrowed words into Tigrigna words increases ambiguity and inconsistency in the classification of Tigrigna texts. During the pre-processing stage of this work, the different forms of a character that have the same sound are changed to one common form in order to normalize word variants caused by inconsistent usage of borrowed words or dialectical differences.

2.9.3 Abbreviation

The abbreviations of Tigrigna words follow different formats because, there is no clear rule in abbreviating the given words. Sometime full stop '.' is used to abbreviate, while other time '/' symbol is used to abbreviate. The abbreviated words can be written without separators. For example, ዶክተር/Doctor can be written as ዶ.ር and ዶ/ር. The inconsistency in the abbreviation creates inconsistency in text classification in general and sentiment analysis in particular.

2.9.4 Under-Resourced language

Tigrigna is one of the least researched and under resourced language in terms of text processing tools and electronic resources. The unavailability of labeled language resources such as annotated corpora, stop word list, lexicon terms and other domain dependent labeled language resources for Tigrigna sentiment analysis and other NLP researches makes it difficult and challenging for the Tigrigna sentiment analysis researches.

2.9.5 Morphological Complexity

Languages that are under Semitic family are rich in morphology. As Tigrigna is one of the Semitic family, a single word has many variant forms. Tigrigna is a highly inflected language and has a complex morphology. It exhibits the root and pattern morphological system. For instance, the word ሰምዐ can be inflected to እሰምዕ/isem', ትሰምዕ/tsem',

ትሰምዒ/tsem’I,ይሰምዕ/ysem’,ንሰምዕ/nsem’,ትሰምዐ/tsem’U, ትሰምዓ/tsem’a,ይሰምዐ/ysem’U, ይሰምዓ/ysem’a and other forms. Morphological analysis is an important phase in sentiment analysis. Its main purpose is to decompose words into morphemes and to associate each morpheme with a morphological information such as stem, root, part of speech, and affix. Tigrigna is a morphologically complex language and this complexity requires the development of appropriate systems that are able to deal with tokenization, spell checking, stemming, lemmatization, pattern matching, and part-of-speech tagging. There is only one freely available morphological analysis tool, HornMorpho by (Gasser, 2011) for Tigrigna as far as our knowledge is concerned. The performance of sentiment analysis systems is highly influenced by this phenomenon. However, the system suffers from significant limitations for example it only considers noun and verb in POS tagging. Applications like information retrieval, text classifications, sentiment analysis could benefit more by the existence and availability of basic and effective tools like stemmer, lemmatizer, and POS taggers(Shoukry & Rafea, 2012; Zitouni, 2014).

2.10 Related Works

In this section, related sentiment analysis researches conducted for Tigrigna, Amharic, Arabic and English languages using different approaches are discussed. In addition to the approaches, the language used, source of the data, procedures, experimental results, performance, and challenges of different sentiment analysis researches are considered.

2.10.1 Sentiment Analysis for Amharic

Selama Gebremeskel (2010) followed a term counting approach to the analysis of sentiments of opined Amharic texts. The researcher has established a model that counts the number of words of sentiment in the sentence and assigns polarity weight if the term is included in the lexicon of sentiment. The sentiment lexicon that includes sentiment terms and contextual valence shifter terms is prepared for checking whether a given word appears in the prepared lexicon as a result of which either sentiment word with assigned polarity will be considered taking into account the effect of context valance shifters such as denial terms value or non-sentiment word. Although the terms of sentiment prepared are not adequate, the results obtained using film and newspaper reviews are encouraging along with the test data.

Tulu Tilahun (2013) deals with Amharic blog's opinion mining model for Amharic texts using a feature-level handmade rules and lexicons. The proposed model consists of five main components: Text Operator, Morphological Analyzer, Feature Extractor, Opinion

Extractor and Summarization of Feature Opinion. The author conducted two experiments to evaluate the determination of features for extraction and opinion words using 484 manually collected reviews for experimental activities from Hotel, University and Hospital. As a result, the first experiment shows an average precision of 95.2% and recall of 26.1% were achieved in the features extraction and an average precision of 78.1% and recall of 66.8% were achieved in the determination of opinion words. The second experiment in features extraction's precision gets lower by 15.4% whereas the precision of opinion words determination gets higher by 1.9% and the recall of both features extraction and opinion words determination gets higher by 7.8% and 25.9% respectively when compared to the first experiment. The research is done on feature level to determine a sentiment of the review sentences; however, it only assumes adjectives.

The research conducted by Wondwossen Philemon & Wondwossen Mulugeta (2014) explored a machine learning approach for multi-scale sentiment analysis of Amharic texts. The research objective was to determine sentiment sentence based on the polarity weight value of the sentiment words. To achieve this, the authors proposed a model that includes components for pre-processing, training a classifier, and classifying a given input post taking into account the language's morphological complexity. The authors also developed their own corpus by collecting around 600 posts from online sources, which were then pre-processed to clean up the data, converting the transliteration into the native Ethiopian script, and changing words into their basic form. The corpus was manually annotated by providing scale values of polarity and frequency of sentiment and using Naive Bayes machine learning algorithm and using variants of unigram, bigram and hybrid as features. The experiment showed that for the intensified positive and negative polarity classes the bigram's accuracy is better performed. The authors, however, used a small size corpus in a low-resourced language.

Abreham Getachew (2014) examined opinion mining as a task of text classification, using two basic feature sets (all unigrams and the review's most informative bag of words).Information Gain feature selection method used to calculate most of the document's informative words and three supervised classifiers from the Natural Language Toolkit (Naïve Bayes, Decision Tree and Maximum Entropy classifiers).Opinion Mining process involves categorizing into predefined categories such as positive and negative or binary classification on opinionated text document. An Opinion Mining model is built in this research work to classify Amharic opinionated

text into positive and negative. The experiments are carried out using 616 Amharic-opinioned texts collected from the sites of Ethiopia Broadcasting Corporation, diretube.com and habesha.com. The experiment shows that all algorithms (Naïve Bayes, Decision Tree, and Maximum Entropy) perform the best selection methods for knowledge gain. NB with 90.9 % accuracy outperforms Decision Tree with 83.1 % and Maximum Entropy with 89.6 %, based on their relative classification performance Even though corpus size is small; the outcome that has been reached is inspiring.

2.10.2 Sentiment Analysis for Arabic

Abdul-mageed et al. (2013) proposed an approach to machine learning to conduct a classification of subjectivity and sentiment at the sentence level. The first work dealt with a dataset of extracted and manually annotated newswire documents from PATB (Arabic Tree Bank) (1281 objective and 1574 subjective). The authors collected and annotated 11,918 sentences from different types of social media services classification is conducted in two stages. In the first stage, a distinction is made between a subjective and an objective text. (i.e., Subjectivity Classification). In the second stage, a distinction is made between a positive and negative sentiment (i.e., Sentiment Classification). Abdul-Mageed et al. used SVM in this work as a learning algorithm along with language-specific and general features. N-grams, domain, unique and polarity lexicon features are linguistic-independent features. To investigate the effect of morphological data on results, Arabic specific features have been added. The findings indicated that using POS tagging and lemmas or lexemes to extract the base forms of words has a positive impact on subjectivity and sentiment classification.

Mountassir et al. (2012) uses three classifiers for binary sentiment classification: NB, SVM and KNN. The first one was developed by these researchers and consists of two domain-specific data sets (movies and sports). The second is OCA, a corpus of other researcher's film reviews. The authors performed a pre-processing task prior to the classification phase by removing stop words, separating words from their base form, eliminating terms used in the dataset only once or twice, and replacing words with their stems. The authors found that pre-processing, combination of n-grams and weighting based on presence improve the performance of classification.

Shoukry & Rafea (2012) examined sentence level Arabic sentiment analysis. The researchers conducted sentiment analysis from various domains on 4000 tweets. They found that 1000 of these tweets were relevant and held opinion without sarcasm. They used two human annotators and found that there were 500 positive reviews and 500

negative reviews. They preprocessed the text by removing user-names, pictures, hashtags, URLs, and non-Arabic words. The authors used unigrams and bigrams as features. Pertaining to the machine learning method, SVM and NB were chosen. They conducted two experiments: one with stop words and another with no stop words. The authors found that eliminating the stop words resulted in very little performance improvement, which means that removing stop words adds little value to the sentiment on the text. SVM performed better than NB by around 4–6% accuracy, with a rate of 72% for unigrams. As for the features, using bigrams did not enhance the results of the unigram model.

2.10.3 Sentiment Analysis for English

Espinosa et al. (2013) conducted the research with the aim of developing a system that can classify an opinion using the classification of sentence level, whether it includes positive, negative or neutral opinions. The study used the Naïve Bayes classification algorithm to classify Facebook posts' status updates into positive and negative. Next, as learning data sets, positive status updates and negative status updates were collected. Then Facebook status updates are classified as positive or negative by using the Naïve Bayes classifier. The drawback this study is that the data collected is a random sample of Facebook status streaming and was not collected using direct queries. On top of that, the data were marked to test the classifier's output variance.

(Lalji & Deshmukh, 2016) carried out with an intention of discovering lexicon-based approach with machine learning to perform sentiment analysis on data collected from Twitter. The research includes data acquisition phase, preprocessing phase, feature extraction, polarity detection, and sentiment classification phase. Data was collected by searching for that match keywords of the word 'car' and collected 28000 tweets as per the request from the researchers using the Twitter API. Preprocessing was done to enhance the quality of the data by removing the noise such as duplicate tweets, retweets, punctuations, numbers, HTML links etc. from the collected data. Tweets that on only contain adjectives were extracted with the help of Tree Tagger part of speech tagging in the feature extraction phase and these tweets were classified as subjective and objective tweets. From the total collected dataset 25000 tweets were classified as subjective tweets and used as an input in the polarity detection phase.in the polarity detection phase, the tweets are classified in to positive, negative and neutral based on the occurrence of words of the tweets in the lexicon dictionary and these classified tweets were considered as a training data in the sentiment classification phase in order

to train the classifier. Support vector machine is used as for the training with unigram, bigram, and trigram features and achieved accuracy of 63.23%,62.23% and 59.98% respectively.

(Pang & Lee, 2002) used machine learning approach for classifying movie reviews data into positive or negative sentiment. The author's approach includes text preparation, text preprocessing, feature selection and sentiment classification steps with the help of three machine learning techniques: NB, SVM and MaxEnt. For experimental purpose 1301 positive-sentiment and 752 negative-sentiment total of 2053 documents corpus were used and then divided this data into three equal-sized folds, maintaining balanced class distribution in each fold. Features such as N-grams (unigrams and bigrams), POS, feature frequency vs. presence, subjectivity (adjectives) and position of word are used to extract the important pattern for classifying data into their sentiment. The experimental result of this research indicates that accuracy of the sentiment classification using the SVM algorithm with unigram feature achieved 82.9% accuracy, although the accuracy differences between those three algorithms are not very large.

2.10.4 Sentiment Analysis for Tigrigna

Mebrahtu Tadesse (2018) developed three-language, Amharic, English and Tigrigna trilingual sentiment analysis on social media using a lexicon approach to classify the given text into seven categories namely, strong positive, strong negative, weak positive, weak negative, positive, negative and neutral. The trilingual sentiment analysis consists of seven main components: preprocessor, language identifier, morphological analyzer, sentence builder using root words, sentiment word detector, polarity weight determiner and sentiment classifier. To evaluate and test functionality of the developed prototype, 564 sentiment sentences collected from Facebook and YouTube, resulting in an average accuracy of 87.49 %, 84.78 % recall and 85.99 % F-measure. Although the result achieved is inspiring, the research takes on small amounts of data and uses a lexicon approach that is not easy to scale up.

Nabyom Shishay (2018) proposed the design of sentiment analysis system for opinionated Tigrigna texts with the use of manually built rules, subjectivity lexicon, and the selection of opinionated language texts. The design proposed contains six main components. These are pre-processing texts, word detection of sentiment, propagation of polarity and weight assignment, total and average weight calculation, strength of classification texts and the developed Tigrigna language subjectivity lexicon. The

developed model detects a review's subjectivity words from the established lexicon and assigns an initial polarity weight to determine the polarity category of the opinionated Tigrigna texts for each detected sentiment terms. To evaluate the proposed model, the author used three domains for the prototype and seven classification classes (Strong positive, positive, weak positive, neutral, weak negative, negative and strong negative). The author achieved 0.816, 0.779 and 0.795 for football domain, 0.869, 0.829 and 0.845 for road-map education police domain and finally 0.762, 0.813 and 0.776 for the Idol show domain on the three respective data sets in terms of precision, recall and f-measure respectively. The result obtained with these test data is very encouraging and promising.

The related works for Tigrigna used the lexicon approach with a small amount of data, but our research uses the machine leaning approach for a Tigrigna multiscale sentiment analysis. The problem with lexicon approach is they are not easy to scale-up and unavailability of huge lexical domain like sentiword makes classification using lexical knowledge difficult. However, the machine learning approach is scalable, have a high speed of learning and also have the ability to improve overtime(Wondwossen Philemon & Wondwossen Mulugeta, 2014) (Pang et al., 2002). Therefore, in this study the machine-learning model that can predict a given Tigrigna text into a five predefined polarity classes (very positive, positive, neutral, negative and very negative). Summary of some of the related works in terms of their language, methods used, data size, result found and limitation of the research works are summarized in Table 2-8 .

Author	Language	Dataset	Method	Accuracy (Research Findings)	Limitation
(Selama Gebremeskel, 2010)	Amharic	303	Lexicon	-	The prepared opinion terms are not sufficient
(Tulu Tilahun, 2013)	Amharic	484	Lexicon	-	-Used adjectives only as sentiment words - The sentiment words are not sufficient
(Wondwossen Philemon & Wondwossen Mulugeta, 2014)	Amharic	608	NB	43.6%,44.3,39.5% for unigram, bigram and hybrid respectively	Developed model has performance limitation due to a smaller number of training data
(Abreham, 2014)	Amharic	616	NB, DT & MaxEnt	-81.8%, NB (BOW) & 90.9%, NB(IG) -74%, DT(BOW)&83.1%, DT(IG) 80.5%, MaxEnt (BOW)&89.6%, MaxEnt (IG)	-Used only unigram as a feature for classification. -Small amount of dataset
(Abdul-mageed et al., 2013)	Arabic	11,918	SVM	-95% (F-Measure)	- Do not use morphological features - Feature selection technique is not used
(Mountassir et al., 2012)	Arabic	1079	NB, SVM & KNN	SVM performed better than NB & KNN	Could include more features

(Shoukry& Rafea, 2012)	Arabic	1000	SVM & NB	SVM achieved better than NB	Could use more algorithms
(Espinosa et al., 2013)	English	7000	NB	72% (F-Score)	Dataset used is not preprocessed
(Lalji & Deshmukh, 2016)	English	28000	Hybrid (Lexicon + ML)	- 63.23%,62.23% and 59.98% for unigram, bigram and trigram respectively.	Considers only adjectives in sentiment type detection
(Pang & Lee, 2002)	English	2053	NB, MaxEnt & SVM	SVM achieved better than NB and MaxEnt	Dataset used for the polarity classes is unbalanced
(Mebrahtu Tadesse, 2018)	Trilingual (Amharic, English and Tigrigna	564	Lexicon approach for multi-scale	87.49%	Small amount of lexicon terms
(Nabyom Shishay, 2018)	Tigrigna	487	Lexicon approach for multi-scale	-	Small amount of lexicon terms

Table 2-8 Related Works Summary

CHAPTER 3
RESEARCH METHODOLOGY

3.1 Overview

In this chapter, the selected research methodology to undergo the research, the designed process model and its components are discussed. Section 3.2 discuss the research methodology used and why it is selected. Section 3.3 discusses how each component of design science research methodology is applied in this research work.

3.2 Research Design

Selection of the right research methodology is a crucial step in conducting research and must be based on the statement of the problem. The starting point of any scientific study is identifying the reason for conducting the research study. In order to accomplish the research aims and objectives, help to collect, analyze and interpret data, provide valid and reliable results, a research methodology that employs methods and techniques, which are the best fit for the research, is selected. As a result, a research paradigm that can produce the intended solution to the problem is employed. This research aims to design and develop model, multi-scale sentiment analysis for Tigrigna texts using a machine learning approach, which classifies sentiments into predefined classes. The research encompasses designing a new artifact to solve observed problems, evaluating the artifact and presenting the results. Therefore, design science research methodology is a perfect fit for this research.

The Design Science Approach proposed by Peffers et al. (2007), is adopted in doing this research which is a process model consisting of six (6) activities in a nominal sequence. Thus, the cyclic process of the DSR suggested by Peffers et al.(2007) is followed along with Hevner et al. (2004) ‘s framework guidance. Figure 3-1 DSR Process Model **(Peffers et al., 2007)** below shows the DSR process model Peffers et al.(2007), which consists of the following processes: problem identification and motivation, defining objective of a solution, design and development, demonstration, evaluation, and communication. Throughout this study, DSRM process is used to create an artifact to efficiently address multi scale sentiment classification of Tigrigna texts, determine objectives for a solution, design and develop an artifact that can be provide in a solution, demonstrate the use of the artifact, evaluate the artifact and to finally communicate the process to others.

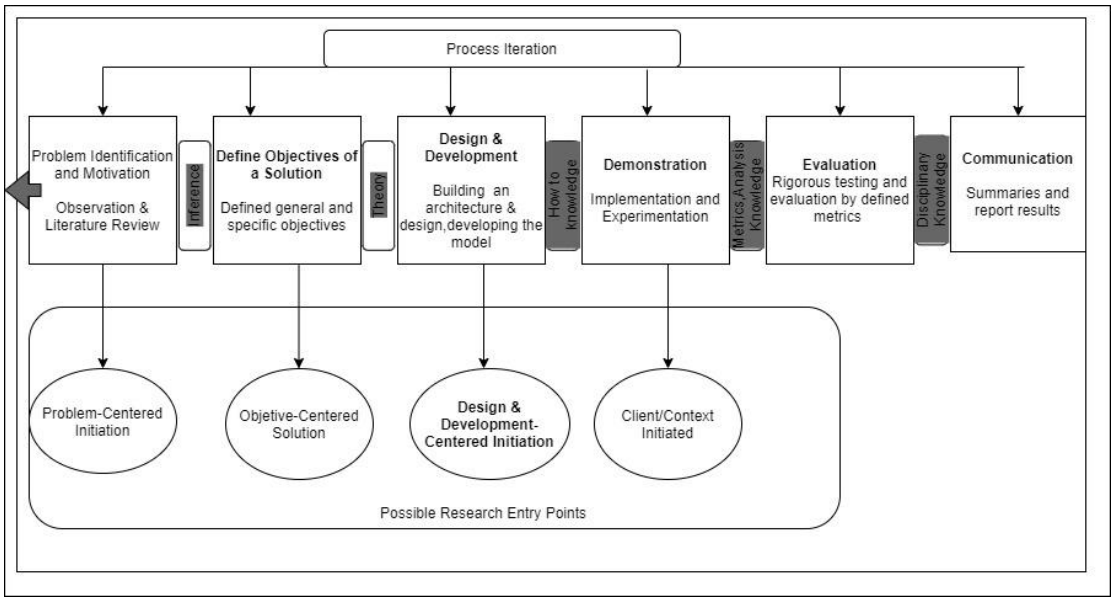

Figure 3-1 DSR Process Model **(Peffers et al., 2007)**

3.2.1 Problem Identification and Motivation

Problem identification is the first step of the research approach. At this stage, the research problem is identified and the importance and motivation of the research is justified (Peffers et al. 2007) with the help of the researcher's prior observation and further investigation through literature review. Literature review is conducted to get a deeper understanding of the area and to get detail knowledge on the various techniques of sentiment analysis, polarity classifications and approaches used in sentiment analysis and to identify improvement points. In addition, to understand more about the Tigrigna language and its morphological behavior, books written by linguistic experts are reviewed. Problem identification and motivation is presented in section 1.2 of the thesis.

3.2.2 Definitions of Objectives

The second step of the DSRM is concerned with the definition of expected outcomes of the research objectives. From the problem identified in the previous step and the solutions that were attempted by previous scholars, objectives of the solution are inferred from the given problem definition. The main objective is designing and developing a model that classifies accurately a given Tigrigna sentence in to one of the five predefined polarity classes.

3.2.3 Design and Development

The design and development phase consist of model designing and its development. These activities are conducted based on the architecture of the proposed system presented in section 4.2. This step involves in artifact design in order to classify

Tigrigna multi scale sentimental texts using a supervised machine learning approach. The main task in this stage is therefore, developing a design and architecture of Tigrigna multiscale sentiment analyzer model. The components of the artifact include tokenizer, normalizer, stop word remover, morphological analyzer, feature extractor, sentiment classifier. Details of each component and techniques used for their development are discussed in CHAPTER 4.

At this stage, the model is implemented using python programming, because python is simple and powerful programming language with excellent features and extensive support libraries to process texts in natural language processing applications. In order to conduct an experiment for this research work, we have used different tools and packages such as Numpy, Pandas, NLTK, Scikit Learn, HornMorpho. NumPy is a library for python that solve scientific computation easily. Pandas is an open-source library that is used to read CSV files and perform different operations on the CSV files. NLTK is a package in python used for many tasks like tokenization, normalization, stop word removal lemmatization, stemming and POS tagging.

Scikit-learn is a library for python machine learning library, which contains simple and efficient tools for data mining and data analysis algorithms for both supervised and unsupervised problems. HornMorpho, part of the L3 project at Indiana University, is freely available python program used for analyzing Afaan Oromo, Amharic and Tigrigna words into their meaningful parts. We have used HornMorpho for both lemmatization and stemming of the dataset. We have also used Notepad++ used for storing CSV file format corpus in UTF-8 encoding and Microsoft word 2019 for writing report of the experimental results.

3.2.4 Demonstration

Demonstration is a process of validating to what extent the artifact to solves the given problem. This may involve its use in experimentation, simulation, a case study, proof, or other appropriate activity. In this phase, in order to demonstrate the performance of the system and its acceptance by the end users, the research uses different test cases in the actual process. Demonstrating how the designed artifact solves problem of the domain area using a selected optimal model is the main task we have in this phase and its detail is presented in section 5.5.

3.2.5 Evaluation

The Evaluation phase consists analyzing and evaluating the results of the Tigrigna multiscale sentiment analyzer model. This involves dataset collection, annotation, testing and evaluation. In order to test performance of the system, a corpus comprise of 1500 sentences manually selected from Facebook, YouTube Tigrigna, BBC Tigrigna, Fana Tigrigna, SBS Tigrigna were prepared. The collected sentences are from politics, education, entertainment and sport domains and the grammar of each sentence is manually checked with the help of linguistic experts. The sentences are also manually annotated as one of the five predefined classes: very positive, positive, neutral, negative and very negative. The model is then tested and evaluated against the testing dataset that contain the predefined classes. The results found from the experimentation phase of the proposed system are evaluated by using a classifier evaluation metrics called confusion metrics shown in Table 3-3-1. The most commonly used evaluation metrics in sentiment analysis are accuracy, precision, recall and F-score(Abdul-mageed et al., 2013).The corpus preparation and evaluation results are presented in detail in section 5.2 and section 5.4 respectively.

In this study, five predefined polarity classes: very positive, positive, neutral, negative and very negative, are employed. Precision, recall and f-measure is calculated for each class but accuracy is calculated for the classifier as a whole. The evaluation result of each metrics for each language model used in the study in the corresponding learning models is presented in CHAPTER CHAPTER 5.

#	Predicted positives	Predicted negatives
Actual positives	True Positives(TP)	False Negatives(FN)
Actual negatives	False Positives(FP)	True Negatives(TN)

Table 3-3-1 Confusion matrix

Where:

True Positives (TP) – predicted as positives they were actual positives

False Negatives (FN) – predicted as negatives they were actual positives

False Positives (FP) – predicted as positives they were actual negatives

True Negatives (TN) – predicted as negatives they were actual negatives

Accuracy measures the ratio of number of correctly predicted reviews to the number of total number of reviews present in the corpus. The accuracy is determined using the equation:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{…….……. (3-1)}$$

Precision measures the ratio of number of correctly predicted positive reviews to the total number of reviews predicted as positive. Out of all the positive classes, we have predicted correctly, how many are actually positive. It is calculated as:

$$\text{Precision} = \frac{TP}{TP+FP} \quad \text{…… (3-2)}$$

Recall measures the ratio of number of correctly predicted positive reviews to the actual number of positive reviews present in the corpus. Out of all the positive classes, how much we predicted correctly. It should be high as possible. It is calculated as:

$$\text{Recall} = \frac{TP}{TP+FN} \quad \text{…… …(3-3)}$$

F1-Measure is a harmonic average of precision and recall. The more precision and recall the better F1 measure. Fl-measure can have best value as 1 and worst value as 0. It is calculated as:

$$\text{Fl}-\text{Measure} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision}+\text{Recall}} \quad \text{…….…… (3-4)}$$

3.2.6 Communication

The final stage of the DSR process, communication, involves summarizing the test results, presenting conclusion and recommendation from the experimentation results, the initial description of this study is presented in the thesis work for scholarly sessions. The result of this research work will be submitted to the department of computer science as partial fulfilment of MSc. degree in Computer Science. Further, the research can also be published in a conference or journal.

CHAPTER 4
SYSTEM ARCHITECTURE AND DESIGN

4.1 Overview

In this chapter, the overall design of our proposed Tigrigna multi-scale sentiment analysis model, the main components and their interaction is discussed in detail. First, we illustrated the general overview of the proposed system architecture. Next, we described how data preprocessing is performed, how supervised machine learning algorithms classify sentence-level sentiment, components along with their sub-components and discussed how each component and its subcomponent is implemented.

4.2 System Architecture

The proposed architecture has four major components: Pre-processor, Morphological analyzer, Feature Extractor, and Sentiment Classifier. Each component and subcomponent are discussed in the following sections. The preprocessor component contains three subcomponents: Normalization, Tokenization and Stop word removal. The Morphological Analyzer contains lemmatization subcomponent. The system architecture is done by adopting the architectures of previously conducted research works on the area by combining and adding new and existing components .generally the architecture is designed after a detail investigation and understanding of the works by (Mebrahtu Tadesse, 2018; Nabyom Shishay, 2018; Tulu Tilahun, 2013; Wondwossen Philemon & Wondwossen Mulugeta, 2014). The general architecture of Tigrigna multi-scale sentiment analysis model is shown in the Figure 4-1.

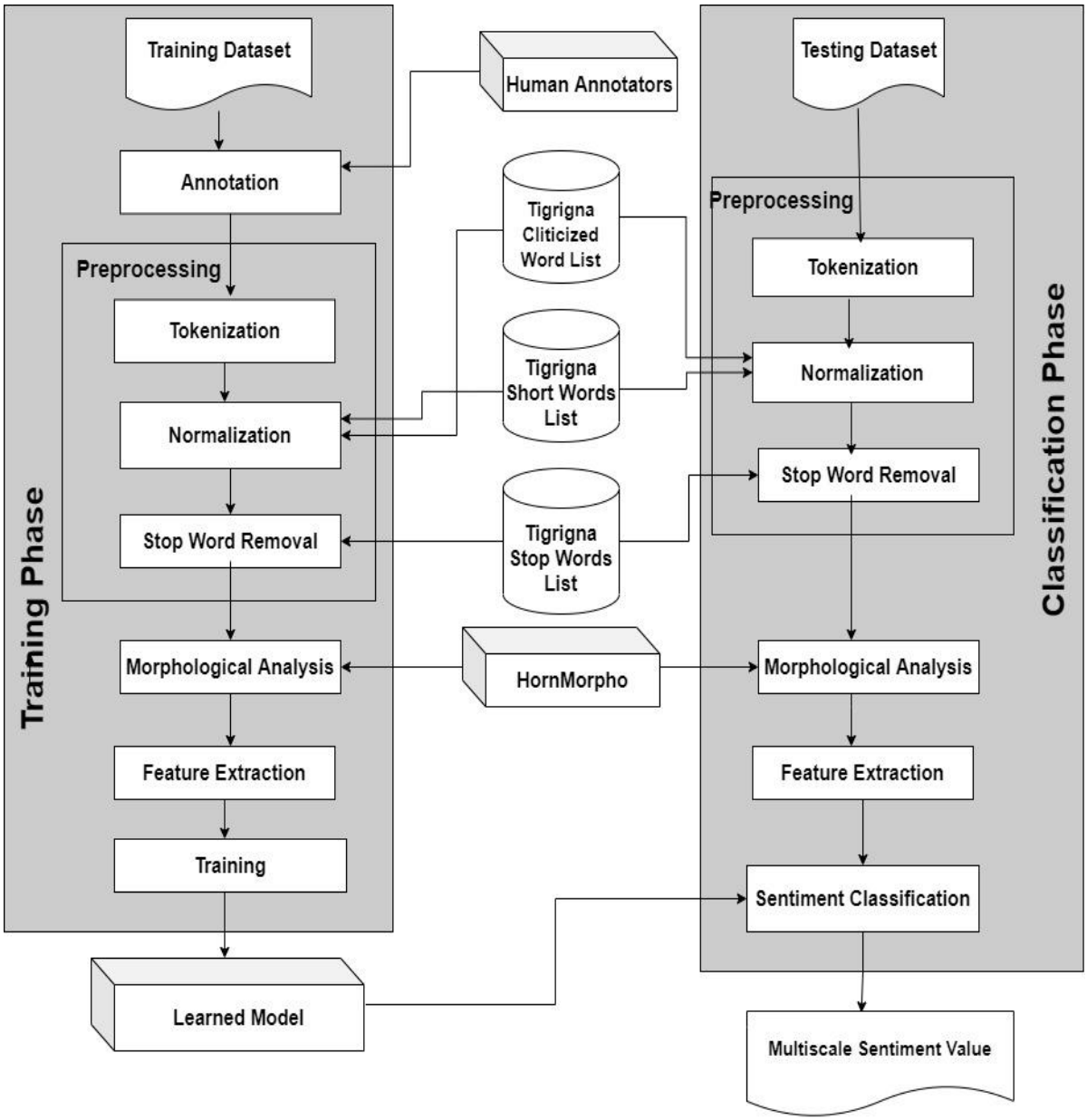

Figure 4-1 Proposed System Architecture

4.3 Annotation

Annotation is the process of labeling a given sentence into one of a predefined polarity classes for sole purpose of training the machine learning algorithm(s) either manually or automatically. Supervised machine learning classifiers are built based on training corpora containing correctly labeled examples for each input. In this work, multi-scale sentiment annotation of sentences is attempted and the corpus is annotated manually with the help of two native Tigrigna speaker human annotators according to their sentiment polarity (very positive, positive, neutral, negative or very negative). These sentiment polarity scales are represented by 2 and 1 for very positive and positive respectively, -2 and -1 for very negative and negative respectively, and 0 for the neutral polarity class.

4.4 Preprocessing

Preprocessing is the process of cleaning noises of the raw data, gathered from different sources, to make it easy and suitable format for the learning task in the sentiment classification based on the selected machine learning method. Preprocessing is done to improve the performance of machine learning sentiment classifiers because social media data or web data is usually inconsistent, incomplete, and lacking certain behaviors, and is likely to contain many grammatical mistakes, which is not feasible for the analysis. The preprocessing component comprises activities such as tokenization, normalization, and stop word removal. These activities are explained in detail in the following sub-sections.

4.4.1 Tokenization

Tokenization is the process of breaking a stream of text up into separate meaningful elements called tokens. The list of tokens was used as an input in next preprocessing subcomponent. Spaces and a number of Tigrigna punctuation marks such as ፡, ፤, and ፧ are used to identify words. For example, the sentence እቲ ጸወታ ቅርሕንቲ ስለ ዝነበሮ መሰልቸዊ ነይሩ።/iti Seweta qrHnti sle znebero meselcewi neyru is tokenized into [እቲ/iti, ጸወታ/Seweta, ቅርሕንቲ/qrHnti, ስለ/sle, ዝነበሮ/znebero, መሰልቸዊ/meselcewi, ነይሩ/neyru, ።]. The output of this tokenization component, tokens of words, are used an input to the normalization subcomponent component in preprocessing. All punctuation marks, control characters, numbers and special characters are removed from the text before the corpus is tokenized. The Tigrigna punctuation marks list is shown in APPENDIX C: List of Tigrigna Punctuation Marks.

```
Input: Original Corpus
Output: Punctuation Marks Removed Corpus
BEGIN
load the Tigrigna punctuation marks list
open the corpus
while not end of the file
    for every character in the corpus
        if the character is in the punctuation marks list
            remove the character
        end if
    end for
end while
close the file
END
```

Algorithm 4-1 Punctuation marks removal

```
Input: Punctuation Marks removed Corpus
Output: Tokenized Corpus
BEGIN
open the corpus
while not end of the file
    while read a sentence from the corpus
        split the sentence with space delimiter
    end while
end while
close file
END
```

Algorithm 4-2 Tokenization

4.4.2 Normalization

Normalization is the process of cleaning or removing unstructured and irrelevant data from a huge collection of extracted textual data. The extracted data is full of noise containing URLs, symbols, tags, links etc. The normalizing process puts the text in a consistent form, thus converting all the various forms of a word to a common form. In Tigrigna, there are different homophonic characters that have the same sound but

written in different forms like ጸ or ፀ as in as ጸሓይ/SeHay(Sun) and ፀሓይ/'SeHay(Sun) that should be changed to the same form for a data uniformity and consistency. Therefore, ሠ is replaced with ሰ, ጸ with ፀ, and although it is not that common the use of ኀ in Tigrigna texts, it is replaced with ሀ. Example pseudo-code is described in Algorithm 4-3.

```
Input: Tokenized Corpus
Output: Homophone Characters Normalized Corpus
BEGIN
open the tokenized corpus
while not end of the file
        for every character in the corpus
            if the character is ኀ or any of its order
                    replace it with ሀ or the respective order
            if the character is ጸ or any of its order
                    replace it with ፀ or the respective order
            if the character is ሠ or any of its order
                    replace it with ሰ or the respective order
            end if
         end for
end while
close file
END
```

Algorithm 4-3 Homophone Characters Normalization

Cliticized words (words joined by an apostrophe) are also separated into their constituent parts in order to normalize the corpus. For example, ንሱ'ዉን/nsu ' wun is a cliticized form of the two words ንሱ/nsu and እዉን/iwun. This tendency occurs because it is customary to mask laryngeals such as እ, ኣ or ኢ with an apostrophe while writing. Correction of spelling and grammatical errors and removing irrelevant contents such as Tweeter hashtags and URLs are removed from the extracted manually as they have no use in sentiment analysis. Example pseudo-code is described in Algorithm 4-4.

```
Input: Homophone Characters Normalized Corpus
Output: Cliticized Words Normalized Corpus
BEGIN
load the cliticized words list
open the tokenized corpus
while not end of the file
      for every word in the corpus
            if the word contains apostrophe
                replace cliticized word with its expanded form
            end if
      end for
end while
close the file
END
```

Algorithm 4-4 Cliticized words normalization

Short forms of characters that are usually written using forward slash "/") and period (".") are also common in Tigrigna texts, for example, ቤት ፅሕፈት/bEt 'SHfet(office) can be written as ቤ/ፅሕፈት, መምህር/memhr(teacher) as መ/ር and ዓመተ ምሕረት/'amete mHret as ዓ.ም. In addition, normalizing words with labialized Tigrigna characters such as ግዋል/gwal to ጓል/gWal. A list of the short word and their expanded forms is prepared manually and the list is show in APPENDIX B: Short Words and their Expanded form List. It basically replaces with its expanded form if it exists in the list. The output of this normalization component, which is a normalized text, used as a direct input to the stop word removal component. Example pseudo-code is described in Algorithm 4-5 for short word expansion.

```
Input: Cliticized Words Normalized Corpus
Output: Normalized Corpus
BEGIN
load the Tigrigna short words list file
open the corpus
while not end of the file
      for each word in the corpus
          if the word is in the Tigrigna short words list
              replace it with its expanded form
          end if
end while
close file
END
```

Algorithm 4-5 Short word expansion normalization

4.4.3 Stop Word Removal

The other preprocessing task is stop word removal, a common approach to reduce noise in the data, when working with text classification methods. The most common words in text documents such as articles, prepositions and pronouns are treated as stop words. These words do give less significance for analyzing sentiment yet their frequency count dominates all other words. Hence, removing stop words helps to reduce the dimensionality of a term space. We have manually prepared stop word list from the prepared corpus for this research and is compiled in APPENDIX A: List of Tigrigna Stop-Words.

```
Input: Normalized Corpus
Output: Preprocessed Corpus
BEGIN
load the Tigrigna stop word list file
open the corpus
while not end of the file
      read the word
      if the word is found in the stop word list
            remove it
      end if
end while
return the remaining words
close the file
END
```

Algorithm 4-6 Stop Word Removal

4.5 Morphological Analysis

Morphological analysis is the segmentation of words into their component morphemes, returning the root of a word. A morpheme is the smallest meaningful unit of a given language e.g., token. Morphological analysis is very vital for various natural language process applications such as sentiment analysis. In this work, stemming and lemmatization, morphological analysis task, is performed using HornMorpho morphological analyzer given a training and testing data for extracting the required features to build a classifier. HornMorpho is a freely available Python program that analyzes Amharic, Oromo, and Tigrigna words into their constituent morphemes (meaningful parts) and generates words, given a root or stem and a representation of the word‘s grammatical structure(Gasser, 2011).

In the process of both stemming and lemmatization, special attention is given to the words like ኣይኮነን/Aykonen (it is not), because it has another form ኮነ/kone (it is) with a positive polarity which is opposite to original words ኣይኮነን/Aykonen with negative sentiment polarity. Therefore, we represent two words containing the stem in stemming, lemma in lemmatization and its negative marker, so that it will keep its original sentiment orientation.

4.5.1 Lemmatization

This morphological analysis task, converts every word of the post to their base forms to avoid data sparseness. Lemmatizing is done in this research by integrating HornMorpho with our system. HornMorpho analyses a given word in detail, as a result, the output is further parsed and unnecessary elements are removed to make it suitable for the next process using regular expression.(Abdul-mageeds et al., 2013)’s experimental result showed that morphological analysis has more influence and impact on sentiment analysis of Semitic languages. As a result, the preprocessed dataset is lemmatized before the training of the learning model to reduce data sparseness.

4.5.2 Stemming

Stemming is the process of reducing inflection towards their stem and it occurs in such a way that depicting a group of relatable words under the same stem, even if the root has no appropriate meaning. (Wahbeh et al., 2011) conducted an experiment using SVM and two test models in to show the effect of stemming on Arabic text classification by using stemming as part of pre-processing steps. The results show that applying stemming negatively affects the accuracy of the model. A research conducted

by (Turegn Fikre, 2020),concluded that stemming in preprocessing step has negative impact on Amharic sentiment analysis. In this study we tried to explore the impact of stemming on Tigrigna multiscale sentiment analysis. Therefore, an experiment is conducted using both stemmed and unstemmed dataset as an input to the classifier, to investigate impact of stemming for multiscale Tigrigna sentiment analysis.

4.6 Feature Extraction and Representation

Feature Extraction, one of the key components in multi-scale sentiment analysis, is a process of extracting and selecting features by which an initial set of raw data is reduced to potential and more manageable features for further processing. Feature extraction is the name for a method that selects and combines variables into features. It accurately and efficiently describes the original data set while effectively reducing the amount of data that needs to be processed. It helps in eliminating irrelevant variables to enhance the generalization performance of the system.

N-gram model is one of the most popular models that is widely used in sentence-based language processing. N-grams are contiguous sequences of words with length where n items are considered from the given sentence. There are multiple n-grams like unigram, bigram, trigrams etc. If we consider a sentence, **ፅቡቕ ኣገንዑ ኣበርትዕኩም ስርሑ።**/'SbuQ Agen'U abert'Kum srHu/Good job,keep it up**, when** n=1, then it will produce **ፅቡቕ**/'SbuQ, **ኣገንዑ**/ Agen'U, **ኣበርትዕኩም**/ abert'Kum**, ስርሑ**/ srHu. If we consider n=2,then it will produce **ፅቡቕ ኣገንዑ**/'SbuQ Agen'U, **ኣገንዑ ኣበርትዕኩም**/ Agen'U abert'Kum, **ኣበርትዕኩም ስርሑ**/ abert'Kum srHu. If we consider n=3, then it will produce **ፅቡቕ ኣገንዑ ኣበርትዕኩም**/'SbuQ Agen'U abert'Kum srHu, **ኣገንዑ ኣበርትዕኩም ስርሑ**/ Agen'U abert'Kum srHu.

The main aim of feature extraction is finding a suitable set of features that improves the classification accuracy. Given its importance, we have performed the feature extraction task through different variants of n-gram model in our study. Once the data is pre-processed, features relevant for sentiment analysis are extracted. On the cleaned data, unigram, bigram, trigram and hybrid of unigram and bigram variants of n-gram model is applied to form a feature vector space. The simplicity and scalable nature of n-grams makes them effective.

N-gram features have been used in various NLP studies for various tasks including sentiment analysis. Many experiments in previous research have proven that n-grams improved the quality of feature sets.(Wondwossen Philemon & Wondwossen Mulugeta,

2014) used unigram, bigram and hybrid variants of n-gram as features and achieved 43.6%, 44.3%, and 39.5% respectively by employing naïve Bayes algorithm. (Abreham Getachew, 2014) used unigram as a feature with the most informative words and achieved 90.9%, 83.1% and 89.6% for naive Bayes, decision tree and maximum entropy respectively. Mourad & Darwish (2013) used number of features like POS, n-gram, tweets-specific features, presence of emoticons, usage of decorating characters, punctuations, elongations and repetitions by implementing Naïve Bayes and SVM classifiers using NLTK tool and Naïve Bayes perform better than SVM.

Feature Representation, main step in a machine learning text classifier, is a process of transforming the text into a numerical representation, usually a vector. The process of automatic feature extraction uses preprocessed lemmatized text as an input. The input is large labeled data, and once this data is preprocessed, both lemmatized and stemmed features are extracted before it is used for training. In feature representation, once the features are extracted from the training and test data, the dataset is transformed into a feature vector. The output from this stage is a fixed-size vector representation for each word using term frequency inverse document frequency (TF-IDF). It reflects the importance of a word in the corpus or the collection. TF-IDF value increases with increase in frequency of a particular word in the document. In order to control the generality of more common words, the term frequency is offset by the frequency of words in corpus. Term frequency is the number of times a particular term appears in the text. Inverse document frequency measures the occurrence of any word in all documents. Except the neural class, both our training and testing corpus is composed of shorter length sentences because our data sources are social media platforms such as Facebook where users write short length sentences as comment or feedback. Besides with the removal of stop words most relevant higher order n-grams will be reduced to a bigram or trigram, as a result, we employed only unigram, hybrid of unigram and bigram, hybrid of unigram and trigram, bigram, hybrid of bigram and trigram and trigram variants of N-gram as a feature for conducting our experiment.

4.7 Training Learning Models

The supervised machine learning algorithms are used for training the Tigrigna multi-scale sentiment analysis model. The classification model decides how to classify based on pattern and associating the patterns to the unlabeled new data. Once the required features are extracted from the dataset, the next step of the experiment is to train the classifier with a pre-defined training feature vectors with a number of machine learning

classifiers that are proved to provide high accuracy in sentiment analysis for similar cases. Three supervised machine-learning classifiers were used: Support Vector Machine, Naive Bayes, and Maximum Entropy because of their reported performance in previous sentiment-analysis studies (Wondwossen Philemon & Wondwossen Mulugeta, 2014);(Abreham Getachew, 2014); (Abdul-mageed et al., 2013). They employed SVM supervised machine learning algorithms for classification and achieved a very promising result. Mourad & Darwish (2013) implemented both Naïve Bayes and SVM classifiers using NLTK tool and claimed that their performance result suppresses all the Arabic subjectivity and sentiment analysis previous studies.

4.8 Sentiment Classification

The final stage of the multiscale sentiment analysis of Tigrigna texts is application of the classification to an unseen (testing) data on the trained learning model. The built sentiment learning model classifies the given sentences into one of the five classes (very positive, positive, neutral negative and very negative). The output of the sentiment classification component is the sentiment polarity value. For example, if this sentence ፅብቕ ስራሕ እዩ ቀፅልዎ/'SbQ sraH iyu qe'Slwo (*Good job! Keep it up)* then the polarity level of this sentence is positive, yet polarity value of the sentence ብጣዕሚ ፅብቕ ስራሕ እዩ ቀፅልዎ/bTa'mi 'SbQ sraH iyu qe'Slwo (very good job! Keep it up) is very positive because of the word ብጣዕሚ(very).

CHAPTER 5
EXPERIMANTATION AND EVALUATION

5.1 Overview

This chapter presents the demonstration of the proposed multiscale Tigrigna sentiment analysis model. demonstration is the process of using the artifact to solve one or more instances of the problem (Peffers et al., 2007).In this chapter, the data collection,annotation,and description, the implementation and experimental setups, procedures, results, evaluations and discussion of the proposed model results is discussed in detail.

5.2 Corpus Preparation

5.2.1 Data Collection

A machine learning approach requires a corpus for both training and testing and the data for this proposed study are the comments, feedback or blog contents written by readers, users or customers. As far as our knowledge, there is no publicly available prepared Tigrigna dataset for sentiment analysis. As a result, we have collected 1500 sentences from different sources such as Facebook, YouTube, Twitter, SBS Tigrigna, VOA Tigrigna, BBC Tigrigna, Fana Tigrigna and Walta Tigrigna websites manually.

The main reason why we selected these data sources is the platforms provide their services in Tigrigna and make their users engage in their issues. The collected texts are handpicked sentences from domains such as politics, education, entertainment and sport. The main reason to why we used the domains is due to the lack of readily available reviews written in Tigrigna and we believe that, large number of reviews can be collected easily from multiple domains than a single domain. In addition, these domains are among topics that users engaged and participated actively and freely by their own language on different platform.

As far as we know, there is also no Tigrigna stop words list available for use and we manually prepared the list, which contains five hundred and fifty-five (555) stop words using mainly the corpus and other Tigrigna texts as a source. To the best of our knowledge, there is also no publicly available, contractions list for use. As a result we have adopted (Hailay Beyene, 2013)’s contractions list and updated it .we have used seventy seven(77) short words in this study. All words in the prepared stop word list and short word list are normalized for data uniformity and consistency. We have also prepared a small list of cliticized words (words joined by an apostrophe), which

contains 10 common cliticized words, that needs to be separated into their constituent parts in order to normalize the corpus. The collected sentences with its size and data sources are summarized and presented in Table 5-1.

Source	Size
Facebook	620
YouTube	655
Fana Tigrigna	135
SBS Tigrigna	25
BBC Tigrinya	28
VOA Tigrinya	22
Twitter	20
Total	1500

Table 5-1 Data Collection Summary

The collected sentences are stored in a CSV formatted file and, any time we want to incorporate new data, we simply append on the file that stored the reviews. A language experts assisted in the sentiment corpus preparation. Correction of spelling and grammatical errors and removing irrelevant contents such as Tweeter hashtags were done manually.

5.2.2 Data Annotation

This activity is concerned with labeling the collected sentences for experimental purpose with the help of human annotators, which takes much developing time and results in disagreement between annotators. All the collected sentences are manually categorized into predefined categories: Very positive (2), positive (1), neutral (0), negative (-1) and very negative (-2). A similar number of sentences for each category were annotated manually. We have engaged two native Tigrigna language speakers' annotators to assist with the corpus annotation. We have measured the conformity between the annotators and found 81.4% value of the Kappa (K) parameter, which shows an acceptable strength of agreement between the two annotators.

Sentences from different domains; politics, entertainment, education and sport were given to the annotators to categorize the sentences based on their judgment. The annotators were trained on how to label the sentences into one of the predefined classes by considering the content of the sentence. In the first round, each annotator was given

750 sentences. In the next pass, the 750 sentences were swapped between the two annotators. In the end, each of the annotators had annotated 1500 sentences. Then we have verified the final polarity annotation of the sentences by examining each sentence manually. If two of the annotators were agreed in annotating polarity of the sentence, then the respective sentence was labeled as given by the annotators. When they do not agree, then we decided polarity of the sentence. The annotated corpus contains 1,500 sentences collected manually from YouTube, Facebook, and websites of FBC Tigrigna, Walta Tigrigna, VOA Tigrigna, BBC Tigrigna, and SBS Tigrigna for sport, politics, education and entertainment domains.

5.2.3 Dataset Description

The corpus consisting of 1500 sentences is a balanced corpus where there are equal number of sentences for each class. The corpus is divided into training set which consists of 80 % (1,200 sentences) and testing set consists the remaining 20 % (300 sentences). A sample sentences for each class is attached in Appendixes F-J.

No.	Name of Classes	Label of Classes	Number of Sentences
1	Very Positive	2	300
2	Positive	1	300
3	Neutral	0	300
4	Negative	-1	300
5	Very Negative	-2	300
Total			1500

Table 5-2 Number of annotated sentences for each class

5.3 Implementation

In this study, python programming language with different development tools and packages is used in the process of experimenting this study. The tools that have been used include Jupyter Notebook, NLTK, Pandas, Numpy, Scikit learn machine learning library and HornMorpho to build a machine learning model.

NumPy is a library for python that solve scientific computation easily. **Pandas** is an open-source library that is used to read CSV files and perform different operations on the CSV files. **NLTK** is a package in python used for many tasks like tokenization, normalization, stop word removal lemmatization, stemming and POS tagging. **Scikit-learn** is a library for python machine learning library, which contains simple and

efficient tools for data mining and data analysis algorithms for both supervised and unsupervised problems. **HornMorpho**, part of the L3 project at Indiana University, is freely available python program used for analyzing Afaan Oromo, Amharic and Tigrigna words into their meaningful parts. We have used HornMorpho for both lemmatization and stemming of the dataset.

Figure 2-1 shows the important libraries that are imported for building the machine learning model.

```
import pandas as pd
import nltk
import re
import string
from nltk.corpus import stopwords
from nltk.tokenize import WordPunctTokenize
import matplotlib.pyplot as plt
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.svm import LinearSVC
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
import seaborn as sns
```

Figure 5-1 Important packages imported for experimentation

Once the necessary libraries and packages are imported, we loaded the prepared and annotated dataset file using pandas to load the data as a DataFrames from the disk as shown in Figure 5-2. The pandas method read_csv reads file in the tab separated file and load it as DataFrames, where instances of the data are accessible via column name.

```
import pandas as pd
tsa= pd.read_csv('Final_TSA_Dataset300.csv', sep='\t',encoding="utf8")
```

Figure 5-2 Loading the dataset

After dataset has been loaded, the next step is to preprocess it which involves removing unimportant and noisy elements for the next stage of analysis. We used NLTK python module to tokenize the dataset and removing Tigrigna stop words from the dataset. We have also used regular expression in order to normalize homophones and remove punctuation marks, special characters, symbols, emojis, numbers, extra spaces, and so on. Figure 5-3 shows the included activities of dataset preprocessing step of the experiment.

```
def preprocess_sent(new_sent):
    preprocessed_sent=new_sent
    dictionary = {'ጸ':'ፀ','ጹ':'ፁ','ጺ':'ፂ','ጻ':'ፃ','ጼ':'ፄ','ጽ':'ፅ','ጾ':'ፆ',
                  'ሠ':'ሰ','ሡ':'ሱ','ሢ':'ሲ','ሣ':'ሳ','ሤ':'ሴ','ሥ':'ስ','ሦ':'ሶ',
                  'ኅ':'ሀ','ኍ':'ሁ','ኂ':'ሂ','ኃ':'ሄ','ኈ':'ሆ','ኋ':'ሏ',}
    preprocessed_sent = preprocessed_sent.translate(preprocessed_sent.maketrans(dictionary))
    preprocessed_sent=identify_tokens(preprocessed_sent)
    preprocessed_sent=con_expansion(preprocessed_sent)
    Tigstopword = nltk.corpus.stopwords.words('Tigrigna')
    preprocessed_sent = [word for word in preprocessed_sent if not word in stopwords.words()]
    preprocessed_sent=remove_punctuations(preprocessed_sent)
    preprocessed_sent = preprocessed_sent.replace('\d+', '')
    preprocessed_sent=remove_emoji(preprocessed_sent)
    preprocessed_sent = preprocessed_sent.replace(r'[^\u1200-\u137F]', ' ')
    preprocessed_sent = re.sub(' +',' ',preprocessed_sent)
    return preprocessed_sent
```

Figure 5-3 Dataset preprocessing

Once the preprocessing work is finished, lemmatization of the dataset followed. We used anal_file () function from HornMorpho tool for lemmatizing the preprocessed dataset. Figure 5-4 presents the code to lemmatize the preprocessed dataset.

```
def lemmatize_all_dataset():
    tsa['Processed_'] = tsa['Processed'].str.replace(' ', '_')
    with open('lemma_tsa_22.txt', 'w',encoding="utf8") as f:
        f.write(tsa['Processed_'].str.cat(sep='።'))
    hm.anal_file('ti', 'lemma_tsa_22.txt','lemma_tsa_output_22.txt',nbest=1)
    mystr='።'
    f = open('lemma_tsa_output_22.txt', 'r+',encoding="utf8")
    lines = f.readlines()
    mystr = ''.join([line.strip() for line in lines])
    mystr=mystr.split('።: ።')
    text_file = open("Lemmatized_tsa_Output_22.txt", "wt",encoding="utf8")
    text_file.write("Lemmatized\n")
    for sent_token in mystr:
        token=sent_token.split('_: _')
        neg='negative'
        for t in token:
            if neg in t:
                t=re.sub(r'^(.*?)word:',r'', str(t))
                t=re.sub(r'POS(.*?)$',r'', str(t))
            t=re.sub(r'<(.*?)>',r'', str(t))
            t=re.sub(r'\|(.*?)\:',r'', str(t))
            t=re.sub(r',',r'', str(t))
            re.sub(r'=(.*?)',r' ', str(t))
            t=re.sub(r'grammar(.*?)',r' ', str(t))
            t=re.sub(r'word:(.*?)citation:',r'x', str(t))
            t=re.sub(r'[^\u1200-\u137F]',r'', t)
            text_file.write(t)
            text_file.write(" ")
        text_file.write("\n")
```

Figure 5-4 Lemmatization

Once the dataset was loaded, preprocessed and lemmatized, the next step is to extract the feature from the lemmatized dataset and we used the python Scikit-learn module TfidfVectorizer as a vector transformation method because machine learning models operate on vectors instead of words. Then three learning models (Naïve Bayes, Maximum Entropy, Support Vector Machine) were trained by using the training dataset and tested using the testing dataset. Figure 5-5 shows TfidfVectorizer implementation for unigram language model. We used 20% of the corpus for testing the model.

```
cv_counts_lg=TfidfVectorizer(ngram_range=(1,1),analyzer='word')
X_counts_lg=cv_counts_lg.fit_transform(tsa.Processed).toarray()
X_train_lg, X_test_lg, y_train_lg, y_test_lg = train_test_split(X_counts_lg, tsa.Polarity, test_size=0.2,shuffle=True,
                                                                random_state=123,stratify=tsa.Polarity)
```

Figure 5-5 Tfidf vectorizer for unigram language model

For implementing the Naïve Bayes, we used MultinomialNB() function of sklearn, as it suitable for classification of multi-class data. Figure 5-6 shows the implementation of Naïve Bayes.

```
clf_Multinomial=MultinomialNB()
clf_Multinomial.fit(X_train,y_train)
y_predict=clf_Multinomial.predict(X_test);
```

Figure 5-6 Naïve Bayes model

we used LogisticRegression() function of sklearn package to build the Logistic Regression (aka MaxEnt) model. Figure 5-7 shows the implementation of MaxEnt.

```
logisReg=LogisticRegression()
logisReg.fit(X_train_lg,y_train_lg)
y_predict_lg=logisReg.predict(X_test_lg);
```

Figure 5-7 MaxEnt Model

We used LinearSVC() function of the sklearn package for building the SVM model. Figure 5-8 shows the implementation of SVM.

```
linSVC=LinearSVC()
linSVC.fit(X_train_svc,y_train_svc)
y_predict_svc=linSVC.predict(X_test_svc);
```

Figure 5-8 SVM Model

5.4 Experimental Results

The experimental procedure is, as is standard in supervised machine learning tasks, first training a classifier on pre-annotated training data set and then evaluating the performance of the classifier on unlabeled data set. Once the prepared dataset was preprocessed, we converted it to a numeric vector matrix using tfidf vectorizer. We have then lemmatized and stemmed the preprocessed dataset in a parallel stage. We have then used the preprocessed lemmatized dataset for conducting experiments that intends to answer RQ1 and RQ2; while the preprocessed stemmed dataset for conducting the experiments that intends to answer RQ3 [explore the effect of stemming in Tigrigna sentiment analysis].

The classification report below shows performance of the unigram model using Naïve Bayes learning model.

Classification Report of Multinomial Naive Bayes with Unigram Language Model

	precision	recall	f1-score	support
-2	0.50	0.55	0.52	60
-1	0.70	0.35	0.47	60
0	0.90	0.90	0.90	60
1	0.59	0.60	0.60	60
2	0.57	0.78	0.66	60
accuracy			0.64	300
macro avg	0.65	0.64	0.63	300
weighted avg	0.65	0.64	0.63	300

Figure 5-9 Unigram Language Model Output using Naïve Bayes

As results shown in the Figure 5-9 , NB classifier achieve an overall accuracy of 65% for classifying Tigrigna multi-scale sentiment texts. The classifier scored a precision of 54%,68%,88%,64% and 57% for very negative, negative, neutral, positive and very positive classes respectively. The neutral class misclassified 12% which is lower error rate comparing to the other classes.

Classification Report of MaxEnt with Unigram Language Model

	precision	recall	f1-score	support
-2	0.56	0.47	0.51	60
-1	0.62	0.70	0.66	60
0	0.82	0.88	0.85	60
1	0.75	0.72	0.74	60
2	0.68	0.68	0.68	60
accuracy			0.69	300
macro avg	0.69	0.69	0.69	300
weighted avg	0.69	0.69	0.69	300

Figure 5-10 presents the classification reports of unigram model using MaxEnt and scored 68% of accuracy, weighted precision, recall and f1-score.

Figure 5-10 Unigram Language Model Output using MaxEnt

The very negative polarity class classified correctly with 58%, and misclassified with 32%. The neutral polarity class classified correctly with 78% and misclassified with 22%. The results show neutral class gained lower precision error rate while very negative class scored higher precision error rate.

Classification Report of SVM with Unigram Language Model

	precision	recall	f1-score	support
-2	0.61	0.55	0.58	60
-1	0.63	0.62	0.62	60
0	0.85	0.87	0.86	60
1	0.79	0.73	0.76	60
2	0.69	0.80	0.74	60
accuracy			0.71	300
macro avg	0.71	0.71	0.71	300
weighted avg	0.71	0.71	0.71	300

Figure 5-11 Unigram Language Model Output using SVM

The results in Figure 5-11 shows that, the very positive, positive, negative, very negative and neutral classes correctly classified true positive 77%,69%,59%,61% and 84% respectively which is better performance result comparing it with above shown results.

The experiments for this study were done using unigram, hybrid (unigram + bigram), hybrid (unigram + trigram), bigram, hybrid (bigram + trigram), and trigram N-gram models with a three learning models: Naïve Bayes, SVM and Maximum Entropy given the same dataset. In this study, we have conducted 36 experiments categorized into to two-categories with 18 experiments for category one and the rest in category two. The first category of experiments is done with a preprocessed lemmatized dataset and they are designed to compare performance of the used language models and learning models. The second category of experiments is conducted with a preprocessed stemmed dataset as an input and are designed to explore the effect of stemming in Tigrigna sentiment analysis. The experiments were conducted to measure the overall performance of the developed Tigrigna sentiment analysis system. At the end of the experiment, all the performance results of each language and learning models mentioned above are recorded and presented in the forthcoming section according to their experimental categories.

5.4.1 Experimental Results: Comparison of Models and Algorithms

In this subsection, we have performed a comparative analysis of the algorithms used with each language models we have employed.

The following table presents experimental results of unigram model using the employed three algorithms.

Algorithm	Weighted Average			
	Precision	Recall	F1-Score	Accuracy
NB	0.65	0.64	0.63	0.64
MaxEnt	0.69	0.69	0.69	0.69
SVM	0.71	0.71	0.71	0.716

Table 5-3 Experimental Results of Unigram Model

As shown in Table 5-3, we achieved an accuracy of 69% using a MaxEnt, 71.6% using SVM and 64% using a Naïve Bayes. Based on the result, SVM works better with 71.6% of accuracy.

The following table presents experimental results of Hybrid (unigram + bigram) model using the employed three algorithms.

Algorithm	Weighted Average			
	Precision	Recall	F1-Score	Accuracy
NB	0.66	0.65	0.65	0.65
MaxEnt	0.69	0.69	0.69	0.69
SVM	0.70	0.70	0.70	0.70

Table 5-4 Experimental Results of Hybrid Model

Based on the result presented in Table 5-4, SVM outperforms Naïve Bayes by 5% and MaxEnt by 1%, achieving accuracy of 70%.

The following table presents experimental results of hybrid (unigram + trigram) model using the employed three algorithms.

Algorithm	Weighted Average			
	Precision	Recall	F1-Score	Accuracy
NB	0.66	0.66	0.65	0.66
MaxEnt	0.70	0.69	0.69	0.69
SVM	0.71	0.71	0.71	0.71

Table 5-5 Experimental Results of hybrid (unigram + trigram) Model

As shown in Table 5-5Table 5-3, we achieved an accuracy of 69% using a MaxEnt, 71% using SVM and 66% using a Naïve Bayes. Based on the result, SVM works better with 71% of accuracy.

The following table shows, experimental results of Bigram model using the employed algorithms.

Algorithm	Weighted Average			
	Precision	Recall	F1-Score	Accuracy
NB	0.65	0.42	0.41	0.42
MaxEnt	0.71	0.47	0.47	0.47
SVM	0.71	0.47	0.47	0.47

Table 5-6 Experimental Results of Bigram Model

Based on the result presented in Table 5-6, SVM and MaxEnt achieved 47% of accuracy and outperformed naïve Bayes with 5%.

The following table shows, experimental results of hybrid (bigram + trigram) model using the employed algorithms.

Algorithm	Weighted Average			
	Precision	Recall	F1-Score	Accuracy
NB	0.57	0.39	0.38	0.39
MaxEnt	0.71	0.46	0.46	0.46
SVM	0.70	0.46	0.47	0.46

Table 5-7 Experimental Results of hybrid (bigram + trigram) Model

Based on the result presented in Table 5-7, SVM and MaxEnt achieved 46% of accuracy and outperformed naïve Bayes with 7%.

The following table presents, experimental results of Trigram model for all employed algorithms.

Algorithm	Weighted Average			
	Precision	Recall	F1-Score	Accuracy
NB	0.56	0.21	0.17	0.21
MaxEnt	0.74	0.26	0.17	0.26
SVM	0.74	0.26	0.17	0.26

Table 5-8 Experimental Results of Trigram Model

The results in Table 5-8 showed that the SVM and MaxEnt performed similar scores in the trigram model with an accuracy of 26%, while naïve Bayes scored lower than the two with an accuracy of 21%.

Figure 5-12 shows summary of naïve Bayes, maximum entropy and support vector machine gained accuracies for each experimented language model.

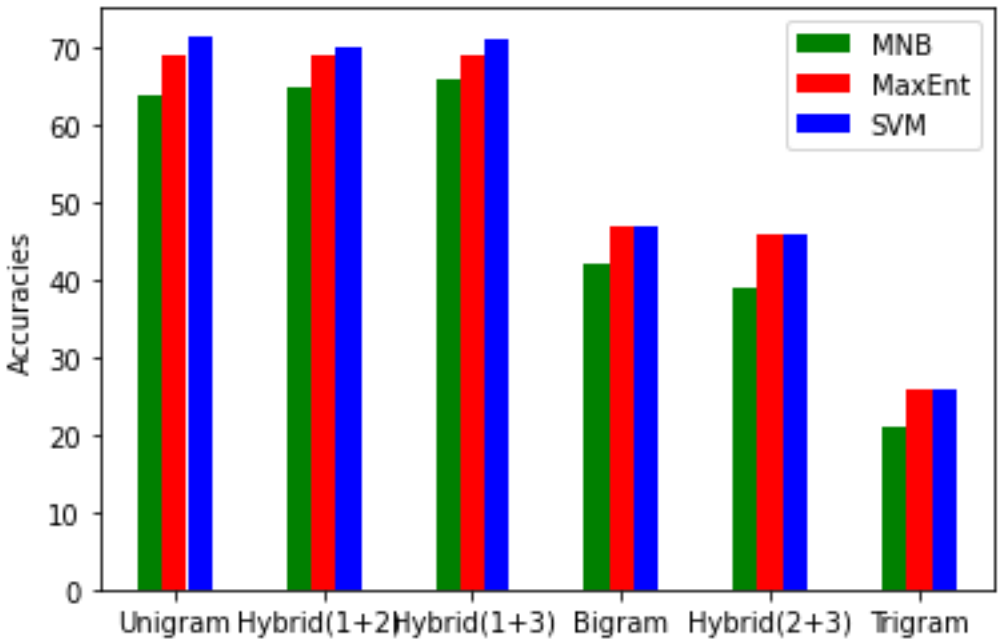

Figure 5-12 Summary Experimental Results in terms of Accuracy

The accuracy of the naïve Bayes in unigram, hybrid of unigram and bigram, hybrid of unigram and trigram, bigram, hybrid of bigram and trigram, and trigram is 64%, 65%,66%,42%, 39%and 21% respectively. The accuracy of the maximum entropy in unigram, hybrid of unigram and bigram, hybrid of unigram and trigram, bigram, hybrid of bigram and trigram, and trigram is 69%, 69%,69%,47%, 46%and 26% respectively. The accuracy of the support vector machine in unigram, hybrid of unigram and bigram, hybrid of unigram and trigram, bigram, hybrid of bigram and trigram, and trigram is 71.6%, 70%,71%,47% ,46% and 26% respectively.

As shown in Figure 5-12, SVM achieved the highest accuracy with unigram language model and it does not take much training data to start providing accurate results, although it took more computational resources. In short, SVM takes care of drawing a line or hyperplane that divides a space into two subspaces: one subspace that contains vectors that belong to a class and another subspace that contains vectors that do not belong to that class.

Naïve-Bayes performs very poorly when features are highly correlated which is limitations of NB classifier and since our dataset contains highly correlated features NB results are poor for all language models when comparing to SVM and MaxEnt. Since MaxEnt is much more robust to correlated features; it has achieved the second-highest result next to SVM.We have also observed that the learning model’s accuracy is better

for unigram, hybrid of unigram and bigram, and hybrid of unigram and trigram language models and this is because they are less sparse than the bigrams and trigrams as we have used small amount of dataset in a multiscale polarity classification. Generally, the achieved evaluation results are encouraging and we are convinced more training data could improve the performance.

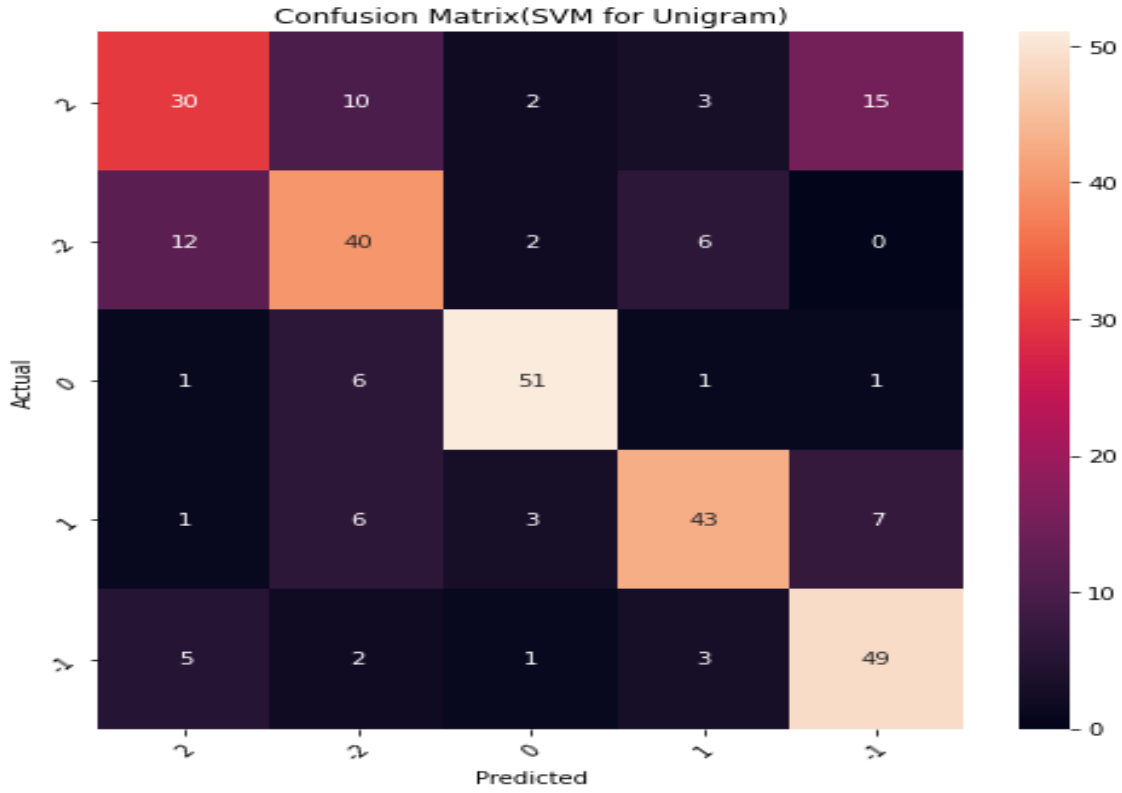

Figure 5-13 Confusion matrix of SVM with Unigram

In general, the experimental results show that SVM with unigram language model performed better and is suggested as a classification model for the Tigrigna multi-scale sentiment analysis system. Figure 5-13 shows the confusion matrix of SVM with unigram language model and it presents the correct classifications and misclassifications of each polarity class represent as 2, -2,0,1 and -1 for very positive, very negative, neutral, positive and negative respectively. We have used 300 sentences which is 20% of the balanced corpus for the purpose of testing the learned models, as a result each polarity class is comprised of 60 sentences in the testing dataset. The confusion matrix shows how many of the sentences are correctly classified and misclassified, for example out the 60 sentences of the neutral polarity class, 51 are correctly classified and 9 are incorrectly classified. Besides to the neutral polarity class, both the positive and negative polarity classes also works better than the very positive and very negative polarity classes. The very negative and very positive polarity classes

are mainly comprised of diminishers and intensifiers in our corpus which the unigram language model could not capture all of them effectively as it only considers one word of the given sentence. As a result, the classifier has poorly classified a correct classification of 30 and 40 out of the 60 given sentences in the dataset for the very positive and very negative polarity classes respectively as it is shown in Figure 5-13.

5.4.2 Experimental Results: Effect of Stemming

In this subsection, 18 experiments are conducted mainly to see the effect of stemming in Tigrigna sentiment analysis using NB, MaxEnt and SVM with each language models employed on both unstemmed and stemmed preprocessed dataset for the experiments. Table 5-9 shows experiment evaluation results of unigram model comparing the effect of stemming using all the employed algorithms.

Algorithm	Stemming Status	Weighted Average			Accuracy
		Precision	Recall	F1-Score	
NB	Without stemming	0.65	0.64	0.63	0.64
	With stemming	0.66	0.64	0.64	0.64
MaxEnt	Without stemming	0.69	0.69	0.69	0.69
	With stemming	0.70	0.69	0.69	0.69
SVM	Without stemming	0.71	0.71	0.71	0.716
	With stemming	0.69	0.69	0.69	0.69

Table 5-9: Experiment Result of Unigram Model: Effect of Stemming

As presented in Table 5-9 the accuracy of Tigrigna sentiment analysis with stemming is lower than without stemming in the unigram language model by 2.6% for SVM and it does not show any change in the case of MaxEnt and NB.

Table 5-10 shows experiment evaluation results of hybrid of unigram and bigram model comparing the effect of stemming using all the employed algorithms.

Algorithm	Stemming Status	Weighted Average			Accuracy
		Precision	Recall	F1-Score	
NB	Without stemming	0.66	0.65	0.65	0.65
	With stemming	0.67	0.66	0.66	0.66
MaxEnt	Without stemming	0.69	0.69	0.69	0.69
	With stemming	0.69	0.69	0.69	0.69
SVM	Without stemming	0.70	0.70	0.70	0.70
	With stemming	0.70	0.70	0.70	0.70

Table 5-10: Experiment Result of Hybrid (unigram + bigram) Model: Effect of Stemming

As shown in Table 5-10 the accuracy of Tigrigna sentiment analysis without stemming is lower than with stemming in the of unigram and bigram language model by 1% for NB. However, in the case of SVM and MaxEnt it does not show any change regarding the accuracy.

Table 5-11 shows experiment evaluation results of hybrid of unigram and trigram model comparing the effect of stemming using all the employed algorithms.

Algorithm	Stemming Status	Weighted Average			Accuracy
		Precision	Recall	F1-Score	
NB	Without stemming	0.66	0.66	0.65	0.66
	With stemming	0.67	0.65	0.65	0.65
MaxEnt	Without stemming	0.70	0.69	0.69	0.69
	With stemming	0.71	0.71	0.71	0.71
SVM	Without stemming	0.71	0.71	0.71	0.713
	With stemming	0.70	0.70	0.70	0.70

Table 5-11: Experiment Result of Hybrid (unigram + trigram) Model: Effect of Stemming

As shown in Table 5-11 the accuracy of Tigrigna sentiment analysis with stemming is lower than without stemming in the hybrid of unigram and trigram language model by

1% for NB and, 2% for MaxEnt. However, in the case of SVM, accuracy of without stemming is higher than the stemmed by 1.3%.

Table 5-12 shows experiment evaluation results of bigram model comparing the effect of stemming using all the employed classification algorithms.

Algorithm	Stemming Status	Weighted Average			Accuracy
		Precision	Recall	F1-Score	
NB	Without stemming	0.65	0.42	0.41	0.42
	With stemming	0.59	0.42	0.43	0.42
MaxEnt	Without stemming	0.71	0.47	0.47	0.47
	With stemming	0.65	0.49	0.50	0.49
SVM	Without stemming	0.71	0.47	0.47	0.47
	With stemming	0.66	0.50	0.50	0.50

Table 5-12: Experiment Result of Bigram: Effect of Stemming

As presented in Table 5-12 accuracy of Tigrigna sentiment analysis with stemming is higher than without stemming in the bigram language model by 2% and 3% for MaxEnt and SVM respectively. However, it does not show any change in NB learning model.

Table 5-13 shows experiment evaluation results of hybrid of bigram and trigram model comparing the effect of stemming using all the employed classification algorithms.

Algorithm	Stemming Status	Weighted Average			Accuracy
		Precision	Recall	F1-Score	
NB	Without stemming	0.57	0.39	0.38	0.39
	With stemming	0.59	0.47	0.48	0.42
MaxEnt	Without stemming	0.71	0.46	0.46	0.46
	With stemming	0.67	0.49	0.50	0.49
SVM	Without stemming	0.70	0.46	0.47	0.46
	With stemming	0.66	0.49	0.50	0.49

Table 5-13: Experiment Result of Hybrid (Bigram + Trigram): Effect of Stemming

As presented in Table 5-13 accuracy of Tigrigna sentiment analysis with stemming is higher than without stemming in the bigram language model by 3% for all the three employed learning models.

Table 5-14 shows experiment evaluation results of trigram model comparing the effect of stemming using all the employed classification algorithms.

Algorithm	Stemming Status	Weighted Average			Accuracy
		Precision	Recall	F1-Score	
NB	Without stemming	0.56	0.21	0.17	0.21
	With stemming	0.56	0.23	0.19	0.23
MaxEnt	Without stemming	0.74	0.26	0.17	0.26
	With stemming	0.64	0.27	0.19	0.27
SVM	Without stemming	0.74	0.26	0.17	0.26
	With stemming	0.64	0.27	0.19	0.27

Table 5-14: Experiment Result of Trigram Model: Effect of Stemming

As shown in Table 5-14 accuracy of Tigrigna sentiment analysis with stemming is higher than without stemming in the trigram language model by 1%, 2% and 2% for NB, MaxEnt and SVM respectively.

Table 5-15 presents summary results of all the employed language and learning models with and without stemming.

	Without Stemming			With Stemming		
Models/Algorithms	NB	MaxEnt	SVM	NB	MaxEnt	SVM
Unigram	0.64	0.69	0.716	0.64	0.69	0.69
Hybrid (1+2)	0.65	0.69	0.70	0.66	0.69	0.70
Hybrid (1+3)	0.66	0.69	0.71	0.65	0.71	0.70
Bigram	0.42	0.47	0.47	0.47	0.49	0.50
Hybrid (2+3)	0.39	0.46	0.46	0.42	0.49	0.49
Trigram	0.21	0.26	0.26	0.23	0.27	0.27

Table 5-15 Summary of Effect of Stemming in terms of Accuracy

Accuracy of Naïve bayes for hybrid of unigram and trigram language model reduces from 66% to 65% and achieved same result for unigram feature. But it increases 1%,5%,3% and 2% for hybrid of unigram and bigram, bigram, hybrid of bigram and trigram and trigram respectively. In the case of MaxEnt, the accuracy increases from for hybrid of unigram and trigram, bigram, hybrid of bigram and trigram, and trigram

by 2%,2%,3% and 1% respectively, however it does not show any change for unigram and hybrid of unigram and bigram models. The accuracy of SVM reduces for unigram and hybrid of unigram and trigram, but increases for bigram, hybrid of bigram and trigram, and trigram language models. It does not show any change for hybrid of unigram and bigram language model. In general, since stemming removes affixes, and these trimmed affixes, some are indicative of sentiment and their removal leads to system under-performance when using a stemmed dataset for sentiment classification.

5.5 Prototype

In order to test the proposed model, we have developed a window based graphical user interface prototype using a python programming language Tkinter package. The main task is to predict whether a given sentence is very positive, positive, neutral, negative or very negative. Sample of the prototype that shows accepting Tigrigna text inputs from the user through the data input widget and displaying the polarity classification of the given input text is shown in Figure 5-14. The demo indicates that the user can write his/her sentiment in Tigrigna towards a target of object in the input text widget through and submit so that the opinion can be pre-processed, morphologically analyzed, classified and the polarity classification result displayed in the polarity value label widget.

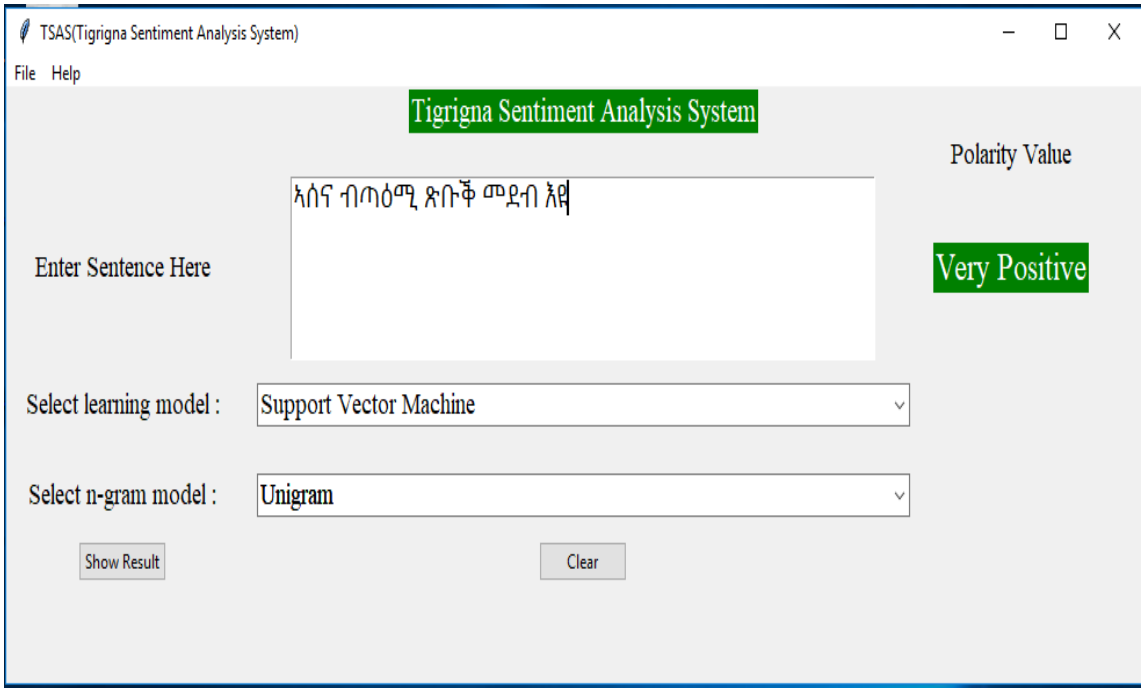

Figure 5-14 Prototype Demo I

Figure 5-14 also shows a given sample sentence and its classified polarity value. When “Show Result” button is clicked the input text goes through classification phase with accordance to the selected algorithm and language model. Finally, the system made prediction and result displayed on the “Polarity Value” label field. The result, which is displayed in the “polarity value” label field one of the five predefines sentiment classes namely very positive, positive, neutral, negative or very negative. The “Clear” button used to clear both the input text field and result displaying label filed.

Similarly, sample of the prototype that shows browsing Tigrigna sentence texts with a csv file format from file and their polarity classifications are presented in Figure 5-15. Large number of reviews can be collected manually or automatically and stored in csv formatted file. This large number of texts can be processed and classified at once. In this case, each sentence is processed and labeled with its polarity value and final statistical data that shows the given total number of sentences and number of classified polarity value for each polarity class is generated so that this data can be used for further analysis.

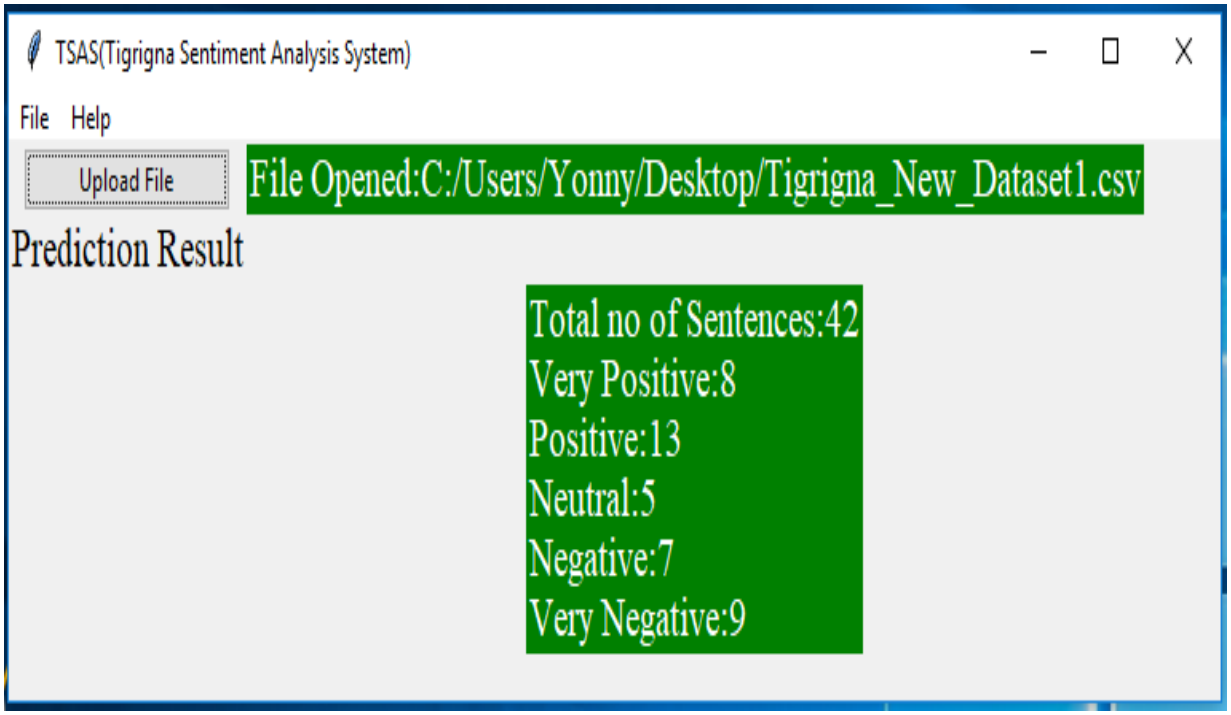

Figure 5-15 Prototype Demo II

5.5.1 User Acceptance Testing

Once the experimentations were conducted and the prototype was developed then, the user acceptance testing evaluation is carried out by the system's possible end-users to ensure that whether the performance of the system is accurate and the system is usable

by the end-users. Thus, in this study 5 users were selected and had been given the chance to use and interact with the system. The users selected were two YouTube channel owners and three activists (used Facebook extensively). Therefore, to analyze the system performance with user evaluations, the selected users put their values based on Likert scale such as Excellent = 5, Very Good =4, Good =3, Fair =2 and Poor =1. Thus, this method helps us to manually examine user acceptance based on the evaluator's response. The interview questions format for feedbacks of the possible end-users on system interactions are shown on APPENDIX K: User Acceptance Testing Evaluation Query. Different researchers have used different types of user acceptance testing evaluation criteria and for this study, the evaluation criteria are customized from (Alemu, 2019). The results of the end-user evaluation are summarized in Table 5-16.

No.	Criteria of Evaluation	Excellent	Very Good	Good	Fair	Poor	Average	Percentage
1	Simplicity of the system	3	2	0	0	0	4.6	92%
2	Efficiency and Effectiveness of the system	1	4	0	0	0	4.2	84%
3	Attractiveness of the system	0	3	2	1	0	3.8	76%
4	Accuracy of the system to classify a given text	3	1	1	0	0	4.4	88%
5	Importance of the system in the domain area	4	1	0	0	0	4.8	96%
6	Error tolerance of the system	0	2	2	1	0	3.2	64%
	Total Average						4.16	83.3%

Table 5-16 User acceptance testing evaluation results

As shown in Table 5-16, 60% of the evaluators assessed the prototype's simplicity as Excellent, 40% rated it as Very Good. In the second criteria of evaluation, the prototype effectiveness and efficiency 80% of the evaluators gave it a Very Good rating and 20% gave it an Excellent rating. In the third category, which is Attractiveness of the prototype, 60% of the evaluators gave it a Very Good rating, 40% gave it a Good rating, and 20% gave it a Good rating. In fourth criteria, 60% of respondents assessed its

accuracy to classify a given sentence with its evaluation criteria as Excellent, while 20% ranked it as Very Good and the rest 20% as a Good rating.

Regarding the importance of the developed prototype in the domain area, 80% of the evaluators gave it an Excellent rating, and 20% gave it a Very Good rating. The final evaluation criterion is error tolerance of the system, in which 60% of respondents assessed the system's capacity to tolerate errors as Very Good,40% of respondents as a Good and 20% of respondents ranked it as a Fair. Finally, according to the user's evaluation results, the prototype average performance is 4.16 out of 5. This result indicates that the Tigrigna Multiscale sentiment analysis prototype overall average performance is 83.3%, which is above Very Good.

5.6 Discussion of the Results

The main goal of this study is to design and develop a machine learning-based multiscale sentiment analysis of Tigrigna Texts. Sentiment analysis is a process of extracting and identifying information from users, to know feelings or opinions of intended group or individuals. In order to achieve the general objective of the study several experiments are conducted. The experiments were designed and conducted to provide an answer for the formulated research questions.

As a result, evaluation results shown in Figure 5-12 provide an answer to RQ1[What are the important features extracted from opinionated Tigrigna texts that have the greatest influence on sentiment analysis?], and the results show hybrid of unigram and bigram ,hybrid of unigram and trigram ,and unigram features are the important features extracted from the experiment that have a great influence on classifying Tigrigna sentiments.

Experimental results presented on the Figure 5-12 also answer RQ2[Which machine learning techniques perform better classification for morphologically rich language, Tigrigna, with respect to a preprocessed dataset?], SVM performs better than naïve Bayes and MaxEnt, with accuracy of 71.6% in unigram language model. Hence, SVM with unigram language model is selected for creating a model used for Tigrigna sentiment classification.

As summarized in Table 5-15, the conducted experiments in the second category answer RQ3[What is the effect of stemming on the Tigrigna language sentiment analysis?], stemming has a positive impact for bigram, hybrid of bigram and trigram, and trigram models on Tigrigna sentiment analysis for all employed learning models.

For SVM in hybrid of unigram and bigram, for NB in unigram and for MaxEnt in both unigram, and hybrid of unigram and bigram language models, it does not show any change. However, it showed a negative impact for naïve bayes in hybrid of unigram and trigram, and for SVM in unigram, and hybrid of unigram and trigram language models. The negative impact is because stemming removes affixes, and some of the affixes are indicative of sentiment and their removal leads to system under-performance.

5.6.1 Comparison of related works

In this section, we have compared our proposed study with the previous studies on Tigrigna sentiment analysis. As we discussed in section 2.10.4 , there are two studies conducted on Tigrigna sentiment analysis and the discussion has focused on the major findings of previous works to compare the findings of this study. The summary of the comparison with the previous studies is shown in the following Table 4.10. We have used a machine learning approach which is different than the previous researches conducted for Tigrigna sentiment analysis and have used relatively large amount of data comparing with the previous studies on this research and achieved encouraging results.

No.	Author	Title	Method	Accuracy
1	Mebrahtu Tadesse	Trilingual sentiment analysis on social media	Lexicon approach	87.49%
2	Nabyom Shishay	Designing sentiment analysis model for opinionated Tigrigna texts	Lexicon approach	not mentioned in accuracy
3	Our study	A machine learning approach to multiscale sentiment analysis of Tigrigna online posts	Machine learning approach	SVM-71.6% MaxEnt-69% NB-66%

Table 5-17 Comparison of related works

CHAPTER 6
CONCLUSION AND RECOMMENDATION

This chapter presents the conclusion and recommendation of the proposed multiscale Tigrigna sentiment analysis model. Section 6.1 discusses the conclusion determined from the research findings. Section 6.2 discusses contribution of this research work and Section 6.3 discusses recommendations for future researchers who are interested to precede in the same or related research area.

6.1 Conclusion

With this rapid invention and growth of web technologies, people, individuals and organizations are increasingly using public opinions in blogs, forums, wikis, review sites, social networks, and so on for expressing their views, opinions and for decision-making. This has changed the manner in which people communicate and influence social, political and economic behavior of other people and organizations. It has also dramatically changed the way people express their views and opinions on all kinds of entities such as products and services. These reviews are useful for service providers and manufactures to make informed decisions and improving their service.

However, the huge volume of reviews stored on the social media in the form of tweets, status updates, posts, comments, product reviews etc. grows so rapidly and becoming increasingly difficult for users to analyze and extract useful information. Therefore, it is important doing analysis on these sentiments to extract and identify the opinions of people regarding company, products and provide a relevant information for organizations which is crucial in making informed and right decision. An automated sentiment analysis is thus needed.

In this research, an attempt is made to apply sentence-level sentiment analysis on sport, education, politics, music and movie domains for Tigrigna online texts. Multi class classification model is constructed using Naïve Bayes, Maximum Entropy and Support Vector Machine algorithms to classify reviews as very positive, positive, neutral, negative and very negative. In this research work, we have collected 1500 sentences and conducted 36 different experiments categorized into two categories for evaluating the performances of the Tigrigna sentiment analysis and come up with a better performance.

SVM with unigram language model outperforms all algorithms with 71.6% accuracy. It showed a 1.6% improvement than MaxEnt, which have 69% accuracy, and 7.6%

improvement than naïve Bayes that have 64% accuracy in the first category of experiments. The second category of experiments evaluation results show that stemming has increased an accuracy of 42% to 47%,47% to 49%, and 47% to 49% in bigram model 39% to 42%,46% to 49%,46% to 49% in hybrid of bigram and trigram, and 21% to 23%,26% to 27%, and 26% to 27% in trigram model for Naïve Bayes, Maximum Entropy and SVM respectively. The results also show that stemming has no effect for Naïve Bayes and Maximum Entropy in unigram feature while it also reduced in unigram for SVM learning model.

As Pang et al.(2002) suggest hybrid model performs better than bigram; hybrid model performs better than bigram ,but experimental result of our study shows that the accuracy of SVM algorithm with a unigram model is higher than hybrid. As Pang & Lee (2008) reported it unigram feature is more effective for sentiment analysis, and the unigram feature performed better in our case too. In conclusion, despite the language's morphological complexity and lack of effective morphological analysis tools, the performance evaluation showed the study results are good and promising. Dealing with the manual subjectivity analysis and data collection is by itself is tedious and resource consuming work, as a result classification of texts as subjective and objective by avoiding manual identification of texts and automatically collecting sentences from their sources can reduce the huge effort to be devoted in building the needed sentiment corpus.

6.2 Contribution of the thesis

The main contributions of this paper can be summarized as follows.

- Proposed a model using machine learning approach to classify sentiment analysis of Tigrigna online posts with acceptable accuracy, precision and recall.
- A prototype based on the model is developed. and evaluated for effectiveness and encouraging results are obtained.
- The collected and prepared resources (corpus, stop words list etc.) can be used for training and evaluation purposes in future research works.
- The study can suggest works to be done on sentiment analysis related research works for sentimental Tigrigna texts.

To the best of our knowledge, these issues have not been addressed by existing approaches in the field of sentiment analysis for Tigrigna texts.

6.3 Recommendation

In this research, an attempt is made to design and develop multi-scale Tigrigna sentiment analysis model using a supervised machine learning approach. Although, the results obtained in this study are encouraging, further research and developmental effort is needed to have a full-fledged sentiment analysis for Tigrigna language. As a result, the following points should be considered and addressed in the future work.

- Development of standard publicly available sentiment analysis corpus for Tigrigna language.
- Efficient and accurate morphological analysis tools for lemmatizing and POS tagging tools that can handle all word categories because it is crucial in subjectivity detection and sentiment analysis.
- Combining lexical and machine learning, a hybrid approach, could improve the performance of sentiment classification.
- We used term weighting tfidf techniques for vectorizing the dataset before training in this study. Thus, future works should explore other approaches such as word embedding.

REFERENCES

Abbasi, A., France, S., Zhang, Z., & Chen, H. (2011). Selecting attributes for sentiment classification using feature relation networks. *IEEE Transactions on Knowledge and Data Engineering*, *23*(3), 447–462. https://doi.org/10.1109/TKDE.2010.110

Abdul-mageed, M., Diab, M., & Kübler, S. (2013). SAMAR : Subjectivity and sentiment analysis for Arabic social media ፎ. *Computer Speech & Language*, 1–18. https://doi.org/10.1016/j.csl.2013.03.001

Abdul-Mageed, M., Mona, T. D., & Mohammed, K. (2011). Subjectivity and Sentiment Analysis of Modern Standard Arabic and Arabic Microblogs. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, *3*, 587–591.

Abreham Getachew. (2014). *Opinion mining from Amharic Entertainment texts*. Unpublishd MSc Thesis, Department of Information Science, Addis Ababa University.

Alemu, T. D. (2019). *College of Computing Department of Information Systems By : Abunu Tesfaw*. Unpublished MSc Thesis, Department of Information Systems, Debre Birhan University.

Amanuel Sahle. (1998). *ሰዋስው ትግርኛ ብሰፊሑ*. The Red Sea Press Inc.

Assefa Gebrehiwot. (2011). *a Two-Step Approach for Tigrigna Text Categorization*. *June*, 98.

Atalay Leul. (2014). *Probabilistic information retreival system for tigrinya* (Issue June). Unpublishd MSc Thesis, Department of Information Science, Addis Ababa University.

Berger, A. L., Della Pietra, V. J., & Della Pietra, S. A. (1996). A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, *22*(1), 39–68.

Buche, A. (2013). Opinion Mining and Analysis: A Survey. *International Journal on Natural Language Computing*, *2*(3), 39–48. https://doi.org/10.5121/ijnlc.2013.2304

Chaturvedi, I., Poria, S., & Cambria, E. (2018). Sentiment Analysis, Basic Tasks of. *Encyclopedia of Social Network Analysis and Mining*, 2434–2454. https://doi.org/10.1007/978-1-4939-7131-2_110159

CSA Ethiopia. (2007). *Tigray_Statistical.pdf*.

Daniel Teklu. (2008). *Modern Tigrigna Grammar(ዘበናዊ ሰዋስው ቋንቋ ትግርኛ)*. Mega Publishing and Distribution PLC.

de Vries, M. (2017). *Machine Learning for Sentiment Analysis of Children's Diaries*. 108.

Espinosa, K. J., Troussas, C., Virvou, M., Llaguno, K., & Caro, J. (2013). Sentiment analysis of Facebook statuses using Naive Bayes Classifier for language learning. *IISA 2013 - 4th International Conference on Information, Intelligence, Systems and Applications*, *April 2014*, 198–205. https://doi.org/10.1109/IISA.2013.6623713

Gasser, M. (2011). HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya. *Conference on Human Language Technology for Development, April 2011*, 94–99.

Girma Berhe. (2006). Addis Ababa University. *Cahiers d'études Africaines*, *46*(182), 291–312. https://doi.org/10.4000/etudesafricaines.5928

Hailay Beyene. (2013). *design and development of Tigrigna search engine*. Unpublishd

MSc Thesis, Department of Computer Science, Addis Ababa University.

Hatzivassiloglou, V., & McKeown, K. R. (1997). *Predicting the semantic orientation of adjectives*. 174–181. https://doi.org/10.3115/979617.979640

Hevner, B. A. R., Esearch, S. Y. R., March, S. T., Park, J., & Ram, S. (2004). *design science in information systems*. *28*(1), 75–105.

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. *KDD-2004 - Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 168–177. https://doi.org/10.1145/1014052.1014073

Jindal, N., & Liu, B. (2006). Mining comparative sentences and relations. *Proceedings of the National Conference on Artificial Intelligence*, *2*(July 2006), 1331–1336.

Joachims, T. (1998). Text Categorization with Support Vector Machines. *Proceedings of the European Conference on Machine Learning*, *October 1999*, 137–142. https://doi.org/10.17877/DE290R-5097

John, S. M. (1996). *Tigrinya grammar*. The Red Sea Press Inc.

Lalji, T. K., & Deshmukh, S. N. (2016). Sentiment Analysis using Hybrid Approach. *International Research Journal of Engineering and Technology*, 1621–1627.

Liu, B. (2010). Sentiment analysis and subjectivity. *Handbook of Natural Language Processing, Second Edition*, 627–666.

Liu, B. (2012a). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, *5*(1), 1–184. https://doi.org/10.2200/S00416ED1V01Y201204HLT016

Liu, B. (2012b). *Sentiment Sentiment Analysis Analysis and and Opinion Opinion Mining Mining*.

Mebrahtu Tadesse. (2018). *Trilingual Sentiment Analysis on Social Media*. Unpublishd MSc Thesis, Department of Computer Science, Addis Ababa University.

Mountassir, A., Benbrahim, H., & Berrada, I. (2012). An empirical study to address the problem of unbalanced data sets in sentiment classification. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 3298–3303. https://doi.org/10.1109/ICSMC.2012.6378300

Mourad, A., & Darwish, K. (2013). Subjectivity and sentiment analysis of modern standard Arabic Microblogs. *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, *2*(June), 587–591.

Mulubrhan Hailegebreal. (2017). *A Bidirectional Tigrigna – English Statistical Machine Translation* (Issue October). Unpublished MSc Thesis, School of Information Science, Addis Ababa University.

Nabyom Shishay. (2018). *Designing sentiment analysis model for opinionated Tigrigna texts*. Unpublished MSc, Department of Information Technology, Jimma University.

Ominglot. (2021). *Tigrinya language, alphabet and pronunciation*. https://omniglot.com/writing/tigrinya.htm

Osman, I. O., & Mikami Yoshiki. (2012). Stemming Tigrinya Words for Information Retrieval. *Proceedings of COLInG 2012*, *1*(December), 345–352.

Pang, B., & Lee, L. (2002). A Sentimental Education : Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Pang, B., & Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *ACL-05 - 43rd Annual Meeting of the*

Association for Computational Linguistics, Proceedings of the Conference, *June*, 115–124. https://doi.org/10.3115/1219840.1219855

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends ® in Information Retrieval*, *2*(1), 1–135.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). *Thumbs up ? Sentiment Classification using Machine Learning Techniques*. *July*, 79–86.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Rothfels, J., & Tibshirani, J. (2010). Unsupervised sentiment classification of English movie reviews using automatic selection of positive and negative sentiment items. *CS224N-Final Project Report*, 52–56.

Saberi, B., & Saad, S. (2017). Sentiment analysis or opinion mining: A review. *International Journal on Advanced Science, Engineering and Information Technology*, *7*(5), 1660–1666. https://doi.org/10.18517/ijaseit.7.5.2137

Salunkhe, P., & Deshmukh, S. (2017). Twitter Based Election Prediction and Analysis. *International Research Journal of Engineering and Technology*, *4*(10), 539–544. https://www.irjet.net/archives/V4/i10/IRJET-V4I1094.pdf

Selama Gebremeskel. (2010). *sentiment mining for opinionated Amharic Text*. Unpublished MSc Thesis, Department of Computer Science, Addis Ababa University.

Shoukry, A., & Rafea, A. (2012). Sentence-level Arabic sentiment analysis. *Proceedings of the 2012 International Conference on Collaboration Technologies and Systems, CTS 2012, May 2014*, 546–550. https://doi.org/10.1109/CTS.2012.6261103

Singh, P. K., & Shahid Husain, M. (2014). Methodological Study Of Opinion Mining And Sentiment Analysis Techniques. *International Journal on Soft Computing*, *5*(1), 11–21. https://doi.org/10.5121/ijsc.2014.5102

Siqueira, H., & Barros, F. (2010). A Feature Extraction Process for Sentiment Analysis of Opinions on Services. *Proceedings of the III International Workshop on Web and Text Intelligence (WTI)*.

Somprasertsri, G., & Lalitrojwong, P. (2010). Mining feature-opinion in online customer reviews for opinion summarization. *Journal of Universal Computer Science*, *16*(6), 938–955.

Steven, B., Ewan, K., & Edward, L. (2009). *Natural Language Processing with Python:Analyzing text withe Natural Language Toolkit*.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-Based Methods for Sentiment Analysis DRAFT DRAFT DRAFT! *Computational Linguistics*, *37*(2), 267–307. http://www.sfu.ca/~mtaboada/docs/Taboada_etal_SO-CAL.pdf

Tedmori, S., & Awajan, A. (2019). Sentiment analysis main tasks and applications: A survey. *Journal of Information Processing Systems*, *15*(3), 500–519. https://doi.org/10.3745/JIPS.04.0120

Teklay Gebregabiher. (2010). *School of Graduate Studies Part of Speech Tagger for Tigrigna Language Part of Speech Tagger for Tigrigna*. Angeles, L., Advocacy, S., Location, O. (2002).

Tripathy, A., Agrawal, A., & Rath, S. K. (2015). Classification of Sentimental Reviews Using Machine Learning Techniques. *Procedia - Procedia Computer Science*, *57*,

821–829. https://doi.org/10.1016/j.procs.2015.07.523

Tromp, E. (2011). Multilingual sentiment analysis on social media. *Master's Theisis. Department of Mathematics and Computer Science, Eindhoven University of Technology*, *July*, 114. http://www.win.tue.nl/~mpechen/projects/pdfs/Tromp2011.pdf

Tulu Tilahun. (2013). *opinion mining from amharic blog*.

Turegn Fikre. (2020). Effect of preprocessing on Long Short Term Memory based sentiment analys for Amharic language. *Addis Ababa University*.

Turney, P. D. (2001). *Thumbs up or thumbs down?Semantic Orientation Applied to Unsupervised Classification of Reviews*. *July*, 417. https://doi.org/10.3115/1073083.1073153

Vaitheeswaran, G., & Dr. L., A. (2016). Combining Lexicon and Machine Learning Method to Enhance the Accuracy of Sentiment Analysis on Big Data. *International Journal of Computer Science and Information Technologies (IJCSIT)*, *7*(1), 306–311.

Wahbeh, A., Al-Kabi, M., Al-Radaideh, Q., Al-Shawakfa, E., & Alsmadi, I. (2011). The Effect of Stemming on Arabic Text Classification. *International Journal of Information Retrieval Research*, *1*(3), 54–70. https://doi.org/10.4018/ijirr.2011070104

Wiebe, J., & Mihalcea, R. (2005). *Word Sense and Subjectivity*.

Wondwossen Philemon, & Wondwossen Mulugeta. (2014). A Machine Learning Approach to Multi-Scale Sentiment Analysis of Amharic Online Posts. *HiLCoE Journal of Computer Science and Technology*, *2*(2).

Yadollahi, A. L. I., Shahraki, A. G., & Zaiane, O. R. (2017). *Current State of Text Sentiment Analysis from Opinion*. *50*(2).

Yonas Fisseha. (2011). *Development of stemming algorithm for Tigrigna text* (Issue June). Unpublished MSc Thesis, Department of Information Science, Addis Ababa University.

Zhuang, L., Jing, F., & Zhu, X. Y. (2006). Movie review mining and summarization. *International Conference on Information and Knowledge Management, Proceedings*, 43–50. https://doi.org/10.1145/1183614.1183625

Zitouni, I. (2014). *Natural language processing of semitic languages*.

APPENDICES

APPENDIX A: List of Tigrigna Stop-Words

ሁንዴሳ
ሂወት
ሃረሪ
ሃውስ
ሃይለ
ሃጫሉ
ሄሊኮፕተር
ህንዲ
ህወሓት
ህዝቢ
ህይወት
ህግድፍ
ሆትስፐርስ
ለሎ
ለተሰንበት
ለካቲት
ሊቨርፑል
ሊግ
ሎሚ
ሎማዕልቲ
ሎማዕንቲ
ሐዚ
ሑመራ
ሓወይ
ሓውና
ሓደ
ሓፍተይ
ሓፍትና
ሕንጣሎ
ሕዚ
ሕጂ
መለስ
መምህር
መቐለ
መን
መንግስቲ
መዓልቲ
መዓዝ
ሙሰሎኒ
ሚኒስቴር
ሚንስትሪ
ሚድያ
ማለተን
ማለቱ
ማለታ
ማለት
ማለትና
ማለትኩም
ማለትኪ
ማለትካ
ማለትክን
ማለቶም
ማርስ
ማዳጋስካር
ም
ምምሕዳር
ምሳና
ምሳኹም
ምሳኽን
ምስ
ምስቲ
ምስኡ
ምስኣ
ምስኣተን
ምስኣቶም
ምስአም
ምስዚ
ምእንቲ
ምኽንን
ምኽኑ
ምኽና
ምኽን
ምኽንና
ምኽንኪ
ምኽንካ
ምኽኖም
ምዕራብ
ሞሪንሆ
ሞዲ
ሩሲያ
ራኢና
ራያ
ርያን
ሰልጠነ
ሰንበት
ሱዳን
ሱፐር
ሲዳማ
ስለ
ስለዚ
ስለዝኾነ
ስለዝኾነት
ስለዝኾነውን
ስለዝኾኑ
ስለዝኾና
ስለዝኾንና
ስለዝኾንኩ
ስለዝኾንኩም
ስለዝኾንኪ
ስለዝኾንካ
ስለዝኾንክን
ስዑዲ
ስጋዕ
ስፖርት
ሶማሊያ
ሻምፒዮና
ሻምፒዮንስ
ሻዕቢያ
ሻዕብያ
ሸረ
ሸንግራዋ
ሸዋ
ቅድሚ
ቅድሚት
በረኸት
በቢ
በቲ
በቶም
በአአም
በዙይ
በዚ
ቢሮ
ባይቶ
ባይደን
ቤት
ብ
ብሊንከን
ብምኽነን
ብምኽኑ
ብምኽና
ብምኽንና
ብምኽንኩም
ብምኽንኪ
ብምኽንካ
ብምኽንክን
ብምኽኖም
ብርሃነ
ብርጋዴር
ብሮድካስት
ብቅድሚ
ብቅድሚት
ብቲ
ብአአም
ብዚ
ብዛዕባ
ብዝኾነ
ብድሕሪት
ቦሪስ
ቦርድ
ቦቢ
ቫይረስ
ተሓቢሩ
ተስፉ
ተስፋጽዮን
ተስፎም
ተከዘ
ተጋሩ
ቱሉ
ቴሌኮም
ቴዲ
ቴድሮስ
ትርከቡ
ትርከባ
ትርከብ
ትርፊ
ትእምት
ትግራይ
ቶተንሃም

ቶኒ
ቶኪዮ
ቻይና
ቻድ
ነበረ
ነበረት
ነበሩ
ነበራ
ነበርና
ነቱይ
ነቲ
ነታ
ነቶም
ነዚ
ነይረ
ነይረን
ነይሩ
ነይራ
ነይርወን
ነይሮም
ናሬንድራ
ናብ
ናቱ
ናታ
ናታተን
ናታትኩም
ናታትክን
ናታቶም
ናትና
ናትኩም
ናትክን
ናይ
ናይተን
ናይቱ
ናይቲ
ናይታ
ናይቶም
ናይና
ናይዘን
ናይዚ
ናይዛ
ናይዞም
ን
ንሕና
ንማለት
ንርከብ
ንሰን
ንሱ
ንሳ
ንሳተን
ንሳቶም
ንስካ
ንስኹም
ንስኹምስ
ንስኺ
ንስኺስ
ንስኻ
ንስኻስ
ንስኻትኩም
ንስኻትክን
ንስኽን
ንሶም
ንተን
ንቶም
ኡጋንዳ
ኢለ
ኢለምም
ኢለሞን
ኢለን
ኢለየን
ኢለያ
ኢለዮ
ኢለዮም
ኢሉ
ኢላ
ኢላተን
ኢላትኩም
ኢላቶ
ኢልና
ኢልናዮም
ኢልኩም
ኢልክን
ኢልዋ
ኢሎም
ኢመር
ኢስያስ
ኢትዮ
ኢትዮጵያ
ኢና
ኢንዳስትሪ
ኢኹም
ኢኺ
ኢኺን
ኢኻ
ኢየ
ኢየን
ኢዩ
ኢያ
ኢዮም
ኣለና
ኣለኹ
ኣለኹም
ኣለኺ
ኣለኻ
ኣለኽን
ኣለዉ
ኣሉላ
ኣላማጣ
ኣሕመድ
ኣመሪካ
ኣሜሪካ
ኣምሓራ
ኣምባሳደር
ኣስመራ
ኣስትራዘኒካ
ኣበይ
ኣብ
ኣብርሃ
ኣብቲ
ኣብኡ
ኣብዚ
ኣብዚኣ
ኣብዚኣም
ኣብዛ
ኣብዩ
ኣብይ
ኣቦና
ኣቱም
ኣቲ
ኣታ
ኣትሌቲክስ
ኣትሌት
ኣትን
ኣነ
ኣንሕና
ኣክሲዮን
ኣውሮፓ
ኣየን
ኣይሁድ
ኣይዶል
ኣዱ
ኣዱየ
ኣዲስ ኣበባ
ኣዴና
ኣፍሪቃ
ኣፍሪካ
ኤሌክትሪክ
ኤምባሲ
ኤርትራ
እለ
እለን
እሉ
እሊ
እላ
እላተን
እላትኩም
እልና
እልኩም
እልክን
እማ
እምበር
እምቢ
እሞ
እሰን
እሱ
እሳ
እስልምና
እስራኤል
እስራኤል
እስኪ
እሶም
እሺ
እባ
እተን
እቲ
እቲኣ
እታ
እቶም
እኒ

እና
እን
እንተሎ
እንተኮንኩም
እንተኪይንኪ
እንተኪይንካ
እንተኪይኖም
እንተኾነ
እንተዝኾና
እንተድኣ
እንታይ
እንትሽውን
እንትኾን
እንደርታ
እንደገና
እንድዩ
እንግሊዝ
እኮ
እኹም
እኺ
እኻ
እኽን
እወ
እዋን
እውን
እዙይ
እዚ
እዚአን
እዚኣ
እዚኣም
እዛ
እዞም
እየ
እየሱስ
እየን
እዩ
እያ
እያሱ
እያተን
እያቶም
እዮም
ኦሎምፒክ
ኦሮሚያ
ኦሮሞ
ከለና
ከለዋ
ከለው
ከምተን
ከምታ
ከምቶም
ከምአን
ከምኡውን
ከምኣተን
ከምኣቶም
ከምዘለና
ከምዘለኹም
ከምዘለኺ
ከምዘለኻ
ከምዘለኽን
ከምዙይ
ከምዚኣ
ከምዚኣተን
ከምዚኣቶም
ከነማ
ከኣ
ከከም
ከዓ
ኩለን
ኩሎም
ኩባኒያ
ኩክ
ኩፅሾ
ካሊእ
ካልእ
ካልኦት
ካብ
ካብተን
ካብታ
ካብቶም
ካብዚ
ካዓ
ከለብ
ክልል
ክርስትና
ክርስቶስ
ክሳብ
ክሳፅ
ክስታይ
ክስቶ
ክብሉ
ክብል
ክብሮም
ክንዲቲ
ክኣ
ክኾና
ኮሚሽን
ኮርያ
ኮሮና
ኮትዲቯር
ኮነ
ኮንፌዴሬሽን
ኮይነ
ኮይኑ
ኸዓ
ኻሊእ
ኻልእ
ኾነ
ወልቃይት
ወረዳ
ወርሒ
ወዘተ
ወያነ
ወይ
ወዲ
ወጀራት
ዋልድባ
ዋሽንግተን
ዋይን
ዌልስ
ውን
ዓለም
ዓመት
ዓረብ
ዓረና
ዓዲግራት
ዓጋመ
ዓፋር
ዘለኩም
ዘለኪ
ዘለካ
ዘላ
ዘሎ
ዘርኣቡሩኽ
ዘብረኣብሩኽ
ዘፀአት
ዝርከቡ
ዝርከባ
ዝርከብ
ዝበሃሉ
ዝባሃላ
ዝብሃላ
ዝኾነ
ዝኾነት
ዝኾኑ
ዝኾንና
ዝኾንኩም
ዝኾንኪ
ዝኾንካ
ዝኾንክን
ዞባ
የለን
የለዋን
የምፃኡ
የምፅኡ
የብለንን
የብለይን
የብሉን
የብልናን
የብልካን
የብልክን
የውሃንስ
ዩሮ
ዩንቨርስቲ
ዩኤል
ያኢ
ዬሴፍ
ይኹን
ዮሴፍ
ደራርቱ
ደርጊ
ደቀምሓረ
ደንጎላት
ደኣ
ደደቢት
ዲሲ
ዲፕሎማሲ
ዳኒ
ዳኒኤል
ዳንሻ

ዳይሬክተር	ጆሲ	ግርማይ	ፓርላማ
ዳደ	ጆሴ	ግደይ	ፕረዚደንት
ድሕሪ	ጆንሰን	ጎጃም	ፕሪሜርሊግ
ድማ	ገሬ	ጓል	ፕሬዚዳንት
ድኣ	ገብረ	ፋብሪካ	ፖለቲካ
ዶክተር	ገብረህይወት	ፋኦ	
ጀብሃ	ገብረመድህን	ፌስቡክ	
ጀነራል	ጊዮርጊስ	ፌደረሽን	
ጄኔራል	ጊግስ	ፌዴሬሽን	
ጅማ	ጋንታ	ፍልስጤም	

APPENDIX B: Short Words and their Expanded form List

ዶ.ር:ዶክተር	ዶ/ር:ዶክተር	ዓ.ም:ዓመተ ምሕረት
መ/ር:መምህር	ቤ/ፅሕፈት:ቤት ፅሕፈት	ቤ/ክርስትያን:ቤተ ክርስትያን
ቤ/ትምህርቲ:ቤት ትምህርቲ	ቤ/ፍርዲ:ቤት ፍርዲ	ገ/ህይወት:ገብረ ህይወት
ኢ/ያ:ኢትዮጵያ	ዝ/የ:ዝተፈላለየ	እ/ሄር:እግዚኣብሄር
ገ/ኪዳን :ገብረ ኪዳን	ቤት ት/ቲ:ቤት ትምህርቲ	ቤት ፍ/ዲ:ቤት ፍርዲ
ክፍለ ት/ቲ:ክፍለ ትምህርቲ	ሃ/ስላሴ:ሃይለስላሴ	መ/ር:መምህር
ወ/ር:ወታደር	ወ/ሮ:ወይዘሮ	ወ/ሪት:ወይዘሪት
ወ/ስላሰ:ወልደስላሰ	ፍ/ስላሴ:ፍቅረስላሴ	ፕ/ር:ፕሮፌሰር
ቀ.ሚንስትር:ቀዳማይ ሚኒስቴር	ቀ/ሚንስትር:ቀዳማይ ሚኒስቴር	ቀ.ሚንስትር:ቀዳማይ ሚኒስቴር
ቀ/ሚንስተር:ቀዳማይ ሚኒስቴር	ቀ.ሚ:ቀዳማይ ሚኒስቴር	ቀ/ሚ:ቀዳማይ ሚኒስቴር
ገ/ጊዮርጊስ:ገብረጊዮርጊስ	ም/ኣቦወንበር:ምክትል ኣቦወንበር	ቤ/ምኽሪ:ቤት ምኽሪ
ተ/ሃይማኖት:ተክለሃይማኖት	ሚ/ር:ሚኒስቴር	ሚኒስተር:ሚኒስቴር
ሚኒስትር:ሚኒስቴር	የሩሳሌም:ኢየሩሳሌም	ቴሌቭሽን:ቴሌቪሽን
ትቪ:ቴሌቪሽን	ቻምፒዮንስ:ሻምፒዮንስ	ፌዴሬሽን:ፌደረሽን
ኣይደል:ኣይዶል	ኮ/ል:ኮሌኔል	ሜ/ጄነራል:ሜጀር ጄነራል
ብ/ጄነራል:ብርጋዴር ጄነራል	ሌ/ኮለኔል :ሌቴናል ኮለኔል	ኣ/ኣ:ኣዲስ ኣበባ
ሓ/ማሕበር:ሓረስቶት ማሕበር	ደ.ኣንስትዮ:ደቂ ኣንስትዮ	ደ/ኣንስትዮ:ደቂ ኣንስትዮ
ገ/ልምዓት:ገጠር ልምዓት	ሕ.ወኪል:ሕርሻ ወኪል	ሕ/ወኪል:ሕርሻ ወኪል
ሓ.ዘመን:ሓዲሽ ዘመን	ሓ/ዘመን:ሓዲሽ ዘመን	ር/ምምሕዳር:ርእሰ ምምሕዳር
ማ/ሰብ:ማሕበረ ሰብ	ዓ.ዓ:ዓመተ ዓለም	ዓ/ዓ:ዓመተ ዓለም
ማ/ኮሚቴ:ማእኸላይ ኮሚቴ	ር/መምህር:ርእሰ መምህር	ፕ/ት:ፕሬዚዳንት
ሃ.ተፈጥሮ:ሃፍቲ ተፈጥሮ	ቤ/ፍትሒ:ቤት ፍትሒ	ሚ/ሕርሻ:ሚኒስቴር ሕርሻ
ቤ/ህንፀት:ቤት ህንፀት	ር/ከተማ:ርእሰ ከተማ	

APPENDIX C: List of Tigrigna Punctuation Marks

፡	ክልተ ነጥቢ	Space
፤	ድርብ ሰረዝ	Semi colon
፣	ነፀላ ሰረዝ	Comma
።	ኣርባዕተ ነጥቢ	Full stop
!	ትእምርተ ኣንክሮ	Exclamation Mark
፧	ሕቶ ምልክት	Question mark
፨		Paragraph separator

APPENDIX D: Sample of Cliticized Words

'ዩ: እዩ	'ዉን: እዉን	'ኳ: እኳ	'ዳ: እንዳ
'ታ:እታ	ንሱ'ዉን:ንሱ እዉን	‘ዮም፡ እዮም	‘የን፡ እየን

APPENDIX E: List of Normalized Characters

ኀ=ሀ	ጸ=ፀ	ሠ=ሰ
ኁ=ሁ	ጹ=ፁ	ሡ=ሱ
ኂ=ሂ	ጺ=ፂ	ሢ=ሲ
ኃ=ሀ	ጻ=ፃ	ሣ=ሳ
ኄ=ሄ	ጼ: ፄ	ሤ=ሴ
ኅ=ህ	ጽ=ፅ	ሥ=ስ
ኆ=ሆ	ጾ=ፆ	ሦ=ሶ

APPENDIX F: Sample of Strong Positive Reviews

1. ግደይ ፤እቲ ዝግጅትካ ብጣዕሚ ደስ ዝብል እዩ
2. ግደይ ሓወይ ብጣዕሚ እየ ዘድንቐካ
3. ግርም ሰንበት እንግዶት ምቁር ባህልና ዝያዳ ንዓቅብ መዓራት ድወት!!!♥♥♥
4. ጋዜጠኛ ኣኽበረት ገ/ስላሴ ኣቀራርባኺ ብሓቂ ዝንኣድ እዩ።
5. ገዛኢ ዓርከይ ኣብዚ ደረጃ ምብፃሕካ፤ ብጣዕሚ ደስ ኢሉኒ እዩ
6. ገዛኢ ሓወይ ናይ ብሓቂ ንፉዕ ኣርቲስት ኣጆኻ ቀጽሎ
7. ገሬ እሙን ብጣዕሚ ንፎትዋ ፊልም ኢያ።
8. ጆሲ ብሓቂ ብጽቡቐ ተንቲንካዮ
9. ጅግና ኣስተውዓሊ መንእሰይ ቀጽሎ
10. ጅግና ብጣዕሚ እየ ዝፈትዎ
11. ጅግና መንእሰይ ከማኻ የብዝሓዮ ኣለና ኣብ ሃንኻ ቐሰን
12. ጀይ ስቱድዮ፡ሓበሬታኻ ብጣዕሚ ጽቡቐ እዩ
13. ጀጋኑ ኣሕዋትና፡ ዬሴፍ ን ንስኻ ን ብጣዕሚ እየ ዝፈትወኩም
14. ድምፂ ወያነ ብጣዕሚ እየ ዝፈትወኩም ቀፅሉሉ።
15. ዳያኑ ብጣዕሚ ንፉዓት እኹም

16. ዳኒ ወዲ መምህር ኣዚኻ ብሉጽ መንእሰይ እኻ
17. ዳኒ ካፍቶም ዝምክሓሎም ንፉዕ ጅግና በሊሕ እኻ
18. ዳኒ እቲ ብሉጽ ጅግንነትካ ማራ 'ዩ ዘሐጉሰኒ!!
19. ዳኒ ብጣዕሚ ፅቡቕ ን ሰናይ ን ተግባር እዩ ።።
20. ዳያኑ ብጣዕሚ ትድነቕ ኢኹም
21. ደስ ዝብል ባህሊ ብጣዕሚ ንፍዕቲ ተዋዳዳሪት ቀጽልዮ
22. ናይ ብሓቂ ደስ ትብሉ ፈራዶ ቀጽልዎ
23. ደሃብ ብራቮ ብጣዕሚ ሃገራዊት ዜጋ ፤እዛ ሃገር ከማኺ ሰብ የድሊያ
24. ዮሴፍ: ብጣዕሚ ብዘገርም ኣገላልጻ ገሊጹዎ።
25. ዮሴፍ ገብርሂወት ዓቢዩ መምህር፤ናይ ብሓቂ ጽቡቕ ትንተና

APPENDIX G: Sample of Positive Reviews

1. ኣሰይ ወድሓፍቲ ፅሩይ ቀፅለሉ
2. ኣርያም ጭዋ ሰብ'ያ
3. ኣማኑኤል ሓርበኛ ተቃላሳይ ንዝርዶአስ ጽቡቕቲ ሓተታ እያ
4. ኣሕመድ ሓቢቢ ኬፍ ኣለካ ያዓኺ ኣጆኻ ወዲ ሳሆ ኣስሊ::
5. ኣሉላ ንፉዕ ተቓላሳይ እዩ
6. ኣሉላ ተባዕ ን ውፉይ ን እዩ
7. ኢትዮጵያ ትዓምብብ ጽቡቕ ስራሕ ይስራሕ ኣሎ::
8. ንፍዕቲ መደላዊት እኺ።
9. ንፉዕ እኻ መዓረይ ንመፃኢ ጎበዝ ደራፊ ናይ ትግራይና ከም እትኸውን ፍሉጥ እዩ
10. ንፉዕ ኢኻ ብባህላዊ ኣካዳድናኻ ባህላዊ መሳርሒ ተጠቂምካ ፅቡቕ ድምፂ ኣለካ
11. ንፉዓት ድምፃዉያን ኣብ ምፍራይ ን ኣብ ምፍላጥ ን ትርከቡ።
12. ንዳያኑ ክዓ ክብርን ምስጋናን ንዓኹም ሒዝና ኹሉ ግዜ ኩሩዐት ኢና
13. ንዕብየት ን ፅሬት ን ኩዕሶ ፅቡቕ ኣስተዋፅኦ ኣለዎ::
14. ንዕብየት ስፖርትን ጥበብ ን እዚ ትካል ፅቡቕ ኣንፈት ኣለዎ::
15. ንዕቤት ትግርኛ ፅቡቕ ኣስተዋፅኦ ትገብሩ ኣለኹም ቀፅሉሉ
16. ንዕላማ ቃልስና ዝድግፍ እምበር ዘዐንቅፍ ኣይኮነን።
17. ንዓይ ዘልዓሎ ሓሳብ የስማዕምዐኒ እዩ።
18. ንኹሉ ወዲ ሰብ ጠቕምቲ ሓሳባት ዘማእኸለ ጥበባዊ ኣቀራርባ እዩ
19. ንስኻ ለባም ሰብ ኢኻ
20. ናይ ፖለቲካ ትምበያኻ ዝገርም እዩ
21. ናይ ደጋፊ ድምፂ ምስምዖም ሓሪፍ እዩ::

22. ናይ ደያኑ ናይ ሓሳብ ፍልልይ ደስ ኢሉኒ ሓሳብ ክትፈላለ ባህርያዊ እዩ
23. ናይ ወጀራት ዘቐረብካዮ መፍትሒ ትኽክል ይመስለኒ
24. ናይ ኹሉ መሰረታዊ ፍታሕ ኣብ ትግራይ ጠንካራ መንግስቲ ንክህሉ ምስራሕ ጥራሕ እዩ።
25. ናይ እታ ትሓትት ዘላ ጋዜጠኛ ኣዘራርርባ ይምረጽ

APPENDIX H: Sample of Neutral Reviews

1. ፀወታ ፅዋዕ ቻምፒዮንስ ሊግ ናብ ፖርቶ ተዘዋዊሩ
2. ፋራ ማለት ዘዋፅኦ ዝፈልጥ ማለት እዩ
1. ፊቼ-ጨምበላላ ኣብ ኣዲስ ኣበባ ተኸቢሩ
2. ፀወታ ፅዋዕ ቻምፒዮንስ ሊግ ናብ ፖርቶ ተዘዋዊሩ
3. ጽባሕ ቀዳም ምሸታዊ ሙዚቃዊ ምርኢት ኣብ ማይ-ዓይኒ ትግራይ ተዳልዩ ኣሎ
4. ጣልያናዊ ኩባንያ ኢታካ ኣብ እንደርታ ንሓድሽ ፋብሪካ ዓለባ ኣመሪቑ
5. ጌታነህ ከበደ ቅድሚ ዕረፍቲ ንጋንታ ቅዱስ ጊዮርጊስ ሽዶ ኣመዝጊቡ፡፡
6. ጋዜጠኛ ፍፁም ብርሃነን ጋዜጠኛ ታምራት የማነን ኣብ ትሕቲ ቁፅፅር ከምዝወዓሉ ተገሊፁ።
7. ጋንታ ፋሲል ከነማ ተዓዋቲት ፅዋዕ ፕሪሜርሊግ ኢትዮጵያ ዓመተ 213 ዓ.ም ኮይና
8. ጋንታ ኩዕሾ እግሪ ቶተንሃም ሆትስፐርስ ንኣሰልጣኒ ጆሴ ሞሪንሆ ከምዘሰናበተቶም ኣፍሊጣ።
9. ጋንታ ኩዕሾ እግሪ መከላከያ ናብ ፕሪሜርሊግ ምምላሳ ኣረጋጊፃ
10. ጋንታ ኣርባምንጭ ከተማ ናብ ፕሪሜርሊግ ምምላሳ ኣረጋጊፃ።
11. ጋንታ ብሽክለታ ኤርትራውያን ትሕቲ 23፡ ኣብ ዓለም ቀዳመይቲ ወጺኣ
12. ጋንታ ስሑል ሽረ ናብ ፕሪምየር ሊግ ሓሊፉ
13. ጋንታ ሃዲያ ሆሳእና ብ19 ፀወታ 32 ነጥቢ ሒዛ ኣብ መበል ሳልሳይ ደረጃ ትርከብ፡፡
14. ጋናዊያን ፕሬዝዳንቶም ክመርፁ ውሒሎም
15. ጉጅለ ባህሊ ኢትዮጵያ ኣብ ስታድየም ከረን ሙዚቃዊ ምርኢት ከምዘቐረበት ተሓቢሩ
16. ገምጋም ኣፈፃፅማ ስራሕ ኣፕላይድ ሳይንስ ዩኒቨርሲቲታት ይሳለጥ ኣሎ
17. ጆሴ ሞሪንሆ ኣብ ጋንታ ቶትንሃም ሓደ ዓመትን ኣርባዕተ ወርሒን ኣሰልጢኖም ኢዮም፡፡
18. ጆ ባይደን ምስ ወራሲ ዓራት መሓመድ ቢን ዛይድ ምዝርራቡ ተገሊጹ
19. ጀኔራል ብርሃኑ ጁላ ምስ ኣምባሳደር እስራኤል ኣብ ኢትዮጵያ ተዛቲዮም
20. ጀፈሪ ፌልትማን ኣብ ቀርኒ ኣፍሪካ ፍሉይ ልኡኽ ዩናይትድ ስቴትስ ኮይኖም ተሸይሞም
21. ጀበርቲ መበቆሎም ካብ ትግራይ እዮም
22. ደዊሎም ኣረጋጊጾም ኤርትራዊ ኣይኮነን ትግራዋይ እዩ ሕጂ እውን
23. ዜጋታት ኣብ ሃገራዊ መረፃ ንቐሕ ተሳትፎ ክገብሩ ከምዝግባእ ቀ/ሚ ኣቢይ ገሊፆም

APPENDIX I: Sample of Negative Reviews

1. ስርዓት ላስመራ ጭፍራ ህግደፍ ናይ ጥፍኣት ን ብርስት ን ስርዓት እዩ
2. ሱዳን መርገጺኣም ላይእመንን፣ ግልብጥ ግልብጥ እዮም።
3. ሰናይ ምምሕድር ሞይታትኩም እያ
4. ሰነድ የብሉን ላሰራርሓኹም ክፍተት ላለዎ።
5. ሰልጠነ ፡ ላድጊ ሓሳዊ እዩ
6. ሰልጠነ ላጋይሽ ላምጺኡ ላነ ዝደልዮ ተዛረቡ ዝብል ላውራጃዊ እዩ
7. ሰልጠነ ሓሳዊ ዘረኛ እኻ
8. ሮዛ ሕብሩ ጫማ ገይሩ ሃለዉለዉ ይብል ላሎ።
9. ረግራግ ሃዳሚ ውኻርያ እንታይ ኮይኑ ደኣ ምስቶም ተጋሩ ዘይከተተ ሃዲሙ ዝመፅአ
10. ረሳሓት ሚድያ መዓልቲኹም ተጸበዩ
11. ምምሕዳር ኣሜሪካ ቀጠፍቲ እዮም።
12. ሚኪ ራያ ላብ ልዕሊ ዓፋር ዘለካ ርድኢት ጌጋ እዩ
13. መንግስቲ ጃም ይገብሮ ላሎ ንኽይስማዕ ዘሕዝን እዩ
14. መንግስቲ ላብይ ላሸባሪ እዩ።
15. መንእሰይ ኤርትራ፡ ንባዕዳዊ ላተሓሳስባ ጀብሃን ሻዕብያን ነጺግዎ እዩ።
16. መሪሕነት ህወሓት ውድቀት እምበር ክሳብ ሐዚ ዓወት የለን፡፡
17. መምህር መሓሪ ላብ ስዓቱ ኹሉ ተናጊሩ ነይሩ ግን ዝሰምዖ ላይረኽበን
18. መለስ በሊሕ ላእምሮ ዘለዎ ላብ ዓለም ታሪኽ ሰሪሑ ዝሓለፈ ጀግና ወዲ ህዝቢ እዩ
19. ሕማቕት ደያኑ እዮም ፤ብደንቢ ሃናፂ ርእቶ ላይህቡን
20. ሕሙም ፍጥረት እኻ!
21. ሕሊንኣም ብኽብዶም ሸይጦም እዮም
22. ሓደ እቶም ደያኑ ሕማቕ ላለዉ::
23. ሓንሳሎም መሸምሹ እዩ
24. ሓተላ ሰብ ኢልኻ ዲኻ ቆጸርካዮ ደንቆሮ
25. ሓስዩ እዩ ካልእ ተንኮል ላለዎ እንበር ትማሊ ካልእ ሎማዕንቲ ካልእ
26. ሓሳቦም ሕማቕ እዩ

APPENDIX J: Sample of Strong Negative Reviews

1. ብጣዕሚ እዩ ዘሐዝን ንስኻ ግን እንኳዕ ላይሞትካ፡
2. ብጣዕሚ ንጸልኣኩም ኢና
3. ብጣዕሚ ተሕፍሩ ኢኹም ንምንታይ ኢኹም ትኾልፍዎም::

4. ብጣዕሚ ብጣዕሚ ኣዝዩ መሕዘንን መገረምን ተግባር እዩ።
5. ብጣዕሚ ብዙሓት ብ ኣምሓሩ ዝተረሸኑ እውን ኣለዉ።
6. ብጣዕሚ መሕፈሪ ተግባር እዩ ዛሩጣት
7. ብጣዕሚ ሕማቅ ወረ ይስማዕ ኣሎ
8. ብጣዕሚ ልብኻ ዝሰብር ንግግር ኡፍፍፍ😢😢
9. ብጣዕሚ ልብኻ ዘቃጽል እዩ
10. ብዛዕባ እቲ ውልቀሰብ ምንም ኣፍልጦ የብለይን!
11. ብዛዕባ ኢትዮጵያ ንክናገር ሞራል የብሉን።
12. ብዙሕ ዘሰክፉ ጉድለታት ዘለዎ ቲም እዩ
13. ብዓይኒ ትግራይ እንተሪኢናዮ ብዙሕ ጌጋ ኣለዎ ።
14. ብወገን እቶም ተፃወቲ እንትረእ ኣብ ብጣዕሚ ሕማቕ ደረጃ ይርከቡ::
15. ብወገነይ እቲ ዛዕባ ምንም ጣዕሚ የብሉን
16. ብኹሉ ነገር ዓንዲ ሕቓኦም ሕምሽሽ ኢሉ ዝተረፎም የለን።
17. ብኽምዚ ዝመጽእ ምንም ለውጢ የለን
18. ብኩዕሾ ዝባኣስ መንእሰይ ትርጉም ን ዕላማ ን ስፖርት ዘይፈልጥ በሃም እዩ።
19. ብኣይ ወገን ብፍላይ ኣብ ደገፍቲ ዝረአየኒ ኹሉ ጎደሎ እዩ::
20. ብትዕቢት ተላዕጢጡ ልዕሊኡ ምሁር ከምዘየሎ እዩ ዝኣምን
21. ብትኽክል ብጣዕሚ ዘሕዝን እዩ
22. ብመዳይ ምስ ናይ ካለኦት ሃገራት ን ሃገራዊ ክለባት እንትርአ ብጣዕሚ ዝሞተ እዩ
23. ብመንጽር ናይ ትግርኛ ክወዳደሩ ከለዉ ግን ኣዝዮም ዉሑዳት እዮም።
24. ብመሰረት ግንዛበ እቲ ደጋፍ ኣዝዩ ትሑት እዩ:
25. ብሓቂ ብሃገር ደረጃ እንትረአ ኣብ ሕማቕ ኩነታት እዩ ዝርከብ።
26. ብሓቂ መቕለ ምንም ደስ ዝብል ጋንታ ኣይኮነን ዘሎ

APPENDIX K: User Acceptance Testing Evaluation Query

Dear Evaluator,

This evaluation form is prepared to evaluate to what extent the developed prototype is usable by the end-users in the domain area. Therefore, we kindly request you to evaluate the system by labeling the X symbol on the space provided for the corresponding criteria. in advance. Note: the values are rated as: Excellent=5, Very good =4, Good=3, Fair= 2 and Poor =1.

No.	Criteria of evaluation	Excellent	Very Good	Good	Fair	Poor
1	Simplicity of the system					
2	Efficiency and Effectiveness of the system					
3	Attractiveness of the system					
4	Accuracy of the system to classify a given text					
5	Importance of the system in the domain area					
6	Error tolerance of the system					

APPENDIX L: Tigrigna Alphabets

1st	2nd	3rd	4th	5th	6th	7th	Normalize				
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ					
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ	ሏ				
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ	ሗ				
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ	ሟ				
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ	ሧ				
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ	ሯ				
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ	ሷ				
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ	ሿ				
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ	ቈ	ቍ	ቋ	ቌ	ቍ
ቐ	ቑ	ቒ	ቓ	ቔ	ቕ	ቖ	ቘ	ቝ	ቛ	ቜ	ቝ
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ	ቧ				
ቨ	ቩ	ቪ	ቫ	ቬ	ቭ	ቮ	ቯ				
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ	ቷ				
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ	ቿ				
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ	ኈ	ኍ	ኋ	ኌ	
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ	ኗ				
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ	ኟ				
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ	ኧ				
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ	ኰ	ኵ	ኻ	ኼ	
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ	ዀ	ዅ	ዂ	ዄ	
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ					
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ					
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ	ዟ				
ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ	ዧ				
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ					
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ	ዷ				
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ	ጇ				
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ	ጐ	ጒ	ጓ		
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ	ጧ				
ጨ	ጩ	ጪ	ጫ	ጬ	ጭ	ጮ	ጯ				
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ	ጿ				
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ					
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ	ፏ				
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ	ፗ				

APPENDIX M: Tigrigna-English Transliteration Table

1st		2nd		3rd		4th		5th		6th		7th			
ሀ	he	ሁ	hu	ሂ	hi	ሃ	ha	ሄ	hE	ህ	h	ሆ	ho		
ለ	le	ሉ	hu	ሊ	li	ላ	la	ሌ	lE	ል	l	ሎ	lo	ሏ	lWa
ሐ	He	ሑ	Hu	ሒ	Hi	ሓ	Ha	ሔ	HE	ሕ	H	ሖ	Ho	ሗ	HWa
መ	me	ሙ	mu	ሚ	mi	ማ	ma	ሜ	mE	ም	m	ሞ	mo	ሟ	mWa
ሠ	‘se	ሡ	‘su	ሢ	‘si	ሣ	‘sa	ሤ	‘sE	ሥ	‘s	ሦ	‘so	ሧ	‘sWa
ረ	re	ሩ	ru	ሪ	ri	ራ	ra	ሬ	rE	ር	r	ሮ	ro	ሯ	rWa
ሰ	Se	ሱ	su	ሲ	si	ሳ	sa	ሴ	sE	ስ	s	ሶ	so	ሷ	sWa
ሸ	xe	ሹ	xu	ሺ	xi	ሻ	xa	ሼ	xE	ሽ	x	ሾ	xo	ሿ	xWa
ቀ	qe	ቁ	qu	ቂ	qi	ቃ	qa	ቄ	qE	ቅ	q	ቆ	qo		
ቐ	Qe	ቑ	Qu	ቒ	Qi	ቓ	Qa	ቔ	QE	ቕ	Q	ቖ	Qo		
በ	Be	ቡ	bu	ቢ	bi	ባ	ba	ቤ	bE	ብ	b	ቦ	bo	ቧ	bWa
ቨ	ve	ቩ	vu	ቪ	vi	ቫ	va	ቬ	vE	ቭ	v	ቮ	vo	ቯ	vWa
ተ	Te	ቱ	tu	ቲ	ti	ታ	ta	ቴ	tE	ት	t	ቶ	to	ቷ	tWa
ቸ	ce	ቹ	cu	ቺ	ci	ቻ	ca	ቼ	cE	ች	c	ቾ	co	ቿ	cWa
ኀ	‘he	ኁ	‘hu	ኂ	‘hi	ኃ	‘ha	ኄ	‘hE	ኅ	‘h	ኆ	‘ho		
ነ	ne	ኑ	nu	ኒ	ni	ና	na	ኔ	nE	ን	n	ኖ	no	ኗ	nWa
ኘ	Ne	ኙ	Nu	ኚ	Ni	ኛ	Na	ኜ	NE	ኝ	N	ኞ	No	ኟ	NWa
አ	A	ኡ	U	ኢ	I	ኣ	a	ኤ	EE	እ	i	ኦ	O		
ከ	ke	ኩ	ku	ኪ	ki	ካ	ka	ኬ	kE	ክ	k	ኮ	ko		
ኸ	Ke	ኹ	Ku	ኺ	Ki	ኻ	Ka	ኼ	KE	ኽ	K	ኾ	Ko		
ወ	we	ዉ	wu	ዊ	wi	ዋ	wa	ዌ	wE	ው	w	ዎ	Wo		
ዐ	‘A	ዑ	‘U	ዒ	‘I	ዓ	‘a	ዔ	‘EE	ዕ	‘i	ዖ	‘O		
ዘ	ze	ዙ	zu	ዚ	zi	ዛ	za	ዜ	zE	ዝ	z	ዞ	zo	ዟ	zWa
ዠ	Ze	ዡ	Zu	ዢ	Zi	ዣ	Za	ዤ	ZE	ዥ	Z	ዦ	Zo	ዧ	ZWa
የ	ye	ዩ	yu	ዪ	yi	ያ	ya	ዬ	YE	ይ	y	ዮ	yo		
ደ	de	ዱ	du	ዲ	di	ዳ	da	ዴ	dE	ድ	d	ዶ	do	ዷ	dWa
ጀ	Je	ጁ	ju	ጂ	ji	ጃ	ja	ጄ	jE	ጅ	j	ጆ	jo	ጇ	jWa
ጰ	Pe	ጱ	Pu	ጲ	Pi	ጳ	Pa	ጴ	PE	ጵ	P	ጶ	Po	ጷ	PWa
ገ	ge	ጉ	gu	ጊ	gi	ጋ	ga	ጌ	gE	ግ	g	ጎ	go		
ጠ	Te	ጡ	Tu	ጢ	Ti	ጣ	Ta	ጤ	TE	ጥ	T	ጦ	To	ጧ	TWa
ጨ	Ce	ጩ	Cu	ጪ	Ci	ጫ	Ca	ጬ	CE	ጭ	C	ጮ	Co	ጯ	CWa
ፀ	‘Se	ፁ	‘Su	ፂ	‘Si	ፃ	‘Sa	ፄ	‘SE	ፅ	‘S	ፆ	‘So		
ጸ	Se	ጹ	Su	ጺ	Si	ጻ	Sa	ጼ	SE	ጽ	S	ፆ	So	ጿ	SWa

ፈ	Fe	ፉ	fu	ፊ	fi	ፋ	fa	ፌ	fE	ፍ	f	ፎ	fo	ፏ	FWa
ፐ	pe	ፑ	pu	ፒ	pi	ፓ	pa	ፔ	pE	ፕ	p	ፖ	po	ፗ	Pwa
Liablized Characters															
ቈ	qWe			ቍ	qWi	ቋ	qWa	ቌ	qWE	ቍ	qW				
ቐ	QWe			ቑ	QWi	ቒ	QWa	ቓ	QWE	ቑ	QW				
ኰ	kWe			ኩ	kWi	ኳ	kWa	ኴ	kWE	ኵ	kW				
ኸ	KWe			ኹ	KWu	ኻ	Kwa	ኼ	KWE	ኽ	KW				
ጐ	gWe			ጒ	gWi	ጓ	gWa	ጔ	gWE	ጕ	gW				

APPENDIX N: Sample Code-I(Preprocessing)

```
dictionary = {'ጸ':'ፀ','ጹ':'ፁ','ጺ':'ፂ','ጻ':'ፃ','ጼ':'ፄ','ጽ':'ፅ','ጾ':'ፆ',
'ሠ':'ሰ','ሡ':'ሲ','ሢ':'ሲ','ሣ':'ሳ','ሤ':'ሴ','ሥ':'ስ','ሦ':'ሶ',
'ኅ':'ሀ','ኍ':'ሁ','ኂ':'ሂ','ኃ':'ሄ','ኌ':'ሄ','ኈ':'ህ','ኆ':'ሆ','ከዋ':'ካ','ግዋ':'ጓ','ቅዋ':'ቋ','ቐዋ':'ቒ',"\'":" እ",}
tsa.replace(dictionary, regex=True, inplace=True)
def identify_tokens(text):
    text= WordPunctTokenizer().tokenize(str(text))
    return text
def con_expansion(con):
    #con=row['Tokens']
    if type(con) is str:
        for key in contractionList:
            value=contractionList[key]
            con=con.replace(key,value)
        return con
    else:
        return con

tsa['Tokens']= tsa['Tokens'].apply(lambda con:con_expansion(con))
def remove_punctuations(text):
    punctuations = ['፡', '።', '፣', '፤', '፥', '፦', '፧', '፨']
    for pun in string.punctuation:
        punctuations.append(pun)
    for punctuation in punctuations:
        text = str(text).replace(punctuation,' ')
    return text

tsa['removed'] = tsa['Tokens'].apply(remove_punctuations)
def rejoin_words(row):
    my_list = row['removed']
```

```
        joined_words = ( "".join(my_list))
        return joined_words

tsa['Preprocessed'] = tsa.apply(rejoin_words, axis=1)
emoji_pattern = re.compile("["
                                u"\U0001F600-\U0001F64F"    # emoticons
                                u"\U0001F300-\U0001F5FF"    # symbols & pictographs
                                u"\U0001F680-\U0001F6FF"    # transport & map symbols
                                u"\U0001F1E0-\U0001F1FF"    # flags (iOS)
                                u"\U00002702-\U000027B0"
                                u"\U000024C2-\U0001F251"
                                "]+", flags=re.UNICODE)

def remove_emoji(string):
        return emoji_pattern.sub(r'', string)
tsa['Preprocessed'] = tsa['Preprocessed'].apply(remove_emoji) # Apply the remove_emoji
function to each row in the text column
tsa['Preprocessed'] =tsa['Preprocessed'].str.replace(r'\b\w\b','').str.replace(r'\s+', ' ')
tsa['Preprocessed'] = tsa['Preprocessed'].str.replace(r'[^\u1200-\u137F]', ' ')
tsa['Preprocessed'] = tsa['Preprocessed'].str.replace(' +',' ',regex=True)
tsa['Preprocessed']= tsa['Preprocessed'].str.lstrip()
Tigstopword = nltk.corpus.stopwords.words('Tigrigna')
tsa['Preprocessed'] = tsa['Preprocessed'].apply(lambda x: ' '.join([item for item in x.split() if item
not in Tigstopword]))
cols_to_drop = ['Sentence','Tokens','removed']
tsa.drop(cols_to_drop,axis=1, inplace=True)
tsa.to_csv('Preprocessed.csv', index=False)
```

APPENDIX O: Sample Code-II(Lemmatization)

```
def lemmatize_all_dataset():
        tsa['Processed_'] = tsa['Processed'].str.replace(' ', '_')
        with open('lemma_tsa_22.txt', 'w',encoding="utf8") as f:
                f.write(tsa['Processed_'].str.cat(sep='።'))
        hm.anal_file('ti', 'lemma_tsa_22.txt','lemma_tsa_output_22.txt',nbest=1)
        mystr='※'
        f = open('lemma_tsa_output_22.txt', 'r+',encoding="utf8")
        lines = f.readlines()
        mystr = ''.join([line.strip() for line in lines])
        mystr=mystr.split('።: ።')
        text_file = open("Lemmatized_tsa_Output_22.txt", "wt",encoding="utf8")
        text_file.write("Lemmatized\n")
```

```
    for sent_token in mystr:
        token=sent_token.split('_: _')
        for t in token:
            t=re.sub(r'<(.*?)>',r'', str(t))
            t=re.sub(r'\|(.*?)\:',r'', str(t))
            t=re.sub(r',',r'', str(t))
            re.sub(r'=(.*?)',r' ', str(t))
            t=re.sub(r'grammar(.*?)',r' ', str(t))
            t=re.sub(r'word:(.*?)citation:',r'x', str(t))
            t=re.sub(r'[^\u1200-\u137F]',r'', t)
            text_file.write(t)
            text_file.write(" ")
        text_file.write("\n")
    text_file.close()

    f = open('Lemmatized_tsa_Output_22.txt', 'r+',encoding="utf8")
    lines = f.readlines()
    mystr = '።'.join([line.strip() for line in lines])
    mystr=re.sub(r'።',r'\n', mystr)
    with open("Lemmatized_TSA_Output_Final.csv", "w",encoding="utf8") as f:
        f.write("".join(mystr))
    tsa2= pd.read_csv('Lemmatized_TSA_Output_Final.csv', sep=',',encoding="utf8")
    tsa['Lemmatized']=tsa2['Lemmatized']
```

APPENDIX P: Sample Code-III (Training Learning Model)

```
test_size=0.2
cv_counts=TfidfVectorizer(ngram_range=(1,1),analyzer='word')
X_counts=cv_counts.fit_transform(tsa.Processed).toarray()
X_counts.shape
X_train, X_test, y_train, y_test = train_test_split(X_counts, tsa.Polarity, test_size=test_size,shuffle=True, random_state=123,stratify=tsa.Polarity)
#X_test = cv_counts.transform(X_test)
clf_Multinomial=MultinomialNB()
clf_Multinomial.fit(X_train,y_train)
print('The Train score for Multinomial is {0}'.format(clf_Multinomial.score(X_train,y_train)))
print('The Test score for Multinomial is {0}'.format(clf_Multinomial.score(X_test,y_test)))

y_predict=clf_Multinomial.predict(X_test);

print(classification_report(y_test, y_predict))
```

```
print(confusion_matrix(y_test, y_predict))
sent1=clf_Multinomial.predict(cv_counts.transform([preprocess_sent("ኣቱም ሰባት ናይ ኤርትራ ጸገም ኣውራጃ ሃይማኖት ዓሌት ኣይኮነን።")]))
print(sent1)
print("Total Sentences:")
print(len(X_counts))
print("Total Trainig Sentences:")
print(len(X_train))
print("Total Testing Sentences:")
print(len(X_test))
```