2019-10

Public sentiment analysis for Amharic news

Dessalew, Chilote

BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF RESEARCH AND POSTGRADUATE STUDIES

FACULTY OF COMPUTING

PUBLIC SENTIMENT ANALYSIS FOR AMHARIC NEWS

By

CHILOTE DESSALEW MENGESHA

BAHIR DAR, ETHIOPIA

October 11, 2019

PUBLIC SENTIMENT ANALYSIS FOR AMHARIC NEWS

CHILOTE DESSALEW MENGESHA

THESIS SUBMITTED TO THE SCHOOL OF RESEARCH AND GRADUATE STUDIES OF BAHIR DAR INSTITUTE OF TECHNOLOGY, BAHIR DAR UNIV IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTERS IN THE INFORMATION TECHNOLOGY FACULTY OF COMPUTING

ADVISOR NAME: GEBEYEHU BELAY (Dr.of Eng)

Bahir Dar, Ethiopia

October 11, 2019

DECLARATION

I, the undersigned, announce that this proposition includes my own particular work. In consistence with globally acknowledged practices, I have recognized and refereed all material utilized as a part of this work.

Name of the student: Chilote Dessalew Signature _____

Date of submission: _____

Place: Bahir Dar

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

Advisor Name: Gebeyehu Belay(Dr.of Eng.)

Advisor's Signature: _____

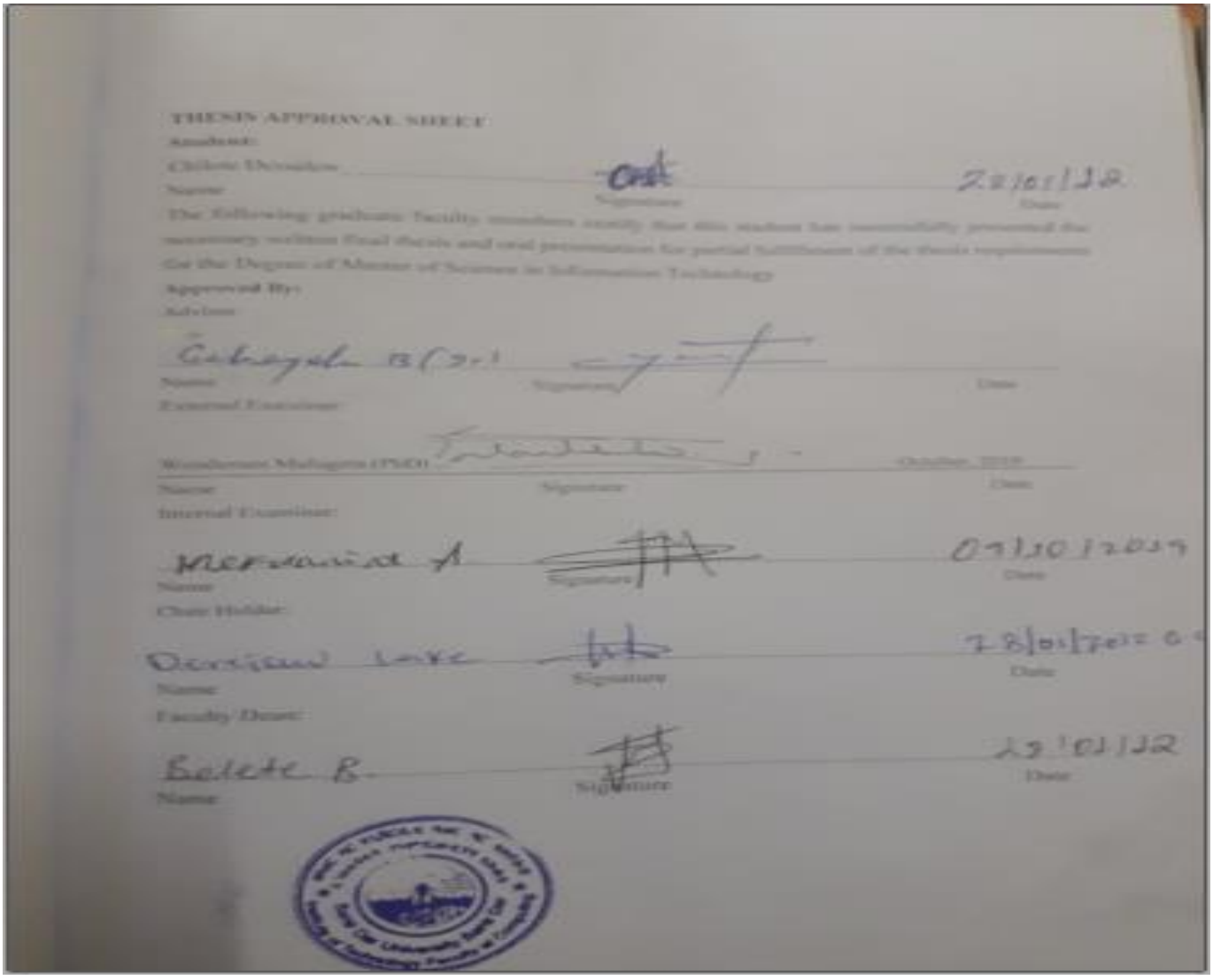
Bahir Dar University

Bahir Dar Institute of Technology

School of Research and Graduate Studies

Faculty of Computing

THESIS APPROVAL SHEET



ACKNOWLEDGMENTS

First and for most, I would like to express my deepest thanks to the Almighty God, for giving me the strength, determination, endurance and wisdom to bring this thesis to completion.

I would like to express my sincere gratitude to my thesis advisor Dr. Gebeyehu Belay for his constructive comments, suggestions, guidance, inspiring and enlightening ideas. Starting from shaping and reshaping the title of this thesis, his support and encouragements were high and I will not forget ever, the way you approach to me and warn me to finish the thesis work on time.

Last, but not least, I would like to thank my family and my friends for sharing me their knowledge and experience, and provided me with constructive comments and suggestions in bringing this thesis to an end.

Dedication

I would like to dedicate this thesis to my family.

Table of Contents

DECLARATION ii
ACKNOWLEDGMENTS iv
List of Figures x
List of Table xi
List of Abbreviations and Acronyms xii
Abstract xiii
Chapter One 1
1.1 Introduction 1
1.2 Statement of the problem 3
1.3 Objective 4
1.3.1 General Objective 4
1.3.2 Specific Objective 4
1.4 Methodology 5
1.4.1 Data collection 5
1.4.2 Design Approach 5
1.4.3 Tools and Techniques 5
1.4.4 Evaluation 6
1.5 Scope and Limitation of the Study 6
1.6 Significance of the Study 6
1.7 Organization of the Thesis 7
Chapter Two 8
Literature Review 8
2.1 Sentiment Analysis 8
2.2 Steps to Sentiment Analysis 9
2.3 Basic components of an opinion 9
2.3.1 Opinion words 10
2.3.2 Basic rules of opinions 10
2.4 Sentiment classification approaches 12
2.4.1 Document level 13
2.4.2 Sentence level 13
2.4.3 Feature or Aspect level 14
2.5 Steps in Sentiment Classification 15

2.6 Approaches and Techniques 15
2.7 Classification Techniques 16
2.7.1 Supervised Learning 16
2.8 Sentiment Analysis and Classification Techniques 17
2.8.1 Machine learning 17
2.8.2 Lexicon based technique 22
2.8.3 Rule-based Approach 23
2.8.4 Hybrid approach 24
2.9 Linguistic Issues in Sentiment Analysis 25
2.10 Amharic Language 25
2.10.1 Lexical Analysis of Amharic Language 27
2.11 Related Work 30
2.11.1 Sentiment Analysis Using Rule-Based Approach for the Amharic Language 30
2.11.2 Sentiment Analysis Using Machine Learning Approach for the Amharic Language 31
Chapter Three 35
Design and Experiment of Aspect/Feature Level Opinion Mining from Amharic Text 35
3.1 Flow of Architecture for sentiment analysis from opinionated Amharic texts using feature level 35
3.1.1 Data collection and preparation 37
3.1.2 Data preprocessing 38
3.1.3 Morphological Analyzer 39
3.1.4 Feature Extraction 40
3.1.5 Aspect-sentiment extraction (opinion word extraction) 42
3.1.7 Sentiment Classification 45
3.1.8 Aspect based polarity classification 46
Chapter Four 47
Experiment 47
4.1 Introduction 47
4.2 Development Environment and Tools 48
4.2.1 Tools 48
4.3 Data collection 48
4.4 Collecting aspect from the data 49
4.4.1 Determining polarity of opinion words and Summarization of aspect-opinions 50
4.6 Results and Discussion 52

4.6.1 Evaluation Procedures **52**
4.6.2 Evaluation Methods **52**
4.6.3 Cross-validation **53**
4.6.4 Confusion Matrix **54**
4.7 Result 55
4.7.1 Experiment using Naive Bayes **56**
4.7.2 Experiment using SVM **56**
4.8 Discussion 57
Conclusion and Recommendation 59
5.1 Conclusion 59
5.2 Recommendation 60
References 61

List of Figures

Figure 2. 1: Flow of Sentimental Analysis Architecture for different Level 16
Figure 2.2 : Supervised classification framework 17
Figure 2.3: data flow of SVM 21
Figure 3.1: The Architecture of sentiment analysis model for opinionated Amharic News……39
Figure 4.1: Sample data collection method using Facebook Graph API……49
Figure 4.2: Sample aspect - opinion Summary……..52

List of Table

Table 2.1: Rule-based Versus Statistical 24
Table 2.2: Shows an example of inflected Amharic adjective words 26
Table 2. 3: Overview of some previous work 32
Table 4.1: Confusion matrix 54
Table 4.2: Confusion Matrix for Amhara Mass Media Agency Facebook page news data set using NaiveBaye's……*56*
Table 4.3: Confusion Matrix for Amhara Mass Media Agency Facebook page news data set using SVM…….. 57
Table 4.4: Evaluation Results using Naïve baye's……58
Table 4.5: Evaluation Results using Naïve baye's……58

List of Abbreviations and Acronyms

ASCII	American Standard Code for Information Interchange
API	Application Programming Interface
dict	dictionaries
IDF	Inverted Document Frequency
IR	Information Retrieval
KNN	K-Nearest Neighbor
LR	Literature review
ME	Maximum Entropy
ML	Machine Learning
NB	Naive Bayes
Neg	Negative
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
POS	Part of Speech tag
Pos	Positive
R-SA	Rule - based Sentiment Analysis
SA	Sentiment Analysis
SVM	Support Vector Machine
TF	Term Frequency

Abstract

Sentiment analysis in the world allows us to extract useful information from the opinions and feelings of followers and customers of an organization. It analyzes the text written in natural language about services and products. Then after, classifying sentiments, opinions, feeling as positive, negative or neutral. However, Collectng data from Facebook of the Amharic News page is challenging. In addition to this, the data is huge and not easy to understand customers feeling and opinion. To solve this problem design Aspect level approach to mine the overall sentiment or opinion polarity from the data is needed.

We followed Aspect/Feature level opinion mining in detail to meet customers and organizations need. We employed crafted rules using rule-based for labeling data and supervised approach to training and testing the data. Research flow seven major components of the architecture which includes data collection, preprocessing of data, features extraction, Morphological Analyzer, aspect extraction, aggregate opinions sentiment classification, and result. Support Vector Machine (SVM) and Naive Bayes (NB) classifiers were used for sentiment classification. The collected Amharic opinionated sentences and phrase texts from Amhara Mass Media Facebook page were 1200. Among those, 960 data for training (80%) and 240 data for testing (20%). Experiments indicated that the bag of word module feature extraction methods performs the best through two algorithms (NB and SVM). The result showed as Naïve baye's precision, recall and F-measure evaluation metrics 84%, 80% and 81 % respectively. For SVM precision, recall and F-measure evaluation metrics are 87%, 82% and 84 % respectively.

Therefore, as to conclude, the SVM revealed the best category of the customers' sentiment and opinion, which gives a valuable approach for researchers and other users.

Keywords: Opinion mining, Sentiment analysis, Sentiment classification, Aspect extraction, Feature extraction, Sentiment polarity

Chapter One

1.1 Introduction

Natural language processing (NLP) is an application area in computer science, heavily supported by the industry with News applications emerging on a constant basis. The goal of this method is to give a different approach and look into natural language. It discovers basic concepts in computer science, machine learning, and statistics that make the natural language processing potential area of research. It studies to use general methods for application to specific problems that need to talk on how to make use of natural language. As such, we take a method-oriented view of NLP instead of an application-oriented one. For computers to examine, recognize, and originate meaning from human language in a clever and convenient way. By applying NLP, designers can establish and construct knowledge to perform tasks such as automatic summarization, translation, named entity recognition, relationship extraction, sentiment analysis, speech recognition, and topic segmentation [1-3].

Sentiment analysis is an important technique to identify the polarity of text opinion (positive, negative, neutral). A typical approach to sentiment analysis is to start with a lexicon of positive or negative words and phrases. In these lexicons, entries are tagged with their previous polarity. In information and technology age, a large amount of textual information is readily accessible which require systematic and efficient means of categorization [4]. Although much of the categorization works had been focusing on the basis of subject matter, the fast development of social media and blogs, which fundamentally express personal positive or negative opinions about a given topic required a different kind categorization. This News field of categorization is called Sentiment Analysis [5-8].

It is extensively applied for reviews like the movie, consumer, product acceptance. Moreover, sentiment analysis on social media is needed for the purpose of defining the attitude and perceptions of the writer/opinion holder with admiration to some topics or the

overall contextual polarity of the document. It controls the subjectivity of whether the expression is negative or positive and uses automated tools to detect the polarity expressions of opinion holders. In general, sentiment analysis is widely used in various fields to examine people's feelings and opinions on various application domains.

Polarity classification is concerned with categorizing a given opinionated document into predefined categories based on the weights obtained from the weight assign and polarity propagation process [9] of the given domain circumstance. For example, Mass Media. According to [10, 11], A large-scale sentiment analysis system for News and blog entities built on top of the Lydia text analysis system used to determine the public sentiment on each of the hundreds of thousands of entities that track, and how this sentiment varies with time. For people's sentiment polarization, we can have different approaches, which include document level [3, 9], sentence level [2, 12] and aspect or feature level [13, 14]

Document level: Classifies the whole document as positive, negative or neutral and commonly known as the document-level sentiment classification. It considers the whole document as a basic information unit (talking about one topic).

Sentence level: Classifies the sentences as positive, negative or neutral commonly known as sentence-level sentiment classification.

Aspect or Feature level: Classifies sentences/documents as positive, negative or neutral based on the aspects of those sentences/documents commonly known as aspect-level sentiment classification. Both the document level and the sentence level analyses do not discover what exactly people liked and did not like. In this paper/study, the **Aspect or Feature level** has been proposed.

A public, as opposed to a mass, can interpret its opinions into effective action. It can change policy as its opinions change. In a mass society, on the other hand, the most characteristic form of communication is a broadcast that delivers one unanswerable voice to millions of quiet and attentive listeners where there is little or no scope for individuals to answer back

to the messages they receive. There is certainly no way that the inhabitants of a mass society can translate their opinions into politically effective action.

1.2 Statement of the problem

Public opinions about their services and products, customers want to know the existing users' feeling or satisfaction before purchasing a product or using a service. "What other people think" has always been important for most of the people during their decision-making process. The attitude of the population to innovative development in News is one of the factors influencing technological, political, and economic factors. The amount and variety of information in Amhara Mass Media Facebook page for Amharic News are huge which made it impossible to manipulate manually and objectively judge overall user opinions and sentiments. Mining opinion from heterogeneous comments and turning it into usable knowledge is a challenging task. It is very difficult to find relevant information, extract important sentences, getting feedback, construct, summarize and organize idea into usable forms. In addition to that, it is hard to understand customers feeling and opinions and make decisions based on the unstructured data.

Many types of researches on sentiment analysis have been done using different languages, including English, Amharic and other languages like Arabic [15, 16] [13, 15]. Most researches were focused on document level and sentence level opinion mining system for the Amharic language. Despite the fact that a document level sentiment classification of Amharic documents fails to show parts that are liked or disliked by the customer. A positive document on an object does not mean that the customer has positive opinions on all aspects or features of the object. Likewise, a negative document does not mean that the customer dislikes everything about the object. In an evaluative document, a product review or service comment, the customer typically writes both positive and negative aspects of the object although the general sentiment on the object may be positive or negative. To achieve such detailed, Aspects/ Feature level opinion mining is needed. This motivated as is a need for sentiment analysis system and develop Aspect level sentiment mining model helps to analyze sentiment texts written in the Amharic language.

Research Questions

- ✓ How machine learning techniques are applied to classify with better classification model for mining opinionated Amharic news text?
- ✓ How the combination of rule based and machine learning approaches achieve competitive performance for aspect level sentiment analysis?
- ✓ Which features of language are appropriate for Amharic New texts on opinion mining task?

1.3 Objective

1.3.1 General Objective

The general objective of this research work is to build an Aspect level sentiment mining model for Amharic News.

1.3.2 Specific Objective

In order to achieve the objective of this research, various tasks are performed, which includes:

- ✓ To detect the features of user opinions in Amharic News.
- ✓ To analyze the dynamism of public sentiment for Amharic News context and broadcasting.
- ✓ To apply appropriate algorithms and classification approach to opinionated Amharic texts
- ✓ To test and evaluate the system

1.4 Methodology

1.4.1 Data collection

Sentiment sentences will be collected from the domain of Amharic news from Amhara Mass Media Agency Facebook page for this research work. The sentiment sentences collected using the Facebook Graph API from the topics posted on the Facebook page. In addition, preprocessing activities will be done on the collected sentiment sentences to enable morphological analysis and sentiment term detections.

1.4.2 Design Approach

To model aspect\feature level opinion mining from Amharic texts, the classification approach acquired into consideration to use rule-based approach and supervised methods that require the training set of texts assigned polarity values and, from these examples, they learn the features (e.g. words) that correlate with the value. Two supervised classifiers are implemented from the Natural Language Toolkit (the Naive Bayes and Support Vector Machine classifiers) the reason we selected naïve bayes and support vector machine classifier Support Vector Machine (SVM) is one of the majorly experimented data mining techniques in sentiment analysis [17]. SVM and Naïve Bayes has the greatest efficiency at text categorization when compared with other classification techniques like Maximum Entropy, KNN.

Create Lexicon of Amharic Opinion Words: Each word in this lexicon will be labelled by either positive (+) or negative (-) sign according to their basic polarity.

1.4.3 Tools and Techniques

Tools

Python programming language: - is employed because it has the capability to easily perform a search and replace operation over a large number of text files with enhanced error checking mechanism. Python is also suitable for the much larger application and

problem domain with its high-level built-in data types such as the flexible arrays and dictionaries.

1.4.4 Evaluation

The Experiment will be conducted to test the functionality sentiment analysis system. The performance of the system will be evaluated in terms of precision, recall and F-measure.

1.5 Scope and Limitation of the Study

The scope of the study is sentiment analysis of public News of the given media, which focused on opinion mining towards polarization of positive, negative or neutral classification. It doesn't cover subjective or objective classification. This research work considers only the inputted sentiment texts grammatically correct, analyzes and classifies sentiment texts written in Amharic language. And focus only sentiment analysis at Aspect level. Hence, grammatically incorrect texts, slang of words, sentiments expressed through an image/picture, an audio, video and other emotional symbols are not the concerns of this study. In addition to, words or texts that have indirect or hidden meanings such as an idiom are not incorporated in this research work. Domain-independence is one of the biggest problems in machine learning and classification. This system works for domain-specific only for Amharic News domain.

1.6 Significance of the Study

In the current business and political situations, knowing what other people think is a determinant factor for reasonable decision making. The significance behind this paper is to identify the polarity of a given text either positive, negative or neutral by collecting data from Amhara Mass Media Facebook pages about the News using Facebook Graph API. And will be able to automatically analyze the sentiment of huge amount of collected data prior to making decisions. In addition to that being an academic exercise to fulfill the requirement for the program, this study can help the Amhara Mass Media Agency organizations to improve their News services in the future.

The results of the research can be used as an input to the development of a full-fledged opinion make it in mining system for Amharic language or any other Ethiopian languages. Another important of the study is that the output can be used as an input data for recommender and opinion retrieval/search systems.

1.7 Organization of the Thesis

The remainder of this thesis is presented as follows. The Second chapter presents reviews made on different kinds of literature regarding opinion mining together with its approaches and different machine learning techniques. Chapter Three is about the overall methodology of the study, illustrates opinion mining techniques and algorithms which contain corpus preparation and preprocessing, system architecture, feature selection methods, classification techniques, and performance measurement. Chapter Four discusses the experimental result and discussion of the findings of how these experiments and methodologies are implemented. Finally, chapter Five describes with the conclusion and the feature work.

Chapter Two

Literature Review

In this section, review of existing literature on sentiment analysis, and the application of various techniques for sentiment analysis are done. A description is made employment of machine learning and statistical techniques to determine sentiment in social media opinion.

2.1 Sentiment Analysis

Sentiment analysis, also called opinion mining. It is the field of study 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 [18]. It represents a large problem space. There are also many names and slightly different tasks, e.g. sentiment analysis, opinion mining, opinion extraction, sentiment mining, subjectivity analysis, effect analysis, emotion analysis, review mining, etc. However, they are now all under the umbrella of sentiment analysis or opinion mining [12]. Opinion is referring to a person's ideas and thoughts towards something. It is an assessment, judgment or evaluation of something. Opinion has not been proven or verified. If it later becomes proven or verified, it is no longer an opinion, but a fact. Accordingly, all information on the web, from a surfer's perspective, is better described as opinion rather than fact [3].

Most commonly, it is referred to as the task of automatically determining the valence or polarity of a piece of text whether it is positive, negative, or neutral. However, more generally it refers to determining one's attitude towards a particular target or topic. Here attitude can mean an evaluative judgment, such as positive or negative, or an emotional or effectual attitude, such as frustration, joy, anger, sadness, excitement, and so on. Sentiment mining can play an important role in satisfying these needs. The process of sentiment mining involves classifying an opinionated document into predefined categories such as positive, negative or neutral based on the sentiment terms that appear within the opinionated document [5, 19-21].

Opinion mining, which is also known as sentiment analysis, emotion, attitude or subjectivity mining, which is a hot research discipline that is concerned with the computational study of attitudes, feelings and emotions articulated in an opinionated text. There are three types of opinion words which includes personal emotions, appreciation, and judicial opinions [2, 19]. However, understanding human sentiments is more challenging and complex as the data contexts [3, 18]. Sentiment prediction is formulated from huge scale collection of internet data via images as the context of image features and conceptual social network information as the huge of the supervised and unsupervised structure.

2.2 Steps to Sentiment Analysis

Sentiment analysis is a complex process that involves 5 different steps to analyze sentiment data. These steps are [19, 22]

1. Data collection was the first step used for collecting data from different sources.
2. Text preparation consists of cleaning the extracted data before analysis.
3. Sentiment detection is used to extracted sentences of the reviews and opinions are examined.
4. Sentences with subjective expressions (opinions, beliefs, and views) are retained and sentences with objective communication (facts, factual information) are discarded.
5. Sentiment classification in this step, subjective sentences are classified in positive, negative and neutral.

2.3 Basic components of an opinion

Opinion holder: The person or organization that holds a specific opinion on a particular object.

Object: Also called "Subject of Opinion" on which an opinion is expressed, a product, person, event, organization, topic or even an opinion.

Opinion: a view, attitude, or appraisal on an object from an opinion holder. An opinion contains often sentiment words which can be classified into polarities such as positive, negative, neutral [23].

2.3.1 Opinion words

Opinion (or sentiment) words, which are words that express positive or negative sentiments. Words that encode a desirable state (e.g., "great" and "good") have a positive polarity, while words that encode an undesirable state have a negative polarity (e.g., "bad" and "awful"). However, opinion polarity normally applies to adjectives and adverbs, there are verb and noun opinion words as well [24].

2.3.2 Basic rules of opinions

A rule of opinion is an implication with an expression on the left and an implied opinion on the right. The expression is a conceptual one as it represents a concept, which can be expressed in many ways in an actual sentence. The application of opinion words/phrases can also be represented as such rules. Let Neg be negative opinion word/phrase and Pos be positive opinion word/phrase. The rules for applying opinion words/phrases in a sentence are given as follows.

1. Neg ⟶ Negative
2. Pos ⟶ Positive

These rules say that Neg implies a negative opinion (denoted by Negative) and Pos implies a positive opinion (denoted by Positive) in a sentence. The effect of negations can be represented
as well:

3. Negation Neg ⟶ Positive
4. Negation Pos ⟶ Negative

These rules state that negated opinion words/phrases take their opposite orientation in a sentence. Other related rules are also outlined as follows. Deviation from the norm or some desired value change in some domains an object feature may have an expected or desired

value range or norm. If it is above and or below the normal range, it is negative e.g. “this drug causes low (or high) blood pressure”.

5. Desired value range ⟶ Positive
6 Below or above the desired value range ⟶ Negative

Decreased and increased quantities of opinionated items: this set rule is to the negation rules above. Decreasing or increasing the quantities associated with some opinionated items may change the orientation of the opinions. For example, “this drug reduced may pain rapidly and significantly.” Here pain is a negative opinion word, and the reduction of “pain” indicated a desired effect of the drug. Hence the decreased pain implies a positive opinion on the drug. The concept of decreasing also extends to “removal” or “disappearance”. e.g.” my pain has disappeared after taking the drug”.

7 Decreased Neg ⟶ Positive
8 Decreased Pos ⟶ Negative
9 Increased Neg ⟶ Negative
10 Increased Pos ⟶ Positive

The last rules may not be as such very important as there is no change of orientation. Producing and consuming resources and wastes: If an object produces resources, it is positive. If it consumes resources, especially a large quantity of them, it is negative. For example, “money” is a resource. The sentence, “Company-x charges a lot of money” gives a negative opinion on “Company x”. Likewise, if an object produces wastes, it is negative. If it consumes wastes, it is positive. These give us the following rules:

11. Consume resource → Negative
12 Produce resource → Positive
13 Consume waste → Positive
14 Produce waste → Negative

These basic rules can also be combined to produce compound rules, e.g., "Consumer decreased waste →Negative" which is a combination of rules 7 and 13. To build a practical system, all these rules and their combinations need to be considered. As noted above, these are conceptual rules. They can be expressed in many ways using different words and phrases in the actual text, and in different domains, they may also manifest differently. However, by no means, it is claimed these are the only basic rules that govern expressions of positive and negative opinions. With further research, additional News rules may be discovered and the current rules may be refined or revised. Neither it is claimed that any manifestation of such rules implies opinions in a sentence. Like opinion words and phrases, just because a rule is satisfied in a sentence does not mean that it actually is expressing an opinion, which makes sentiment analysis a very challenging task [6].

2.4 Sentiment classification approaches

Sentiment classification is one main task of opinion mining. Plenty of research publications have focused on sentiment classification. The approaches of sentiment classification can roughly fall into two basic categories. The methods in the first category rely on language resources which are constructed before carrying out a sentiment classification. The language resources include sentiment lexicons and natural language corpus libraries. Researchers usually use some natural language processing techniques combined with language resources to improve the accuracy of the sentiment classification. The methods in the second category try to employ machine learning to do sentiment classification [25].

Machine learning based sentiment classification methods contains supervised and semi-supervised methods which need some training instances to learn and get the final sentiment classifiers[26]. If we just simply apply sentiment classifier trained by instances coming from one domain on the other domain, it could be a hard task to get high classification accuracy because different domains might have different sentiment features or opinion words. It could be a hard task to label training instances for each domain. Though a semi-supervised-based method only requires a small number of training examples, it deeds useful information extracted from a huge number of unlabeled instances to boost the

opinion sentiment classification[27]. It is a task of classifying a target unit in a document to positive (favorable) or negative (unfavorable) classes. There are three main classification levels. These are: document level, sentence level, entity or aspect level [12] which are described as follows:

2.4.1 Document level

In this process sentiment is extracted from the entire review and a whole opinion is classified based on the overall sentiment of the opinion holder. The aim here is to determine the overall sentiment of an entire document, i.e. Given a product review, the task is to determine whether it expresses positive or negative opinions about the product. Example "I" bought an iPhone a few days ago. It is such a nice phone, although a little large. The touch screen is cool. The voice quality is clear too. I simply love it!" Is the review classification positive or negative? the Document-level classification works best when the document is written by a single person and expresses an opinion/sentiment on a single entity. This level looks at the document as a single entity, thus it is not extensible to multiple documents [13, 28]

2.4.2 Sentence level

This level of analysis is very close to subjectivity classification and the task at this level is limited to the sentences and their expressed opinions. Specifically, this level determines whether each sentence expresses a positive, negative or neutral opinion. This process usually involves two steps: subjectivity classification of a sentence into one of two classes: objective and subjective. Sentiment classification of subjective sentences into two classes: positive and negative an objective sentence presents some factual information, while a subjective sentence expresses personal feelings, views, emotions, or beliefs.

Subjective sentence identification can be achieved through different methods such as Naïve Bayesian classification [29]. However, just knowing that sentences have a positive or negative opinion is not sufficient. This is an intermediate step that helps filter out sentences with no opinions and helps determine to an extent if sentiments about entities and their

aspects are positive or negative. A subjective sentence may contain multiple opinions and subjective and factual clauses. For example, “iPhone sales are doing well in this bad economy. “Sentiment classification at both the document and sentence levels are useful, but they do not find what people like or dislike, nor do they identify opinion targets.

2.4.3 Feature or Aspect level

Aspect-based sentiment analysis is also known as a feature or attribute-based sentence analysis. It is used to analyze different features/attributes/aspects of the product. For example, smartphones can have different features like camera, battery life, touch screen etc. So, you analyze sentiments for these features for a given product. At last you would be generating tuple like [(product1, feature1, sentiment1), (product1, feature2, sentiment2), (product1, feature-n, sentiment-n) ...]. Later you can aggregated feature level sentiments on the basis of feature importance.

Both the document level and the sentence level analyses do not discover what exactly people liked and did not like. Aspect level performs the finer-grained analysis. It was earlier called feature level (feature-based opinion mining and summarization)[12, 13] Instead of looking at solely analyzing language constructs (documents, paragraphs, sentences, clauses or phrases), aspect level directly looks at the opinion itself. It is based on the idea that opinion consists of sentiment (positive or negative) and a target (of opinion).

An opinion without its target being identified is of limited use. Realizing the importance of opinion targets also helps us understand the sentiment analysis problem better. For example, although the sentence “although the service is not that great, I still love this restaurant” clearly has a positive tone, we cannot say that this sentence is entirely positive. In fact, the sentence is positive about the restaurant (emphasized) but negative about its service (not emphasized). In many applications, opinion targets are described by entities and/or their different aspects.

Both the document level and sentence level classifications are already highly challenging. The aspect-level is more challenging than both document and sentence levels and consists of several sub-problems. It finds different available sentiment analysis methods that could be categorized into two groups, language processing based and application-oriented methods [30, 31].

2.5 Steps in Sentiment Classification

With the aim of categorizing a given document into predefined categories, automatic text classification involves pre-processing and classification steps.

Preprocessing and Feature Extraction

Preprocessing of data plays an important role, both in terms of organizing it into a usable form, as well as reducing the number of features by removing redundant information [32]. In order to perform automatic text classification, the document must first be converted to an acceptable representation that can be used by the classifier. The pre-processing activity involves lexical analysis, normalization, stop-word removal, stemming, and index term selection. The input for this process is a text document, but not every word in the text is meaningful for categorization or retrieval. For this reason, documents must be processed and represented to a concise and identifiable format or structure. This will then bring the major benefit of data size reduction which increases performance in terms of memory size and processing time. The following subsection discusses each of them in detail [4].

2.6 Approaches and Techniques

The core objective of this research is to polarize public perception, opinion, feeling towards the media News services. The sentiment analysis system is a tool that helps to analyze texts written in Amharic language. To achieve this objective, the model design of sentiment analysis system, the components of the sentiment analysis system and the algorithms used for sentiment term detection and polarity weight determination in sentences are discussed in this chapter Sentimental analysis architecture, and the model developed as Figure 2.1.

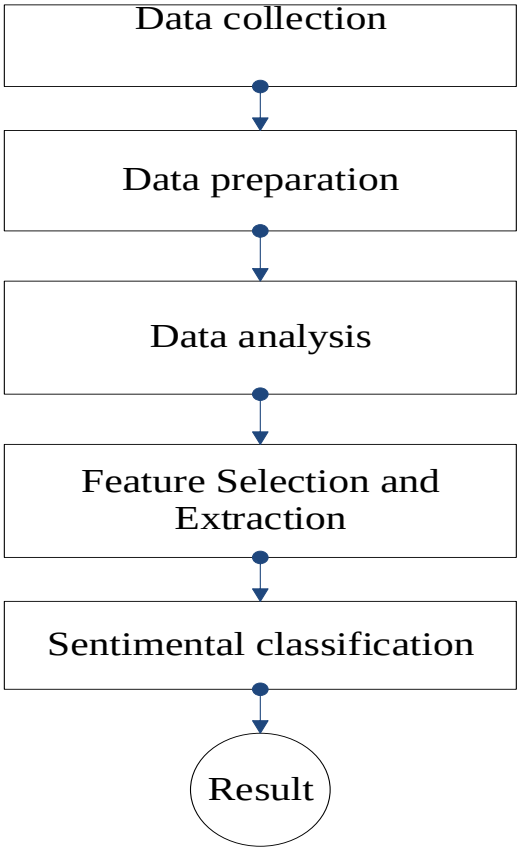


Figure 2. 1: Flow of Sentimental Analysis Architecture for different Level

2.7 Classification Techniques

Machine learning usually distinguishes between three learning methods: supervised weakly, supervised and unsupervised learning.

2.7.1 Supervised Learning

Supervised machine learning techniques involve the use of a labeled training corpus to learn a certain classification function and involve learning a function from examples of its inputs and outputs [8]. The output of this function is either a continuous value (‘regression’) or can predict a category or label of the input object (‘classification’). A classifier is called supervised if it is built based on training corpora containing the correct label for each input. The framework used by supervised classification is shown below in Figure 2.2.

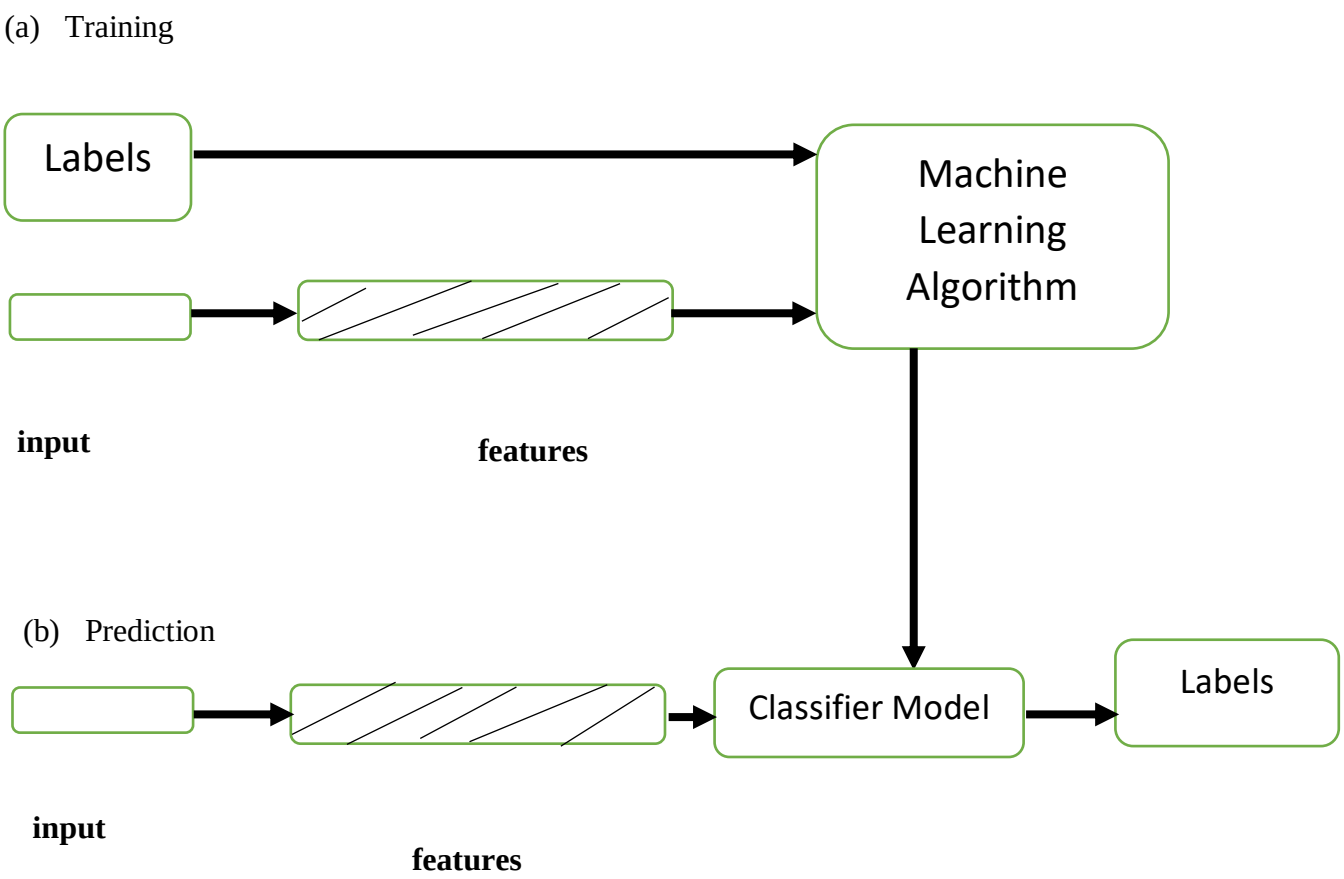


Figure 2.2 : Supervised classification framework

During training, feature extractor is used to convert each input value to a feature set. These feature sets which capture the basic information about each input that should be used to classify it, are discussed in the next section. Pairs of feature sets and labels are fed into the machine learning algorithm to generate a model. During prediction, the same feature extractor is used to convert unseen inputs to feature sets. These feature sets are then fed into the model, which generates predicted labels[1].

2.8 Sentiment Analysis and Classification Techniques

The sentiment classification approaches can be classified in (i) machine learning (ii) lexicon based and (iii) hybrid approaches[7, 19, 33].

2.8.1 Machine learning

Machine learning (ML) is a subfield of artificial intelligence dealing with algorithms that allow computers to learn. This usually means that an algorithm is given a set of data and

subsequently infers information about the properties of the data; that information allows it to make predictions about other data that it might come across in the future[8, 34, 35]. It is one of the most prominent techniques gaining the interest of researchers due to its adaptability and accuracy. In sentiment analysis, mostly the supervised learning variants of this technique are employed. It comprises of five stages: data collection, pre-processing, training data, classification and plotting results. In the training data, a collection of tagged corpora is provided.

ML is usually divided into the supervised and unsupervised approach. Supervised ML approach has a pre-defined or large amount of trained data set rules, but in unsupervised ML approach don't have any trained data set that's why it's difficult to find the level of trained dictionary rules in data set. In machine learning, first, train the algorithm with some known data rules before applying it to an actual dataset. In machine learning the algorithm is by the supervised or unsupervised method. The Supervised ML have the following algorithms Classifier, Linear Classifier (it can use Support Vector Machine and neural networks rule-based , the Probabilistic classifier (it can use Naïve Bayes, Bayesian Network) [36].

The classifier has presented a series of feature vectors from the previous data. A model is created based on the training data set which is employed over the News/unseen text for classification purpose. In the machine learning technique, the key to the accuracy of a classifier is the selection of appropriate features. Generally, unigrams (single word phrases), bi-grams (two consecutive phrases), tri-grams (three consecutive phrases) are selected as feature vectors. There are a variety of proposed features namely the number of positive words, number of negative words, and length of the document, Support Vector Machines (SVM) and Naive Bayes(NB) algorithm [4, 37]. The machine learning approach is used for predicting the polarity of sentiments based on trained as well as test data sets. Unigrams (single word phrases), bi-grams (two consecutive phrases), tri-grams (three consecutive phrases) are selected as feature vectors. The machine learning approach is used for predicting the polarity of sentiments based on trained as well as test data sets.

Naive Bayes (NB) is a simple but effective learning & classification algorithm. It is mostly used in text classification. The classification method is based on the theory of probability. It plays a vital role in probabilistic classification. It is also used in a statistical method for classification and supervised learning method. When Bayesian classifiers applied to large databases, it has exhibited high accuracy. Naive Bayes classification method is easy to implement. It requires only a small set of practical training data to judge a standard quantity which satisfies a particular set of equations. In most of the cases, good results are acquired through this classification method[21, 34].

Bayes theorem is defined as;

$$P(c|d) = \frac{P(d|c)\,P(c)}{P(d)} \quad \text{.....(2.1)}$$

d is review and c is class. For a given textual review d and for a class c (positive, negative), the conditional probability for each class given a review is P(c|d) [8, 38].

K-Nearest Neighbor (KNN) is the simplest algorithm of all machine learning algorithms. It is also referred to as Lazy Learning, Case-based Reasoning or Memory-based Reasoning. KNN is simple, it is yet able to solve the most complicated problems. It is a non-parametric method used for classification [21].

Support Vector Machines

Support Vector Machines (SVM) have been shown to be highly effective at traditional text categorization, generally outperforming Naive Bayes. They are large-margin, rather than probabilistic, classifiers, in contrast to Naive Bayes and Max ent. In the two-category case, the basic idea behind the training procedure is to find a hyperplane, represented by vector ~w, that not only separates the document vectors in one class from those in the other, but for which the separation, or margin, is as large as possible. This search corresponds to a constrained optimization problem; letting cj 2 [39](corresponding to positive and negative) be the correct class of document dj, the solution can be written as~w:=Xj®jcj ~ dj , ®j ¸ 0,where the ®j 's are obtained by solving a dual optimization problem. Those ~ dj such

that ®j is greater than zero are called support vectors since they are the only document vectors contributing to ~w. Classification of test instances consists simply of determining which side of ~w's hyperplane they fall on [22, 38].

SVM is often considered as the classifier that makes the greatest accuracy outcomes in text classification issues. They function by building a hyperplane with maximum Euclidean range to the nearest exercising cases. Basically put, SVM signifies cases as factors in the area which are planned to a high-dimensional area where the planned cases of individual sessions are separated by as large as a possible peripheral range to the hyperplane. News cases are planned into that same area, and based on which part of the hyperplane they are placed, they are expected to fit in with a certain category. SVM hyperplanes are completely established by a relatively small part of the training circumstances, which are known as the support vectors. The relaxations of the exercising data have no impact on the qualified classifier. SVM has been applied efficiently in text classification and in a large range of series handling programs [35].

HOW SVM WORKS

The idea for SVM is to find a boundary (known as a hyperplane) or boundaries that separate clusters of data. SVM does this by taking a set of points and separating those points using mathematical formulas as it is shown in Figure 2.3 [39].

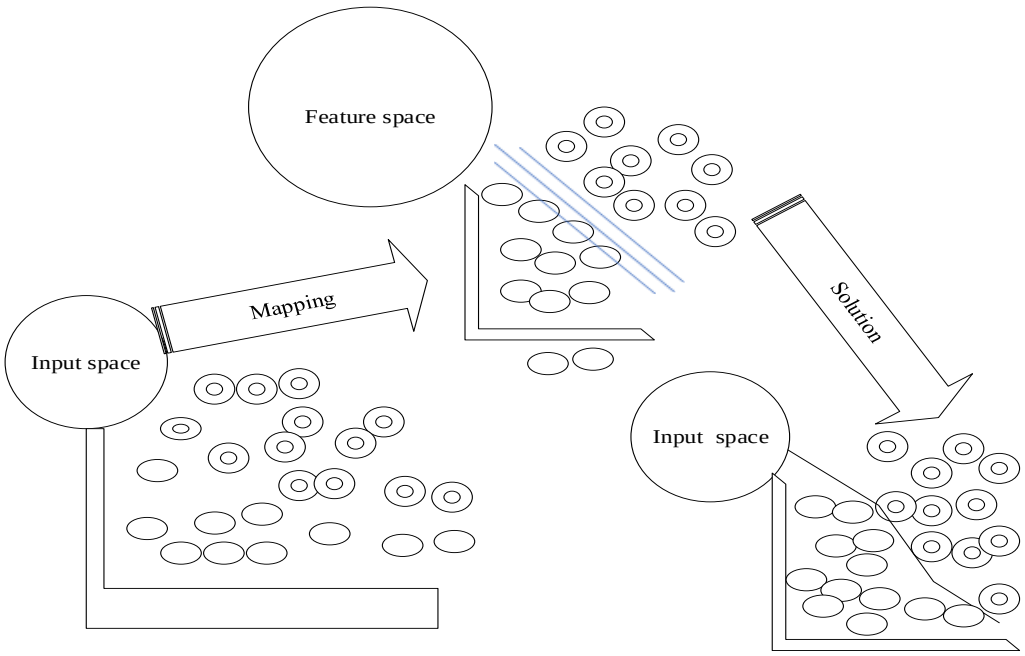


Figure 2.3: data flow of SVM

Maximum Entropy

Maximum Entropy (ME) classification is yet another technique, which has proven effective in a number of natural languages processing applications. Sometimes, it outperforms Naive Bayes at standard text classification. Its estimate of P(c |d) takes the exponential form as in [40].

$$PME(c/d) = \frac{1}{Z(d)} \exp(\sum i \lambda i,cFi,c(d,c)) \quad \text{.....(2.2)}$$

Where Z (d) is a normalization function. Fi, c is a feature/class function for feature fi and class c,

$$Fi,c(d,c') = \begin{cases} 1 & ni(d)>0 \text{ and } c'=c \\ 0 & \text{otherwise} \end{cases} \quad \text{..... (2.3)}$$

For instance, a particular feature/class function might fire if and only if the bigram “still hate” appears and the document’s sentiment is hypothesized to be negative. Importantly,

unlike Naive Bayes, maximum entropy makes no assumptions about the relationships between features and so might potentially perform better when conditional independence assumptions are not met.

Unsupervised method

supervised methods cannot always be used, because labeled corpora are not always available. Unsupervised and weakly-supervised methods are another option for machine learning that does not require pre-tagged data. Unsupervised involve learning patterns in the input when no specific output values are supplied [64], this means that the learner only receives an unlabeled set of examples. Unsupervised methods can also be used to label a corpus that can later be used for supervised learning.

Examples of unsupervised learning methods are (k-means) clustering or cluster analysis and the expectation-maximization algorithm, an algorithm for finding the maximum likelihood.**.** Weakly-supervised learning involves learning a classification task from a small set of labeled data and a large pool of unlabeled data [65].

2.8.2 Lexicon based technique

Lexicon based approach is an unsupervised technique. In this approach no need to maintain a large amount of training data set and rules which makes the whole process much faster. Lexicon approach is divided into dictionary-based and corpus-based to analyze the sentiment polarity [41]. On the other hand, it depends heavily on linguistic resources including a sentiment lexicon composed of pairs of words and its polarity values. Since particular words exhibit polarity values, it is genuinely essential to construct sentiment lexicon data meticulously [42]. This technique is governed by the use of a dictionary consisting of pre-tagged lexicons.

The input text is converted to tokens by the tokenizers. Every News token encountered is then matched for the lexicon in the dictionary. If there is a positive match, the score is added to the total pool of score for the input text. For instance, if “dramatic” is a positive match in the dictionary, then the total score of the text is incremented. Otherwise, the score is decremented or the word is tagged as negative. Though this technique appears to be amateur in nature, its variants have proved to be worthy[7, 43, 44].

2.8.2.1 Dictionary Based approach

The dictionary-Based approach involves using a dictionary which contains synonyms and antonyms of a word. Thus, a simple technique in this approach is to use a few seed sentiment words to bootstrap based on the synonym and antonym structure of a dictionary [33]. The main strategy of dictionary based approach is that it's working on manually creating a set of opinions that are repeated and then find, their synonyms and antonyms by iterations and save these words in the seed list. These iterations repeat until no synonyms and antonyms are found. After that, manually remove and correct errors. The limitation of dictionary approaches is its low applicability.

2.8.2.2 Corpus-Based approach

Corpus approach improves the limitations and helps to improve the finding of opinion in a particular area or orientation. Limitation of lexicon approach is that it cannot show high quality result in a big amount of data, such that to analyze the movie review comments that are posted online. This can't analyze well but this approach is good for small review data set like Facebook post comments or tweets [33] . The dictionary used may be WordNet or Sent WordNet or other. The corpus-based method helps to find opinion word in a context specific orientation start with a list of opinion word and then find another opinion word in a huge corpus [45].

2.8.3 Rule-based Approach

Rules based approach involves a list of positive and negative words the presence of which defines whether a sentence is positive or negative. This is very limiting because some times the same word can be pos or neg depending on the context. With machine learning, train an algorithm to understand sentiment based on many examples but then the model can predict sentiment in completely News sentences considering the whole sentence i.e. the context. The rule-based sentiment analysis technique consists of three main phases: 1) product feature extraction rule learning, 2)opinion sentence extraction, and 3) opinion orientation identification [46-48].

According to [49] Rule-based approach to sentiment analysis allowed deep analysis of the opinion content of the review, meaning can not only find the general sentiment of the review, but also a separate, sentiment of each clause. This provides sufficient information to single out separate positive and negative points about an entity or event. Thus, such analysis gives more information than the general sentiment of the review or the rate of a certain organization or person in social networks. Also the rule-based approach makes it possible to later compute the sentiment of the sentences with relative clauses and get more precise sentiment information.

2.8.4 Hybrid approach

It's a combination of machine learning and lexicon-based approaches. The hybrid technique is a sentiment lexicon constructed using product reviews for initial sentiment analysis. These sentiment analysis reviews are features in the machine learning method. A hybrid approach is much faster than both of two approaches. In this approach, the sentiment symbol detection is fasts and detection of sentiment is measured at a conceptual level and lesser sensitivity to change the domain. The only limitation in this approach its noisy reviews [4, 45].

Both rule-based and statistical approaches have their own pros and cons (Table 1). Thus, rule based and statistical approaches are usually combined to benefit from their synergy effect that rises to hybrid approach. Even supervised and unsupervised learning techniques that were mentioned under rule based and statistical approach have their own pros and cons. For instance, according to supervised machine learning is likely to provide more accurate classification result than unsupervised semantic orientation but a supervised machine learning model is tuned to the training corpus, and thus needs retraining if it is to be applied elsewhere.

Table 2.1: Rule-based Versus Statistical

Rule-based Approach	**Statistical Approach**
Requires linguistic expertise	Not much linguistics expertise required
Training dataset is not required	Large and ideal training dataset required
No frequency information	Based on frequency information
More brittle and slower	Robust and quick
Often more precise	Generalized model built from corpora
Error analysis is usually easier	Error analysis is often difficult

2.9 Linguistic Issues in Sentiment Analysis

Language is the ability to acquire and use complex system of communication. It is the human system of communication that uses arbitrary signals, such as voice, sound, gesture and/or written symbols that enable humans to express their feeling, sentiment, thought, idea and experience [50]. Natural languages like English, Amharic and Tigrigna are used for communication purposes, whereas these languages have their own linguistic behaviors. The linguistic behaviors include phonology, morphology, syntax, semantics and pragmatics.

In linguistics, morphology involved in word formation, deals with words, their internal structure and how they are formed with the help of morphemes. One fundamental computational task for a language analysis of its morphology is to derive the root and grammatical properties of the word based on its internal structures. The analysis of word structure performs using the process of inflection and derivation. In general, morphological analysis plays the vital role in sentiment analysis applications [51].

2.10 Amharic Language

Ethiopia is a linguistically diverse country where more than 80 languages are used in day-to-day communication. Although many languages are spoken in Ethiopia, Amharic is dominant in that it is spoken as a mother tongue by a substantial segment of the population and it is the most commonly learned second language throughout the country. The language is the official language of the federal government of the country. Amharic is the first language of more than 17 million people and a second language for more than 5 million people [54]. The present writing system of Amharic is taken from Ge'ez. Ge'ez in turn, took its script from the ancient Arabian language mainly attested in inscriptions in the Sabean dialect. Amharic did not discriminate in adopting the Ge'ez Fidel; it took all of the symbols and added some News ones that represent sounds not found in Ge'ez. These added alphabetic characters are □, □, □, □, □, □, □, and □ [54].

In the Amharic language, there are no common ways of writing words especially loan or foreign words that are taken from other languages. Hence, usage of foreign language words or loan words in Amharic (transliteration) is also found to be another source of word

spelling variation. The cause of the difference in the Amharic spellings of these foreign language words seems to be the difference in the pronunciations of these words. For example, the word computer can be written as □□□□□□ or □□□□□□. likewise, there are word spelling variations that could be attributed to variations in pronunciations at different parts of the country, for example using the two words □□□ and □□□ to mean temperament or using the three words □□□□, □□□□ and □□□□ to mean beetle. Therefore, such different kinds of compound word writings can lead machine learning algorithms to perform worse and should be detected in opinion mining or sentiment classification task [55].

Nouns: Amharic nouns are inflected for number, gender, person and cases and the result of the inflected word is with affixes. The inflected noun helps to express pluralism, gender and possession. Morphemes like -□□, -□□, -□□, -□ are used to inflect Amharic nouns by attaching to the stem words. For Example, aadding the suffixes -□□, -□□ and -□□ to the words □□, □□□□ and □□□ generates □□□, □□□□□ and □□□□ respectively [52].

Adjective: Amharic adjectives similar affixation process like Amharic nouns can be marked for number, gender, definiteness, cases and results of the inflected word with affixes. In Amharic, morphemes that used to inflect Amharic adjective words are -□/^o, -□□/^it and - □□/^oc.

Table 2.2: Shows an example of inflected Amharic adjective words

No	Adjectives	Male	Female	Pluralism
1	□□	□□	□□□	□□□
2	□□□	□□□	□□□□	□□□□
3	□□	□□	□□□	□□□

Amharic word classes can be derived from other Amharic word classes. For example, nouns can be derived from adjectives, verb roots and stems (by inserting vowels between consonants) and nouns themselves. For instance, □□□□, □□□□, □□□□ (sewnet, ljnet, bEtnet) are derived from the nouns □□, □□ and □□ (Sew, lj, bEt) respectively by adding

the suffix -□□/ net. In a similar way, verbs are also derived from different verbs by adding affixes in many ways. Amharic adjectives are also derived from nouns, verbal roots and compound words by adding affixes. For instance, the adjectives □□□, □□□□ and □□□□□ (qimeNa, heyleNa and gulbeteNa) are derived from the nouns □□, □□□ and □□□□ (qim, heyl and gulbet) respectively by adding the suffix – □□/`oNa.

Verbs: Amharic verbs are inflected for any combination of person, gender, number, case, tense, aspect and mood. As a result, from single Amharic verb thousands of Amharic verbs (in surface forms) can be generated. In Amharic, the verbs are found in different forms such as perfective, imperfective, gerundive, jussive and imperative by using affixes. The morphological inflection of the perfective verbs has suffixes such as -□/a, -□/k, -□/ku, -□/h, -□/u, -□□/uc, -□/x, -□/n and serves for the expression of 18-person, gender and number to the perfect verb stem. For example, the perfect verb stem □□□/seber (breaking) generates the inflected words such as □□□, □□□□, □□□□, □□□□, □□□□□, □□□□, □□□□ (seber, seberk, seberku, seberh, seberachu, sebersh and sebern). We can also add prefix like □/ye, □/be, □/ke, □□□/ind/ in the perfective verbs regardless of the gender, number and person [52].

2.10.1 Lexical Analysis of Amharic Language

The first task in the lexical analysis is tokenization, which is the process of converting the stream of characters in a document into tokens or list of terms, where a term is defined as a string of letters, digits or other special characters, separated by punctuation marks and spaces.

In addition to tokenization, the lexical analysis includes data cleaning, which is the process of removing characters that have no meaning in the dictionary. In a given document there could be words, which are composed of letters and digits. It is the task of cleaning for removing words of such type. The cleaning avoids errors which are against the syntactical rules of the language.

In order for a document processing activity to be error free, each character must be valid and the terms should be constructed following appropriate syntactical rules of the language. As the first step of enforcing the validity of characters and terms, dictionaries of the language should be used. After lexical analysis, it is normalization of homophones that follows. Amharic writing system has homophone characters. For example, it is common that the character □ and □ are used interchangeably as □□ and □□ to mean "work", The word "sun" can be written as, □□□, □□□, □□□, □□□, etc … all mean the same, although they are written differently [56]. Such type of inconsistency in writing words will be handled by replacing characters of the same sound with a common symbol. These characters cause an unnecessary increase in the number of document representative words that causes large data size processing.

2.10.1.1 Stop-word removal

The next process after lexical analysis and normalization is stop-word removal. Stop-words are words that occur most frequently in documents, but are not relevant or have no impact to discriminate among documents. The common words in English such as of, a, and the, are stop words and such words are not used to discriminate the documents. Such frequently used words generally, "glue" sentences together but they usually do not carry meanings.

The most frequently used stop-words in Amharic documents are □□, □□, □□, and □□□. In order to remove stop words, a list of stop-words should be identified and listed. A stop-word list is a list of such non-content bearing words that have to be removed during pre-processing of the document.

This process is normally done automatically by comparing words in the input text with words in a 'stop word list'. From the point of sentiment, removing such words from the documents is the best way to avoid retrieving unnecessary documents that are not relevant to satisfy the user's need. From the point of categorization, applying stop-word removal also reduces the complexity of the document representation and the number of tokens to be processed.

2.10.1.2 Stemming

Words that appear in a document often have many morphological variations. In most Cases, morphological variations have similar interpretation and can be considered as equivalent for the purpose of IR applications. The process of stemming is an attempt to reduce a word to its stem or root form. For example, stemming will bring the different forms of the word □□ "House" (□□,□□□, □□□□, □□□□□, □□□□, □□□□□) into their stem word □□. Thus, terms of a document are represented by stem words rather than by the original words. This also reduces the number of different terms needed for representing a document and also saves storage space and processing time. There are a number of stemming algorithms, such as table lookup approach, successor variety, n-gram stemmers, and affix removal. The study presented a stemmer for processing documents and query words which facilitate searching Amharic databases. So far, tokenization, normalization, stop-word removal, and stemming have been done on the documents. After these processes, to classify documents automatically, the documents should further be processed in order to be represented using their representative words.

A document is represented using a collection of index terms or words. An index term is simply any term or word that appears in a document. Commonly, an index term is a word which has its own meaning and is capable of representing a document or reflecting the content of the text. Such a selection follows lexical analysis, normalization, stop-word removal and stemming. Out of the list of terms, selection of representative words will be made based on the criteria of selecting a document. There are different techniques of selection of index terms to represent a document.

Among the most widely used techniques to select document, representative terms are statistical techniques. Statistical techniques for text analysis are based on Term Frequency (TF) and Inverted Document Frequency (IDF). TF is the number of times a term occurs in a given document. It is a frequency of words in the document to determine which words were sufficiently significant to represent the document. It is based on the principle that a term which is frequently used in a document is useful to represent the document. In short,

the frequency shows the usefulness of the term in the document, and it is formally stated as follows.

Term frequency TF (d, t), is the number of times a term t occurs in document d and is defined as Where di is the ith document, tk is the kth term of document di and Tj is the jth term in the document. IDF is the occurrence of a term in the collection of all input documents; if a word occurs in all documents, the relevance of the document will decrease because the probability of the word to represent the document is less. That is, terms that appear in many documents are not very useful as they do not allow discriminating between documents. A formal definition of IDF is [18, 53]

$$Tf(di,tk) = \sum f\ (Tj)\ j=1$$

Where df(t) is the number of documents including the term t, N the number of all documents. It is possible to use either term frequency or inverted term frequency depending on the application. However, term frequency is more appropriate for this study because the study considers a single document at a time instead of a collection of documents to categorize. For this study, all the pre-processing activities are adopted from a previous study by [53] . Each of the adopted components is described in Chapter 3 of this thesis.

2.11 Related Work

2.11.1 Sentiment Analysis Using Rule-Based Approach for the Amharic Language

According to [57] feature level opinion mining model for Amharic texts, the objective of the study was to determine an opinion on features of the domains. In this study, the author first extracted features of the domain and then determined the opinions in the extracted features by employing some rules. An opinion word in the sentence was detected from Amharic general-purpose opinion word lexicons that contain a total of 1001 sentiment words which are 578 negative and 423 positive words. The author collected 484 Amharic reviews manually from the hotel, university and hospital for experimental activities. The effectiveness of the system was evaluated using precision, recall and F-measure metrics through two different experiments. From experiment one, the author got the result of an average precision of 95.2%, an average recall of 26.1% for feature extraction and an

average precision of 78.1%, average recall of 66.8% for opinion word determination. From experiment two, an average precision of 79.8%, an average recall of 34% for feature extraction and average precision of 80%, an average recall of 92.7% for opinion word determination. The strength of this study was to determine an opinion on features of the domain in the review sentence. However, the author used only adjectives as sentiment words to determine the opinions of the review sentences, but sentiment words are not only adjectives but also include adverbs, verbs, and nouns. In addition to this, the sentiment words are not sufficient.

2.11.2 Sentiment Analysis Using Machine Learning Approach for the Amharic Language

[5] studies sentiment analysis for Amharic Language Multi-scale sentiment analysis model for Amharic online posts written in Ethiopic scripts using a supervised machine learning approach. The objective of the study was to determine the multi-scale sentiment sentence based on the polarity weight value of sentiment words. To achieve the objective, the author has prepared a sample corpus which contains 608 posts. The corpus was collected from social media sources such as Facebook, Twitter, Dire Tube, and Ethiopian reporter websites.

The researcher employed preprocessing activities in the corpus dataset before the actual sentiment classification. After the preprocessing activities were done, the corpus was manually annotated by giving polarity values and sentiment intensity scale values. Adopted two scale schemes which are scale positive sentiment further as +1 27 and +2 for less and more positive respectively and negative sentiment polarities as -1 and -2 for less and more negative respectively, hence neutral sentences were annotated as 0. To distinct the polarity strength of Amharic sentiment words, the author limited into five polarity rank scales. Naive Bayes algorithm was used to classify sentiment texts based on the trained on the entire corpus. The algorithm used n-gram features to acquire the knowledge from the training corpus, and then the algorithm classifies the sentiments of the test posts based on the acquired knowledge. Among the sample corpus, 486 posts were selected for the training dataset and 122 posts were for the testing dataset. The author achieved an accuracy of

43.6%, 44.3% and 39.3% for unigram, bigram, and hybrid language models respectively. However, the accuracy of the system was very low since the training dataset was very insufficient and the tool used for morphological analysis was not effective.

Studied by [58] Sentiment analysis for classifying Amharic opinionated text in to positive, negative or neutral by using ML approach in ERTV, Fana broadcasting and diretube.com domains. The author employed three machine learning classification techniques (Naive Bayes, Multinomial Naive Bayes and Support Vector Machines) using n-grams presence, n-grams frequency and n-grams-TF-IDF features selection methods. The experiments are conducted using 576 Amharic opinionated texts collected from ERTA, Fana Broadcasting and diretube.com manually. The Experiment indicates that uni-grams term frequency feature selection methods perform the best for all algorithms (Support Vector Machine, Naive Bayes and multinomial Naïve Bayes). Based on their relative performance of classification, Support Vector Machine registers with 78.8% accuracy outperform, Naive Bayes with 77.6% and multinomial Naive Bayes with 74.7%.as shown from the result obtained SVM performed better than NB and MNB algorithm.

Studied by [16] works on the title of opinion mining from Amharic entertainment text using machine learning approaches (Naïve Bayes, Decision Tree and Maximum Entropy). The experiment was conducted using 616 Amharic optioned texts. The study obtained 90.9 %, 83.1% and 89.6% using Naïve, Bayes, Decision Tree and Maximum entropy algorithms respectively. However, the study did not control negation, because the study uses uni-gram as a feature for classification. The result only shows positive and negative polarity but it did not include neutral.

Comparison of some related work

Summary of some of the related works in terms of their objective/goal, methods used, data source and result found in remark are summarized below in Table 2.3.

Table 2. 3: Overview of some previous work

Author	Objectives/goals	Methods/ techniques	Data Resource /domain	Sentiment classification approaches	Result
[57]	Develop feature level opinion mining and summarization model for the Amharic language.	rule-based approach	Amharic blog	Feature level	Two experiments are conducted and have achieved the accuracy of 85%.
[16]	Appling opinion mining to create a classification model for Amharic entertainment reviews.	Machine Learning Approach (Naïve Bayes, Decision Tree and Maximum Entropy)	Entertainment review	Document level	By combining the two methods they are able to improve the results over either of the methods alone. and has achieved the accuracy of 90.9%, 83.1 and 89.6 respectively
[51]	design a trilingual sentiment analysis system on social media using English, Amharic and Tigrigna languages.	Lexicon based approach	Social media Facebook, YouTube	Sentence level	Better result found in using basic lexica by using Term Counting method achieved the accuracy of 85%
[40]	to determine the polarity of the customer reviews of mobile phones at aspect level as positive, negative and neutral	Ditionary based approach	Product review (mobile phone review)	Aspect level	Aspect based Sentiment Orientation System' perform well and has achieved the accuracy of 67%.
[6]	Build sentiment mining model for Arabic texts	Supervised Machine learning approach, SVM	Movie Review	Both sentence level and document level approach for document level use hierarchical approach and for sentence level grammatical and semantic approach	Grammatical sentence level is better accuracy than the others because of considering a general structure for the Arabic Sentence.
[20]	Build News synthetic approach	synthetic approach	Restaurant review	Aspect level	Total

	for aspect-based opinion mining				The accuracy of 78. 04% was obtained on the manually annotated test dataset.
Author	Objectives/goals	Methods/ techniques	Data Resource /domain	Sentiment classification approaches	Result
[57]	Develop feature level opinion mining and summarization model for the Amharic language.	rule-based approach	Amharic blog	Feature level	Two experiments are conducted and has achieved the accuracy of 85%.
[16]	Appling opinion mining to create a classification model for the Amharic entertainment reviews.	Machine Learning Approach (Naïve Bayes, Decision Tree and Maximum Entropy)	Entertainment review	Document level	By combining the two methods they are able to improve the results over either of the methods alone. and has achieved the accuracy of 90.9%, 83.1 and 89.6 respectively
[51]	design a trilingual sentiment analysis system on social media using English, Amharic and Tigrigna languages.	Lexicon based approach	Social media Facebook, YouTube	Sentence level	Better result found in using basic lexica by using Term Counting method achieved the accuracy of 85%
[40]	to determine the polarity of the customer reviews of mobile phones at aspect level as positive, negative and neutral	Ditionary based approach	Product review (mobile phone review)	Aspect level	Aspect based Sentiment Orientation System' perform well and has achieved the accuracy of 67%.
[6]	Build sentiment mining model for Arabic texts	Supervised Machine learning approach, SVM	Movie Review	Both sentence level and document level approach for document level use hierarchical approach and for sentence level grammatical and	Grammatical sentence level is better accuracy than the others because of considering a general structure for the Arabic Sentence.

				semantic approach	
[20]	Build News synthetic approach for aspect-based opinion mining	synthetic approach	Restaurant review	Aspect level	Total The accuracy of 78. 04% was obtained on the manually annotated test dataset.

Chapter Three

Design and Experiment of Aspect/Feature Level Opinion Mining from Amharic Text

Identifying and extracting features, determining opinions regarding identified features, organizing and summarizing structured subjective text are the most common activities in aspect level opinion mining.

3.1 Flow of Architecture for sentiment analysis from opinionated Amharic texts using feature level

The general architecture of opinion mining model for opinionated Amharic texts shown in Figure 3.1. The architecture has Seven major components, these are Data collection, Data Preprocessing, Feature extraction, Morphological Analysis, Classification task, Aspect based polarity classification and Evaluation.

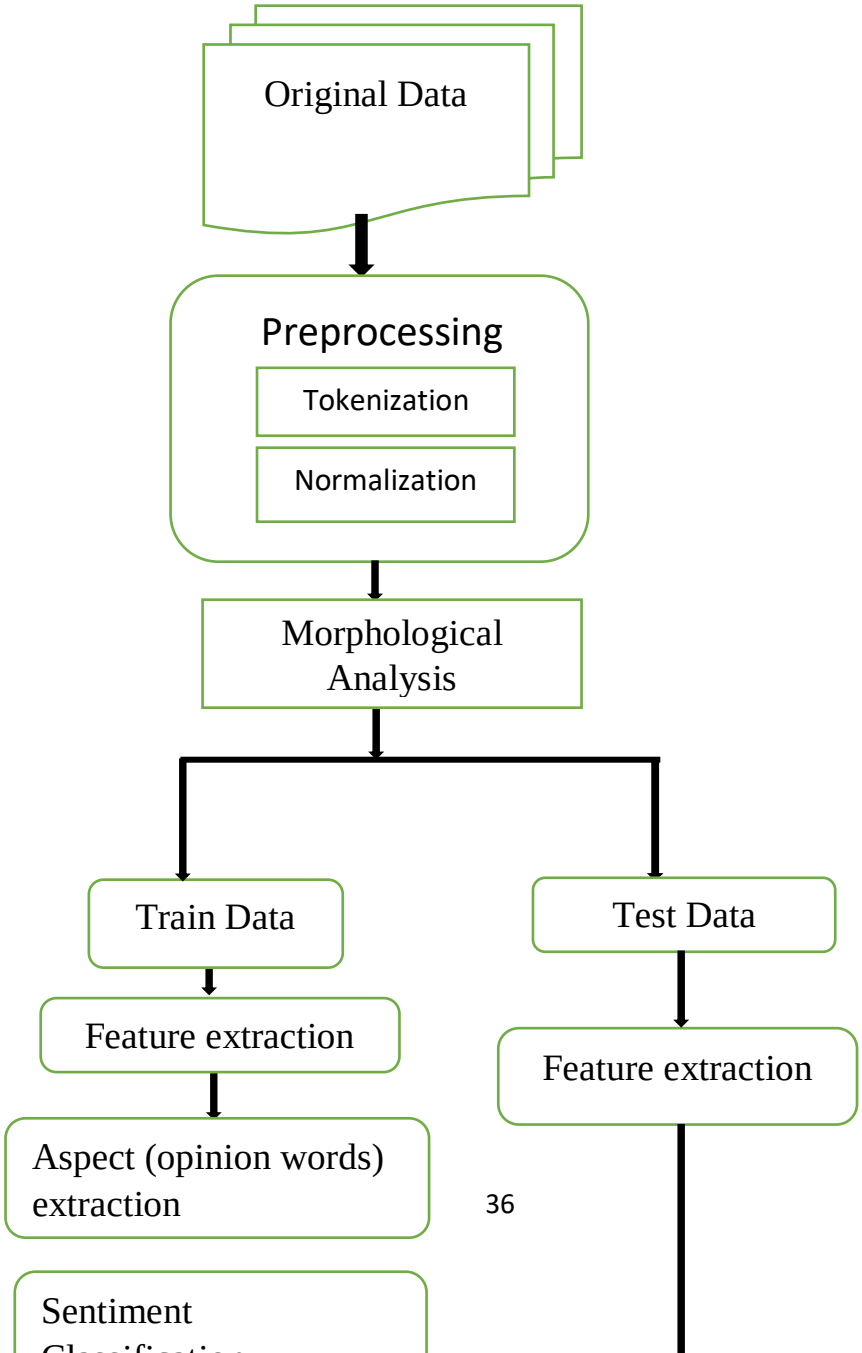

Original Data
Preprocessing
Tokenization
Normalization
Morphological Analysis
Train Data
Test Data
Feature extraction
Feature extraction
Aspect (opinion words) extraction
Sentiment

Figure 3.1: The Architecture of sentiment analysis model for opinionated Amharic News

The model takes opinionated text or News data as an input. The texts were preprocessed to make the system efficient and effective in their performance. Then the supervised learning algorithms (NB and SVM) are applied to build the model from the labeled training set. The next task is an evaluation of the model and performance status of the model.

3.1.1 Data collection and preparation

Data collection the first step of sentiment analysis consists of collecting data from Amharic News in Amhara Mass Media Facebook pages. These data cleaning and demanding systematic data preprocessing tasks. We use a supervised machine learning for opinion classification tasks that require an annotated corpus to train and test a classifier. So, the construction of a labeled corpus is a very important step because it would allow for more experiments, especially with supervised classification.

The main reason why we used the News data domain is due to the lack of readily available data written in the Amharic language electronically such as in web, blogs and online forums in others domain. As a result, it is relatively easier and more manageable to collect Amharic News data manually than any other domains. Hence of the Amharic News data, we used for

conducting the experiments are collected by an organization manually after collecting from the Facebook page coding into a computer and categorize into three labeled classes. These are positive, negative and neutral.

As a result, a total of 1200 News data are collected from the sources described above. After the data is collected, preprocessing tasks were applied to construct the final data set (data that are used as input for the modeling tool) from the initial raw data. Collected data using Facebook Graph API.

Amharic News data are more difficult for sentiment analysis. This is because News program reviews domains often contain many others domain such as political News, sport, music, drama, film and guest program and the data contain opinions and/or subjective sentences about the character of football players, guests, artist of the Amhara News program hosts. The review possibly will also contain many positive, negative and neutral. It is also difficult to collect neutral data.

3.1.2 Data preprocessing

The data pre-processing is done so that we can remove unwanted data and make the data more efficient. In order to perform opinion mining, the document should first be converted to an acceptable representation that can be used by the classifier. The pre-processing activity is important to improve the accuracy, efficiency, and scalability of the classification process. Preprocessing activity involves normalization and tokenization. The input for this process is a text document, but not every word in the text is meaningful for categorization or retrieval. For this reason, documents must be processed and represented to a concise and identifiable format or structure. Non-standard words such as numbers, abbreviations, and dates are removed from the dataset.

Following pre-processing steps are split and tokenization sentence which performed using an open source tool called NLTK. A text file was taken as an input in which documents were present and pre-processed output was taken out in a text file.

3.1.2.1 Tokenization

Tokenization refers to the process of splitting the text into a set of tokens (usually words). This process detects the boundaries of a written text. The Amharic language uses a number of punctuation marks which demarcate words in a stream of characters which include ‘arati net’ ibi’ (□□), ‘net’ ela ‘serezi’ (□), ‘diribi ‘serezi’ (□) exclamation mark ‘!’ and question mark’?’. These punctuation marks don’t have any relevance in opinion mining task and have to be removed. Collected document divided into sentences and a sentence further divided into words.

In this case, tokenization is the first step in the preprocessing component. In this step, input texts are tokenized into a stream of characters using white spaces and punctuation which helps to convert into a list of words.

3.1.2.2 Normalization

Normalization refers to the consistency of characters**,** after tokenization. It is normalization of homophones that are followed. Amharic writing system has homophone characters, characters with the same pronunciation but different symbols; for example, it is common that the character □ and □ are used interchangeably as □□ and □□ to mean work. The fidels are □ and □, □ and □, □ and □ and □, □, and □. In Amharic, the letters □, □, □ and □ converted into □, □ and □ converted into □, □ and □ converted into □ and □ and □ converted into □; Therefore, in this research work, the letters □, □, □ and □ are taken as common representations for the letters that have the same meaning and pronounciation in Amharic language. For example, □□□□□□, □□□□, □□, □□□□, □□□□, □□□, □□□□□□□□□□, □□□□□□□, □□□□□□□, □□□, □□□□□□, □□□□□□□, □□□□ converted into □□□□□□, □□□□, □□, □□□□, □□□□, □□□, □□□□□□□□□□, □□□□□□, □□□□□□□, □□□, □□□□□□, □□□□□□□, □□□□.

3.1.3 Morphological Analyzer

The morphological analyzer is essential and foremost one for any type of Natural Language processing work [52]. The morphological analyzer component is used to analyze English, Amharic and Tigrigna words into their constituent morphemes. Morphological analysis is

important especially for morphologically rich languages like Amharic and Tigrigna because it is practically difficult to store all possible words in a lexicon. In this study, the morphology Amharic words are analyzed using HornMorpho.

3.1.3.1 Amharic Morphological Analysis

Amharic is a morphologically rich language and highly inflected in number, gender, tense, person etc. Therefore, to handle those inflected words, a morphological analyzer tool is needed. So far, various researches have been carried out on Amharic morphological analysis although there is no effective Amharic morphological analyzer tool available. Even if there is Horn Morpho morphological analyzer tool, We have used Michael Gasser‟s HornMorpho 2.2 to analyze the reviews. HornMorpho is a Python program that analyzes Amharic, Orominya, and Tigrinya words into their constituent morphemes (meaningful parts) and generates words along with their class category based on their morphology.

3.1.4 Feature Extraction

Feature extraction from text is to be used for a machine learning model using python and scikit learn. Most machine learning algorithms can't take in the straight text so we will create a matrix of numerical values to represent our text using CountVectorizer.

The bag-of-words model is the simplest method, it constructs a word presence feature set from all words of an instance [36]. The ways data can be represented are feature-based or bag-of words representation. By feature we a mean that to capture the pattern of the data selected and the entire dataset must be represented in terms of them before it is fed to a machine learning algorithm. The NLTK classifier expects *dict* style feature sets to transform out text into a dic*t*.

In this paper/study, the bag-of-words model has been used for feature extraction. Because of the way to represent text data for machine learning algorithm and the bag-of-words model helps us to achieve that task. The bag-of-words model is simple to understand and implement. It is a way of extracting features from the text for use in machine learning algorithms.

The *bag-of-words ()* function is a simple list comprehension that constructs a *dict* from a given

words, when every word gets the value *True*. Since we have to assign a value to each word in order to create a *dict*, *True* is a logical choice for the value to indicate word presence. If we knew the universe of all possible words, we could assign the value *False* to all the words that are not in a given list of words. But most of the time, we do not know all the possible words. Plus, the *dict* that would result from assigning *False* to every possible word would be very large. So instead, to keep feature extraction simple and use less memory, we stick with assigning the value *True* to all words that occur at least once. We don't assign the value *False* to any word since we do not know what the set of possible words are; we only know about the words we are given. In the default *bag-of-words* model, all words are treated equally. The implementation of the bag of words model this function takes an input of a sentence and words (our vocabulary). It then extracts the words from the input sentence using the previously defined function. It creates a vector of zeros using numpy zeros functions with a length of the number of words in our vocabulary.

Lastly, for each word in our sentence, we loop through our vocabulary and if the word exists, we increase the count by 1. We return the numpy array of frequency counts.

```
from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer(analyzer = "word", tokenizer = None, preprocessor = None, stop_words = None, max_features = 3000)

def bagofwords(sentence, words):
    sentence_words = extract_words(sentence)
    # frequency word count
    bag = np.zeros(len(words))
    for sw in sentence_words:
        for i,word in enumerate(words):
            if word == sw:
                bag[i] += 1
            return np.array(bag)
```

This is sample features □□□□, □□□□, □□□□□, □□□□, □□□, □□□, □□□ , □□□, □□□ , □□□□, □□□

3.1.5 Aspect-sentiment extraction (opinion word extraction)

Adjectives are considered to be the opinion words through which feelings of the opinion holder have expressed nouns and noun phrases are considered to be featured on which the opinions are expressed. Adjectives or opinion words are extracted using the Morfessor tool. Adjectives are extracted and saved in a text file for further processing. We could identify and extract features from the text by using their category name labelled as a noun.

Examples:

1. □□□□□□ □□ □□ □□ □□ is an opinion while □□□ is a feature

2. □□□□ □□□ □□ □□□□□ □□ □□ □□□□□ is an opinion while □□ □□□ is a feature

3.1.6 Detection of sentiment words

This activity is responsible for detecting polarity terms and contextual valence shifter terms. After the review is preprocessed, every valid term in the review is checked whether it is sentiment word or not. This is done by a simple detection mechanism where the whole lexicon is scanned for every term. If the term exists in the dictionary, then the term is a polarity word (positive or negative) or a contextual valence shifter (negation or intensifier). Polarity words are terms that can express opinions towards an object such as '□□' (good) that expresses a positive opinion and '□□□' (bad) that expresses negative opinion towards an object. These terms are properly tagged in the lexicon with computer interpretable values as '+' for positive opinion terms and '-' for negative opinion terms. Then, if a term is found in the lexicon and if its corresponding value is '+', then this opinion term is positive. Similarly, if a term is found in the lexicon and if its corresponding value is '- ', then this opinion term is negative.

3.1.6.1 Incorporating contextual valence shifters

There are two different aspects of valence shifting that are used to improve the basic system (a system without considering contextual valence shifters). These are negations and intensifiers. Negations are terms that reverse the sentiment polarity of a certain term[59]. For

example, consider the following sentence '□□□ □□ □□' versus '□□□ □□ □□□□□'. In the first one '□□' (good) is a positive term so this sentence is positive. When '□□□□□' (not) is applied to the clause, '□□' (good) is being used in a negative context and so the sentence is negative. Intensifiers are terms that change the degree of the expressed sentiment. For example, in the sentence '□□□ □□□ □□ □□', the terms □□□ □□ (very good) are more positive than just '□□' (good) alone. On the other side, in the sentence '□□□ □□ □□□□', the term □□□□ (even though), makes this statement less positive.

These are examples of overstatements and understatements. Overstatements are terms that increase the intensity of a positive/negative term, while the understatements decrease the intensity of that term. Terms that overstate or understate are also listed in our lexicon. To identify overstatements and understatements, all positive sentiment terms in our model are given a value of +2. If they are preceded by an overstatement in the same clause, then they are given a value of +3. If they are followed by an understatement in the same clause, then they are given a value of +1. Negative terms are given a value of -2 by default. If they are preceded by an overstatement in the same clause, they are given a value of -3. If they are followed by an understatement in the same clause, they are given a value of -1.

3.1.6.2 Weight assignment and polarity propagation

In this phase the main activities are: weight assignment and polarity propagation. All possible sentiment terms are tagged in the lexica by '+' and given a default value of +2 at run time. All the negative sentiment terms are tagged by '-' and given a default value of -2. Before the final average polarity weight is calculated, the polarity propagation is done which is used to modify the initial value of the sentiment terms. This modification of the initial value or weight is done only if the sentiment word is linked to a modifier term (negations or intensifiers).

The polarity propagation is done according to the following rules.

Rule 1: if a sentiment term is not linked to any contextual valence shifting term, the initially assigned weight is considered to have maintained their default polarity weight values with +2.

Rule 2: If a sentence contains only negative sentiment words without any modifier words, then the negative sentiment word is assigned its default polarity weight values -2.

Rule Three If a positive sentiment word in a sentence is preceded by overstatement word, then the default polarity weight value of the positive sentiment word is increased by one (i.e., shifts from +2 to +3). □□□ □□□ □□□ □□ □□ the word □□□ modify the polarity weight values of the sentiment words in each sentiment sentences. This example shows an overstatement words and amplify the polarity strength of the sentiment sentences

Rule Four If a negative sentiment word in a sentence is preceded by an overstatement word, then the default polarity weight value of the negative sentiment word shifts from -2 to -1 and the sentiment expression will be strong negative.

Example: □□□ □□□□□ □□□□□ □□□□□ □□□□□ □□□ □□□ □□□□□□□ □□□ □□□ □□ In this example,□□□ is negative sentiment words. While extremely, □□□ amplifier sentiment words which amplify the semantic orientation of the negative sentiment words in the sentence.

Rule Five If a positive sentiment word in a sentence is preceded by understatement word, then the polarity weight value/strength of the positive sentiment word is decreased by one. The understatement word makes to attenuate the polarity strength of the sentiment words in the sentence. In this case, the semantic orientation of the sentiment will be weak positive. Example: in the sentence '□□□ □□ □□□□' (even though), the polarity weight of the sentiment term '□□' (good) is decreased from the initial value +2 to +1 due to the understatement term '□□□□' (even though).

Rule Six If the negative sentiment word in a sentence is preceded by understatement word, then the polarity weight value of the negative sentiment word is decreased by one. In this case, the sentiment expression of the sentence becomes weak negative and polarity weight value shifts from -2 to -3.

Example: in the sentence '□□□ □□□ □□□ □□' (it is very bad), due to the overstatement '□□□' (very), the initial weight of the sentiment word '□□□'(bad) is decreased from -2 to -3.

Rule Seven If a positive sentiment word is preceded or followed by negation terms in the sentence, then the sentiment of the positive word reverses into negative sentiments. Example, in the sentence '□□□ □□□ □□□□ (even though), the initial weight of the sentiment term is increased from -2 to -1 due to the understatement term.

Rule Eight If a negative sentiment word is preceded or followed by negation words in the sentence, then the polarity of the sentiment sentence is shifted into positive sentiments.

Example: □□□ □□ □□□□□ in this example, the sentences have positive sentiments since the negation words change the semantic orientation of the negative sentiment words in the sentence into positive sentiments.

Rule Nine: If a positive sentiment word preceded or followed by negation and intensifier words, then the semantic orientation (i.e., polarity and strength) value of the sentiment sentence is unchanged. For example, □□□ □□ □□ □□□□□ in this example, the sentences have positive sentiments with polarity weight value of +2.

Rule Ten: If a negative sentiment word in a sentence preceded or followed by negation and intensifier words, then the sentiment of the sentence has still negative sentiments and its polarity weight value is unchanged. For example, □□□ □□□□□□ □□ □□□□□ in this example, the sentences have negative sentiments with a polarity weight value of -2.

3.1.7 Sentiment Classification

Sentiment classification component is used to classify the sentiment sentence into positive, negative and neutral sentiment categories based on the average polarity weight value of the sentiment terms in the sentence. In this research work, sentiment classification performs based on rule-based techniques which take the clue from sentiment lexicons. It extracts sentiment terms from the sentence using the dictionary of sentiment terms and its polarity weight values in the sentiment lexicon. The polarity value of the sentence is calculating from the polarity value of the sentiment words.This is performed by adding individual polarity weight values of the sentiment words in the sentence.

$Dp\sum_{i=0}^{n} Pi$……(3.1)

Where, Dp is data polarity value, n is the total number of positive , negative or neutral words in the sentence Pi is the individual polarity value of the sentiment words (i.e., positive, negative or neutral).

According to the result of the equation, if the value of Dp is greater than zero then the review is categorized into a predefined category positive. Similarly, if the value of Dp is less than zero then the review is categorized in to a predefined category negative. Finally, if the total average weight of all the individual terms is equal to zero, the data is categorized in to the category neutral based on its aspects.

3.1.8 Aspect based polarity classification

The aspect-sentiment extraction module deals with the detection, extraction, and classification of explicit aspects and their associated sentiments. For example, the review sentence “The battery life is good” contains “battery life” as an aspect and “good” as a sentiment word. □□□□□□ □□□ □□ □□ □□□ □□ □□□□□ □□□ □□□□□ □ □□□□ □□□ □□ □□□□□□ □□□□□□ □□□ □□ □□ contains □□□ as an aspect and □□□ □□ as sentiment word polarity positive and □□□ as an aspect and □□ □□□□□ as sentiment word polarity negative. After completion of the following task, number of positive, negative and neutral words are calculated.

Chapter Four

Experiment

4.1 Introduction

Rule-based and supervised Algorithm (Naive Bayes and Support Vector Machine classifiers) used for conducting the experiment. Rule-based for labeling the data and Naive Bayes and Support Vector Machine classifiers used to train and test the data. We tested each technique individually and evaluate its performance. The procedure is, as a standard in supervised machine learning tasks, first training a classifier on pre-classified training data and then evaluating the performance of the classified one on the unlabeled set of test data. We selected to work Natural Language Processing ToolKit (NLTK). This package is equipped with several classifiers (i.e. NB, SVM). All programming has been done in the Python programming language and executed in the programming environment Window 10 python interactive shell. In this study generally, two experiments have been done.

Generally, in this research experiments are done by the two learning algorithms Naive Bayes and Support Vector Machine classifiers. All the results are presented in the subsequent section. All works are done using NLTK classification packages and python programming. Testing environment, manual data collection, evaluation metrics such as precision, recall, and F-measure, experimental results and discussions are all subtopics that will be discussed in the subsequent sections.

4.2 Development Environment and Tools

The testing has been done on a laptop computer with Windows 10 ultimate operating system, 2.17 GHz Intel Pentium Dual CPU, 4GB RAM, and 150 GB hard disk. Python 3.4.3 was configured for the testing of the proposed model. Every text file has been saved with a UTF-8 encoding system for Unicode characters processing, EndNote X7 for the citation of the referenced materials.

4.2.1 Tools

The tools that we have used Python, the rationale behind the choice of python programming language is that it is easy to learn but powerful programming language especially for text processing in NLP Toolkit (NLTK) used for preprocessing and tokenize data. It has sophisticated syntax and dynamic typing, together with its interpreted nature makes it an ideal language for scripting and rapid application development in many areas, particularly in natural language processing on most platforms. Notepad++ is a free source code editor and the replacement of Notepad that supports several languages. In this work, we used Notepad++ version of 7.5.1 with 32-bits which is used to build and edit the lexicon.
Morfessor-2.0.1 Morfessor is a family of methods for unsupervised morphological segmentation. SVM and Naive Bayes models used to train data. Scikit-learn machine learning library is used to implement a Naive Bayes and Support Vector Machine.

4.3 Data collection

Aspect level opinion mining techniques are evaluated on 1200 data manually collected from the Amhara Mass Media Agency domains News Facebook page is provided by the graph API. Facebook Graph is a graphic representation of the Facebook community, showing Facebook users and connections. This API presents a simple, consistent approach for developers to interact with the Facebook social graph. However, we don't want to use the Graph API Explorer to manually save all the comments, so we will use a Python script instead Appendix A. How Getting a Facebook Graph API Access Token and how the Facebook Graph API works and after this download comments as shown Figure 4.1.

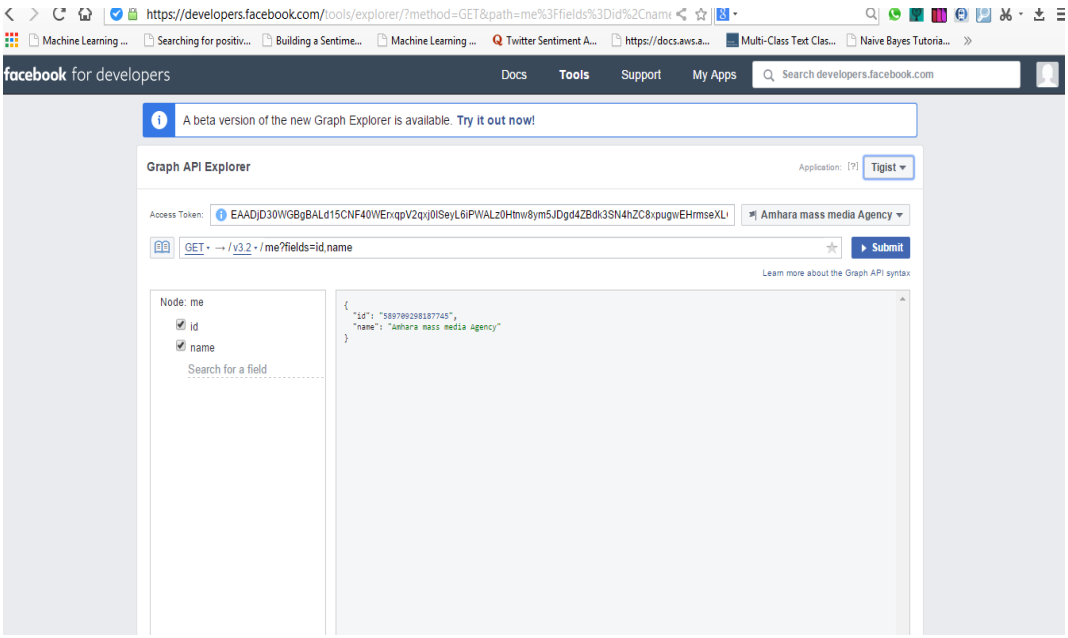


Figure 4.1: Sample data collection method using Facebook Graph API

Data preparation tasks are usually performed multiple times depending on the quality and size of the initial data set. A task includes cleaning, normalization, and tokenization of the data were performed to come up with the final suitable dataset for the selected algorithms.

4.4 Collecting aspect from the data

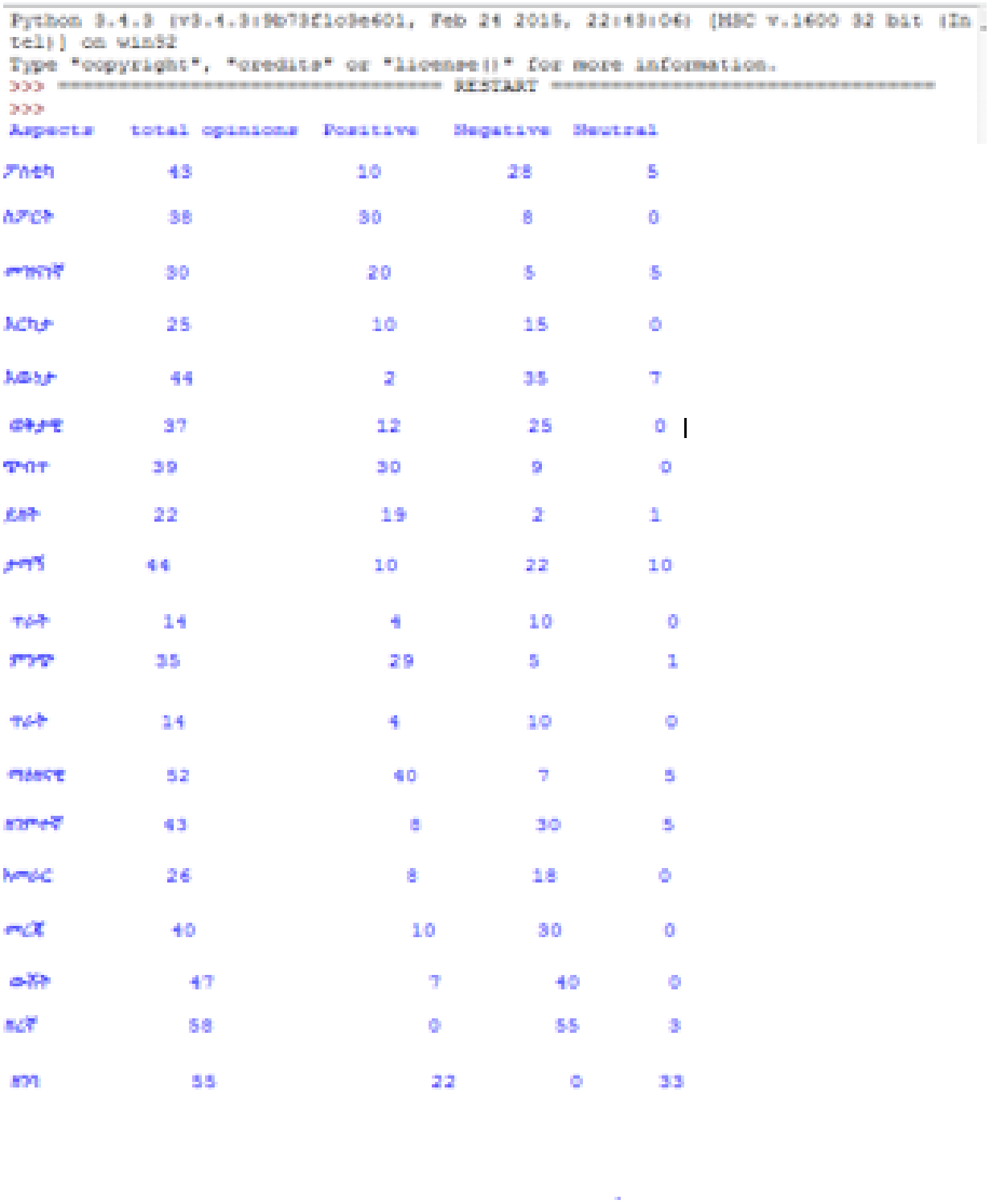
The term feature throughout this paper is the prominent attribute or aspect or component of collected data. It was sometimes challenging to identify a word as a feature. But this problem was solved by using an adjective, □□, before the word and if the phrase is linguistically

acceptable, the word that was tested by this adjective is a feature. For instance, □□ □□□□□, this phrase is linguistically acceptable. Therefore, the word □□□□□ could be taken as a feature.

4.4.1 Determining polarity of opinion words and Summarization of aspect-opinions

We have developed a lexicon of Amharic opinion words. These words are general opinion words for service domains. Employed lexicon, consists of 1200 (appendix G) negative and 1100 positive (appendix H) opinion words. Among 3300 opinion words, more than half are from [51]. work and the remaining are ours. We have used this lexicon to check whether the words around the features are opinion words or not and also to determine the polarity of words around the features.

When we say opinions along with features, we mean that the adjacent adjective, either to the left or to the right, of the identified aspects. During manual collection, we have considered only adjacent words that means not all adjectives are collected. These adjacent words are determined as opinion words along with their polarity. Count overall polarity of opinion words using lexicon words as shown in Figure 4.2.



```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ========== RESTART ==========
>>>
```

Aspects	total opinions	Positive	Negative	Neutral
ምክክ	43	10	28	5
ሰላምና	38	30	8	0
መጠጥ	30	20	5	5
አርሻ	25	10	15	0
አመራር	44	2	35	7
ወቅታዊ	37	12	25	0
ግብርና	39	30	9	0
ደህና	22	19	2	1
ታጣኝ	44	10	22	10
ጥሪት	14	4	10	0
ምንም	35	29	5	1
ጥሪት	14	4	10	0
ማጠቃለ	52	40	7	5
ጽሞናዊ	43	8	30	5
ኮሚሽ	26	8	18	0
መርጃ	40	10	30	0
ውጥኖ	47	7	40	0
ጸረ	58	0	55	3
ጽህ	55	22	0	33

Figure 4.2: Sample aspect - opinion Summary

4.6 Results and Discussion

4.6.1 Evaluation Procedures

The experiment is done to measure the overall performance of the aspect level sentiment analysis system. In this research work, a total of 1200 sentiment sentences were used to test the accuracy of the system. The test results achieved were presented in Section 4.8 in Table 4.4 and Table 4.5.

4.6.2 Evaluation Methods

The role of this activity is to describe the evaluation metrics of the designed system and followed by its test results. We have used precision, recall and F-measure evaluation metrics to measure the effectiveness of the selected approach. In this study, the experiment is done based on the algorithms and evaluates each evaluation metrics corresponding to each sentiment polarity classes. At last, the precision, recall and F-measure were calculated in the experiment for each algorithm.

4.6.2.1 Precision

It is a quantity within cross validation that represents a fraction of retrieved instances that are relevant. It is also called positive predictive value. High precision means that an algorithm returned substantially more relevant results than irrelevant. It is the number of true positives divided by the total number of elements labeled as belonging to that class. High precision means that the majority of items labeled for instance 'positive' indeed belong to the class 'positive'.

P (precision) = TP /(TP+FP) ……. (4.2)

4.6.2.2 Recall

Recall on the other hand is the fraction of relevant instances that are retrieved. It is also called sensitivity. It is the number of true positives divided by the total number of items that actually belong to that class. A high recall means that the majority of the 'positive' items were labeled as belonging to the class 'positive'.

R (recall)= TP /(TP+FN) ……. (4.3)

4.6.2.3 F-measure

A measure that combines Recall and Precision into a single measure of performance. This is just the product of Precision and Recall divided by their average.

Which is defined by the formula

F-measure =2(Precision×Recall)/(Precision+Recall)……(4.4)

4.6.2.4 Accuracy

To evaluate the performance of the different classifiers, the accuracy of each separate classifier is computed. Accuracy measures the proportion of a document that correctly obtained.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad \ldots\ldots (4.5)$$

4.6.3 Cross-validation

Cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it. It is the de facto approach used to assess the accuracy and validity of a statistical model. It is used for this study as well. The holdout method is the simplest kind of cross validation. The data set is separated into two sets, called the training set and the testing set. The errors it makes are accumulated as before to give the mean absolute test set error, which is used to evaluate the model [60, 61].

The advantage of this method is that it is usually preferable to the residual method and takes no longer to compute. However, its evaluation can have a high variance. The evaluation may depend heavily on which data points end up in the training set and which end up in the

test set, and thus the evaluation may be significantly different depending on how the division is made. In this study randomly split shuffled 100 objects into train set and test set 80–20 is randomly selected.

4.6.4 Confusion Matrix

A confusion matrix is a technique for summarizing the performance of a classification algorithm. Classification accuracy alone can be misleading if you have an unequal number of observations in each class or if you have more than two classes in your dataset. In general, the performance of sentiment classification is evaluated by using four indexes. They are Accuracy, Precision, Recall and F1-score [62]. The common way for computing these indexes is based on the confusion matrix as shown below:

Table 4.1: Confusion matrix

	Predicted positive	Predicted negative
Actual positive instance	True Positive (TP	False Positive (FP)
Actual negative instance	False Negative (FN)	True Negative (TN)

Where, TP: True Positive, TN: True Negative: False Positive, FN: False Negative

True positives are positive items that we correctly identified as positive for positive class and Negative items that we correctly identified as Negative for negative class.

True negatives are irrelevant items that we correctly identified as irrelevant. (negative comments not classified under positive for positive class and vice versa)

False positives (or **Type I errors**) are negative comments that we incorrectly identified as positive for positive class and positive comments that we incorrectly classified as negatives for negative class.

False negatives (or **Type II errors**) are relevant items that we incorrectly identified as irrelevant. (positive comments that incorrectly not classified under positive for positive class and negative comments that incorrectly not classified under negative for negative class.)

Table 4.2: Confusion Matrix for Amhara Mass Media Agency Facebook page news data set using Naive Baye's

	Predicted positive	Predicted negative
Actual positive instance	400	75
Actual negative instance	100	625

Table 4.3: Confusion Matrix for Amhara Mass Media Agency Facebook page news data set using SVM

	Predicted positive	Predicted negative
Actual positive instance	350	52
Actual negative instance	75	723

4.7 Result

This section presents the results of the empirical evaluation for determining the best approach to sentiment analysis in terms of classification algorithm and data preprocessing. We also provide a comparison of the developed sentiment classifier with a selection of publicly available sentiment classifiers. The experiments are performed using the datasets and the methods described below.

To evaluate the performance of the system and its effectiveness on sentiment polarity classifications, the experiment is conducted using the collected sentiment datasets. In this

experiment, we considered the effects of contextual valence shifter terms and taking an account of negation terms in sentiment sentences. The negation and contextual valance shifter terms include the amplifiers, diminishers and conjunction terms in sentiment sentences, due to this, the sentiment polarity classes are classified into three categories which are positive, negative, and neutral classes.

We evaluated two algorithms for sentiment analysis in order to select the most suitable one for our study. As a result, we found this approach inadequate for our study, and therefore we consider only the two other algorithms the supervised learning approach Naive Bayes and Support Vector Machine algorithm. Table 4.4 and Table 4.5shows the evaluation results of the experiment on each sentiment polarity classes.

4.7.1 Experiment using Naive Bayes

We conducted the first experiment by using NB algorithm experimentation, we use a simple bag-of-words approach with all unigram words in the corpus. In each stage, the results will be presented. We discuss the accuracy results and performance analysis by computing recall, precision, and F-measure.

Table 4.4: Evaluation Results using Naïve baye's

Polarity class	**Evolution matrices**		
	Precision	Recall	F-measure
Positive	**0.82**	**0.75**	**0.80**
Negative	**0.91**	**0.84**	**0.82**
Neutral	**0.79**	**0.82**	**0.83**

4.7.2 Experiment using SVM

To implement the Support Vectors Machines the library LIBSVM used [63, 64] Support Vector Machine is a kind of large-margin classifier which from previous studies has been reported to perform particularly well for sentiment analysis [38]. It aims at finding a decision boundary between classes that is maximally far from any point in the training data.

Table 4.5: Evaluation Results using SVM

Polarity class	**Evaluation Metrics**		
	Precision	Recall	F-measure
Positive	**0.85**	**0.72**	**0.80**
Negative	**0.94**	**0.85**	**0.87**
Neutral	**0.81**	**0.90**	**0.85**

4.8 Discussion

Table 4.4 and Table 4.5 shows the experimental results for the efficiency of the proposed model in different polarity classes for extracting features using the bag of word model along identified features. We developed a trained model to observe labeled opinions and classify the un observed ones. A corpus of 1200 data (opinions) is used to train and test the model with the labels positive, negative and neutral. Among these data 960 for training and 240 for testing.

The result under experiment one and two shows in Table 4.4 and Table 4.5, the precision of positive polarity classes is higher than the remaining polarity classes, because much of the test datasets contain the positive sentiment sentences and the sentiment sentences belong in the positive sentiment classes. In negative polarity classes, the precision is higher than their recalls. This indicates the classification approach predicts the right sentiment classification in sentiment polarity classes and the system achieves better correctness in the classification. In neutral polarity class, the system achieves the lower precision, but the highest recall from the other polarity classes. The reason is the effect of misspelling sentiment words in sentiment sentences, the equal occurrence of sentiment terms in the sentence, the size of the sentiment words in the Amharic sentiment lexicons in sentiment sentences.

In general, the absences of spelling checker, the existence of sarcastic sentences and ambiguous words in sentiment sentences have their own negative impacts/influences in

overall performance of the system in sentiment classifications. The values of the evaluation metrics are encouraging, large size of positive and negative sentiment terms in the lexicon, concerning the effects of valance shifters and negation terms in sentiment sentences are assumed to improve the performance of the sentiment polarity classifications

Generally, the overall accuracy for sentiment analysis with support vector linear classifier has achieved better result compared to naïve bayes approach. In short, most accurate sentiment analysis would be executed and implemented in the system for sentiment classification purpose. In practice, this system also provides greater accuracy when the more accurate result for testing sentimental words.

Chapter Five

Conclusion and Recommendation

5.1 Conclusion

Aspect level opinion mining is the process of extracting aspects or attributes of the target object, identifying opinions along with the extracted aspects, determine their orientation and summarize the reviews by grouping multiple opinions along features. This research work came up with a design of aspect/feature-level opinion mining from Amharic texts.

We have collected 1200 data from Amhara Mass Media Agency Facebook page. NB and SVM algorithms are used to classify text in to positive, negative and neutral. We evaluate our work using precision, recall, and F-measure performance metrics.

It can be concluded that the classification model is able to classify data within our domain, correctly with a reasonable level of accuracy. To perform the effectiveness experiment using Naïve baye's precision, recall and F-measure evaluation metrics were conducted and an average precision of 84%, average recall of 80% and average F-measure of 81% were obtained.

To perform the effectiveness experiment using SVM precision, recall and F-measure evaluation metrics were conducted and an average precision of 87%, average recall of 82% and average F-measure of 84% were obtained. This study showed that the promising results, but more comprehensive future works make this more findings improved. The SVM algorithm out performs than NB.

5.2 Recommendation

In this research, an attempt is made to design and develop an Aspect/Feature level opinion mining model. Arriving at a full-fledged Aspect/Feature level opinion mining for the specified language is time consuming and involves coordinated team effort from information science and linguistic professionals that work on different levels and subcomponents. The following are some of the recommendations for further research and improvement:

- ✓ There are two types of features: explicit and implicit. We have dedicated to the extraction of explicit features. Extracting implicit features is also very important that we will consider in the future work.
- ✓ It is better to prepare a standard corpus for Amharic sentiment analysis.
- ✓ Modeling Amharic ambiguous words will be a pioneering research area.
- ✓ Perform trend analysis on sentiments over a given time period.

References

[1] 1 Lucila Ohno-Machado Prakash M Nadkarni, 2 Wendy W Chapman2, "Natural language processing: an introduc tion," 2011.

[2] RM.Chandrasekaran G.Vinodhini "Sentiment Analysis and Opinion Mining: A Survey," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. Volume 2,, 2012.

[3] Feiyu XU & Xiwen CHENG, *Opinion Mining*, 2010.

[4] Deepak Singh Tomar and Pankaj Sharma, "A Text Polarity Analysis Using Sentiwordnet Based an Algorithm," *Deepak Singh Tomar et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, ,* vol. 7, 2016.

[5] Wondwossen Philemon and Wondwossen Mulugeta, "A Machine Learning Approach to Multi-Scale Sentiment Analysis of Amharic Online Posts

" *HiLCoE Journal of Computer Science and Technology,,* vol. Vol. 2, p. 8, 2015.

[6] Elie Challita Noura Farra, Rawad Abou Assi, Hazem Hajj, "Sentence-level and Document-level Sentiment Mining for Arabic Texts," presented at the 2010 IEEE International Conference on Data Mining Workshops, American University of Beirut, 2010.

[7] Harsh Thakkar and Dhiren Patel, "Approaches for Sentiment Analysis on Twitter:

A State-of-Art study," 2016.

[8] Sarah Schrauwen, "MACHINE LEARNING APPROACHES TO SENTIMENT ANALYSIS USING THE DUTCH NETLOG CORPUS," 2010.

[9] Bo Pang, "AUTOMATIC ANALYSIS OF DOCUMENT SENTIMENT," Philosophy, computer Science, Cornell University, Augest 2006.

[10] Manjunath Srinivasaiah Namrata Godbole, Steven Skiena, "LargeScale Sentiment Analysis for News and Blogs," 2010.

[11] B. Barla Cambazoglu Onur Kucuktunc , Ingmar Weber, Hakan Ferhatosmanoglu, "A Large-Scale Sentiment Analysis for Yahoo! Answers," USA, 2012.

[12] Bing Liu, *Sentiment Analysis and Opinion Mining*: Morgan & Claypool 2012.

[13] Shweta Nigam and Rekha Jain Richa Sharma, "MINING OF PRODUCT REVIEWS AT ASPECT LEVEL," *International Journal in Foundations of Computer Science & Technology (IJFCST),* vol. 4, 2014.

[14] Shibily Joseph Chinsha T C, "Aspect based Opinion Mining from Restaurant Reviews," *International Journal of Computer Applications (0975 – 8887),* 2014.

[15] Selama Gebremeskel, "SENTIMENT MINING MODEL FOR OPINIONATED AMHARIC TEXTS" DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE, COMPUTER SCIENCE, ADDIS ABABA UNIVERSITY, 2010.

[16] ABREHAM GETACHEW, "OPINION MINING FROM AMHARIC ENTERTAINMENT TEXTS," Degree of Master of Science in Information Science, INFORMATION SCIENCE, ADDIS ABABA UNIVERSITY, 2014.

[17] Ramaswami M Sharmista A, "SVM and Fuzzy SVM Based Opinion Mining In Tamil Using R," *American Journal of Engineering Research (AJER)* vol. Volume-7 2018.

[18] Alexandra Balahur, "Sentiment Analysis in Social Media Texts," presented at the European Commission Joint Research Centre, 2012.

[19] Fernando Ferri Alessia D'Andrea, Patrizia Grifoni and Tiziana Guzzo, "Approaches, Tools and Applications for Sentiment Analysis Implementation," *International Journal of Computer Applications (0975 – 8887),* vol. 125, 2015.

[20] Shibily Joseph Assistant P rofessor Chinsha T C, "A Syntactic Approach for Aspect Based Opinion Mining," presented at the Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC), Computer Science & Engineering Govt. Engineering College, Thrissur,Email: chinsha555@gmail. com,Email: shibilyj@gmail. com, 2015.

[21] Dr. L. Jayasimman2 S. Kasthuri1, and Dr. A. Nisha Jebaseeli3, "An Opinion Mining and Sentiment Analysis Techniques: A Survey," *International Research Journal of Engineering and Technology (IRJET),* vol. 03, 2016.

[22] Manasee Godsay, "The Process of Sentiment Analysis : A Study " *International Journal of Computer Applications (0975 –8887)* vol. 126 2015.

[23] Feiyu XU & Xiwen CHENG, *Opinion Mining*, 2010.

[24] Riddhiman Ghosh Lei Zhang, Mohamed Dekhil, Meichun Hsu, Bing Liu, "Combining Lexicon-based and Learning-based Methods for Twitter Sentiment Analysis," in *Combining Lexicon-based and Learning-based Methods for Twitter Sentiment Analysis*, R.

G. Lei Zhang, Mohamed Dekhil, Meichun Hsu, Bing Liu, Ed., ed. Hewlett-Packard Development Company, L.P: External Publication, 2011.

[25] B. Anitha N. Anitha, S. Pradeepa, "Sentiment Classification Approaches – A Review," *International Journal of Innovations in Engineering and Technology (IJIET),* vol. Vol. 3 3013.

[26] Jayashri Khairnar, "Analysis of different approaches to Sentence-Level Sentiment Classification," 2013.

[27] Suke Li, "Sentiment Classification using Subjective and Objective Views," *International Journal of Computer Applications (0975 – 8887)* vol. 80, 2013.

[28] Abhishek Anand Abinash Tripathy, Santanu Kumar Rath, "Document-level sentiment classification using hybrid machine learning approach," 2017.

[29] Karishma Pawar V. S. Jagtap, "Analysis of different approaches to Sentence-Level Sentiment Classification," *International Journal of Scientific Engineering and Technology*

vol. Volume 2 2013.

[30] Vimalkumar B. Vaghela Bhumika M. Jadav , PhD, " Sentiment Analysis using Support Vector Machine based on Feature Selection and Semantic Analysis," *International Journal of Computer Applications (0975 – 8887)* vol. Volume 146, July 2016 p. 5.

[31] 1 JoséMedina-Moreira María del Pilar Salas-Zárate, 2 Katty Lagos-Ortiz,Harry Luna-Aveiga,2 Miguel Ángel Rodríguez-García,3 and Rafael Valencia-García1, "Sentiment Analysis on Tweets about Diabetes: An Aspect-Level Approach," *3Computational Bioscience Research Center, King Abdullah University of Science and Technology, 4700 KAUST, P.O. Box 2882, Thuwal 23955-6900, Saudi Arabia,* vol. 2017, p. 9 pages, 2017.

[32] Niladri Chatterjee Nimit Bindal, " A Two-step Method for Sentiment Analysis of Tweets " *International Conference on Information Technology,* 2016

[33] Arun Kumar Solanki Rahul Rajput, "Review of Sentimental Analysis Methods using Lexicon Based Approach," *International Journal of Computer Science and Mobile Computing,* vol. Vol. 5, pp. pg.159 – 166, 2016.

[34] Shabib Aftab Munir Ahmad, Syed Shah Muhammad and Sarfraz Ahmad, "Machine Learning Techniques for Sentiment Analysis," *NTERNATIONAL JOURNAL OF MULTIDISCIPLINARY SCIENCES AND ENGINEERING*, vol. VOL. 8, , 2017.

[35] * Abd. Samad Hasan Basaria, Burairah Hussina, I. Gede Pramudya Anantaa, Junta Zeniarjab, "Opinion Mining of Movie Review using Hybrid Method of Support Vector Machine and Particle Swarm Optimization," presented at the Conference on Engineering & Technology 2012, MUCET 2012 Part 4 - Information And Communication Technology, Malaysian Technical Universities, 2013

[36] Michal Konkol Toma´s Brychcˇ ´ın, Josef Steinberger, "UWB: Machine Learning Approach to Aspect-Based Sentiment Analysis

" Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014) , pages 817–822, Dublin, Ireland, August 23-24 2014.

[37] * Walaa Medhat a, Ahmed Hassan b, and Hoda Korashy b, "*S e n t i m e n t a n a l y s i s a l g o r i t h m s a n d a p p l i c a t i o n s* :

A survey," *Ain Shams Engineering Journal*, vol. 3, April 2014.

[38] PhD Bhumika M. Jadav and Vimalkumar B. Vaghela, "Sentiment Analysis using Support Vector Machine based on Feature Selection and Semantic Analysis," *International Journal of Computer Applications (0975 – 8887)*, vol. 146, 2015.

[39] Mayura Kinikar** Jayashri Khairnar*, "Machine Learning Algorithms for Opinion Mining and Sentiment Classification," *International Journal of Scientific and Research Publications, ,* vol. Volume 3, 2013.

[40] Shweta Nigam2 and Rekha Jain Richa Sharma1, "OPINION MINING OF MOVIE REVIEWS AT DOCUMENT LEVEL," *International Journal on Information Theory (IJIT)*, vol. Vol.3, 2014.

[41] Prof. J. V. Shinde Bhagyashri Wagh, Prof. P. A. Kale, "A Twitter Sentiment Analysis Using NLTK and Machine Learning Techniques," *International Journal of Emerging Research in Management &Technology* vol. Volume-6, 2017.

[42] Gwanghoon Yoo and Jeesun Nam, "A Hybrid Approach to Sentiment Analysis Enhanced by Sentiment Lexicons and Polarity Shifting Devices," 2017.

[43] Alistair Kennedy and Diana Inkpen, "Sentiment Classification of Movie and Product

Reviews Using Contextual Valence Shifters," 2016.

[44] Rahman ullah1 Muhammad Zubair Asghar1, Shakeel Ahmad1, Fazal Masud Kundi1, Irfan ullah Nawaz, "Lexicon based Approach for Sentiment Classification of User Reviews," *Life Science Journal* vol. 11, 2014.

[45] Francisco Chiclana and Jenny Carter Orestes Appel, "A Hybrid Approach to Sentiment Analysis," presented at the 2016 IEEE Congress on Evolutionary Computation (CEC), De Montfort University, 2016.

[46] Hsiao-Ping Shih Chin-Sheng Yang, "A RULE-BASED APPROACH FOR EFFECTIVE SENTIMENT ANALYSIS," presented at the Pacific Asia Conference on Information Systems (PACIS) 2012 Proceedings, Yuan Ze University, 2012.

[47] Manasa Prakash M.S. Abirami M. Uma, "Sentiment analysis of informal text using a rule based model," *Journal of Chemical and Pharmaceutical Sciences,* vol. Volume 9 2016.

[48] Dani jel a M ERK L ER Z el jko AG I C 1 and "Rule-Based Sentiment Analysis in Narrow Domain: Detecting Sentiment in Daily Horoscopes Using Sentiscope," pp. pages 115–124, 2012.

[49] Mariana Romanyshyn, "RULE-BASED SENTIMENT ANALYSIS OF UKRAINIAN REVIEWS " *International Journal of Artificial Intelligence & Applications (IJAIA),* vol. 4,, 2013.

[50] Beth Skwarecki. (2019, 01). *https://en.wikipedia.org/wiki/Language*.

[51] Mebrahtu Tadesse G/Medhin, "Trilingual Sentiment Analysis on Social Media," Degree of Master of Science in Computer Science, Department of Computer Science, ADDIS ABABA UNIVERSITY, 2018.

[52] Martha Yifiru Tachbelie, "Morphology-Based Language Modeling for Amharic," Dissertation to Obtain the Degree of Doctor of Science, Department of Computer Science, Hamburg University, 2010.

[53] Solomon Atnafu Tessema Mindaye, "Design and Implementation of Amharic Search Engine," presented at the 2009 Fifth International Conference on Signal Image Technology and Internet Based Systems, Addis Ababa, Ethiopia, 2010.

[54] Meron Sahlemariam Tessema Mindaye, Teshome Kassie, "The Need for Amharic WordNet," 2010.

[55] (2019). *Amharic Studies*. Available: https://languages.ufl.edu/academics/llc-languages/amharic-studies/

[56] Meron Sahlemariam, "CONCEPT-BASED AUTOMATIC AMHARIC DOCUMENT CATEGORIZATION," DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE, COMPUTER SCIENCE, ADDIS ABABA UNIVERSITY 2009

[57] TULU TILAHUN HAILU, "OPINION MINING FROM AMHARIC BLOG," DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE, DEPARTMENT OF COMPUTER SCIENCE, ADDIS ABABA UNIVERSITY, 2013.

[58] Mengistu Kassa, "Sentiment analysis for classifying Amharic opinionated text " DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE, COMPUTER SCIENCE, ADDIS ABABA UNIVERSITY, 2013.

[59] Livia Polanyi and Annie Zaenen, "Contextual Valence Shifters," *In proceedings of the AAAI Symposium on Exploring Attitude and Affect in Text: Theories and Applications*

(published as AAAI technical report SS-04-07), 2004.

[60] (2019). *https://towardsdatascience.com/cross-validation-70289113a072*. Available: https://towardsdatascience.com/cross-validation-70289113a072

[61] "https://www.cs.cmu.edu/~schneide/tut5/node42.html," 2019.

[62] S Padmaja and Prof. S Sameen Fatima, "Opinion Mining and Sentiment Analysis - An Assessment of Peoples' Belief: A Survey " *International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC)* vol. 4, 2013.

[63] Chih-Chung Chang and Chih-Jen Lin. (2014). *LIBSVM*. Available: https://www.csie.ntu.edu.tw/~cjlin/libsvm/oldfiles/index-1.0.html

[64] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: A Library for Support Vector Machines," 2013.

[65] S. J. Vick, D. Bovet, and J. R. Anderson, "How do African grey parrots (*Psittacus erithacus*) perform on a delay of gratification task?," *Animal Cognition*, vol. 13, pp. 351-358, Mar 2010.

Appendix A: Facebook comment posts downloading code

```
import requests
import signal
import sys
import codecs
graph_api_version = 'v2.9'
access_token = 'YOUR_FACEBOOK_ACCESS_TOKEN_HERE'
# LHL's Facebook user id
user_id = '249655662614552'
post_id = '589709298187745'
# the graph API endpoint for comments on LHL's post
url = 'https://graph.facebook.com/ {}/[65]_[65]/comments'.format(graph_api_version, user_id, post_id)
comments = []
limit = 1200
def write_comments_to_file(filename):
    print()

    if len(comments) == 0:
        print('No comments to write.')
        return

    with open(filename, 'w', encoding='utf-8') as f:
        for comment in comments:
            f.write(comment + '\n')
    print('Wrote {} comments to {}'.format(len(comments), filename))
def signal_handler(signal, frame):
    print('KeyboardInterrupt')
    write_comments_to_file('comments.txt')
    sys.exit(0)
```

```
signal.signal(signal.SIGINT, signal_handler)

r = requests.get(url, params={'access_token': access_token})
while True:
    data = r.json()

    # catch errors returned by the Graph API
    if 'error' in data:
        raise Exception(data['error']['message'])

    # append the text of each comment into the comments list
    for comment in data['data']:
        # remove line breaks in each comment
        text = comment['message'].replace('\n', ' ')
        comments.append(text)

    print('Got {} comments, total: {}'.format(len(data['data']), len(comments)))

    # check if we have enough comments
    if 0 < limit <= len(comments):
        break

    # check if there are more comments
    if 'paging' in data and 'next' in data['paging']:
        r = requests.get(data['paging']['next'])
    else:
        break

# save the comments to a file
write_comments_to_file('comments.txt')
```

Appendix B: Training the model

```
import numpy
import os
import scipy
import string
import codecs
from sklearn import cross_validation
from sklearn import metrics
from sklearn import svm
from sklearn import naive_bayes
from sklearn.utils import check_arrays
import datasettings
from analyzer.parser import parse amh_corpus
from analyzer.parser import parse_training_corpus
from analyzer.vectorizer import Vectorizert
from sklearn.cross_validation import train_test_split
class Trainer(object):
Trains the classifier with training data and does the cross validation.
def __init__(self):
Initializes the datastructures required.
# The actual text extraction object (does text to vector mapping).
self.vectorizer = Vectorizer()
# A list of already hand classified tweets to train our classifier.
self.data = None
# A list containing the classification to each individual tweet
# in the tweets list.
self.classification = None
self.classifier = None
self.scores = None
def initialize_training_data(self):
```

```
Initializes all types of training data we have.
corpus_file = open(os.path.join(datasettings.DATA_DIRECTORY,
'full-corpus.csv'))
classification, tweets = parse_training_corpus(corpus_file)
reviews_positive = parse_imdb_corpus(
os.path.join(datasettings.DATA_DIRECTORY, 'positive'))
num_postive_reviews = len(reviews_positive)
class_positive = ['positive'] * num_postive_reviews
reviews_negative = parse_imdb_corpus(
os.path.join(datasettings.DATA_DIRECTORY, 'negative'))
num_negative_reviews = len(reviews_negative)
class_negative = ['negative'] * num_negative_reviews
self.data = tweets
self.classification = classification
#self.date_time = date_time
#self.retweet = retweets
#self.favorited = favorited
def initial_fit(self):
Initializes the vectorizer by doing a fit and then a transform.
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test  = train_test_split(
      features_nd,
      data_labels,
      train_size=0.80,
      random_state=1234)
classification_vector = numpy.array(map(
self.classification))
feature_vector = self.vectorizer.fit_transform(self.data)
return (classification_vector, feature_vector)
def build_word_dict(self):
```

```
Build sentiment dictionary and build vector of
weights for tweets.
fileIn = open(os.path.join(datasettings.DATA_DIRECTORY,
'AFINN-96.txt'))
wordDict = {}
line = fileIn.readline()
while line != '':
temp = string.split(line, '\t')
wordDict[temp[0]] = int(temp[1])
line = fileIn.readline()
fileIn.close()
fileIn = open(os.path.join(datasettings.DATA_DIRECTORY,
'AFINN-111.txt'))
line = fileIn.readline()
while line != '':
temp = string.split(line, '\t')
wordDict[temp[0]] = int(temp[1])
line = fileIn.readline()
fileIn.close()
word_dict_vector = []
for tweet in self.data:
word_list = tweet.split()
sum = 0
for word in word_list:
if word in wordDict.keys():
sum += wordDict[word]
word_dict_vector.append(sum)
return word_dict_vector
def transform(self, test_data):
Performs the transform using the already initialized vectorizer.
```

```
feature_vector = self.vectorizer.transform(test_data)
def score_func(self, true, predicted):
Score function for the validation.
return metrics.precision_recall_fscore_support(
true, predicted,
pos_label=[
SENTIMENT_MAP['positive'],
SENTIMENT_MAP['negative'],
SENTIMENT_MAP['neutral'],
],
average='macro')
self.classification_vector,
sparse_format='csr')
classifier=True)
for train, test in cv:
self.classifier1.fit(self.feature_vector[train],
self.classification_vector[train])
self.classifier2.fit(self.feature_vector[train],
self.classification_vector[train])
classification1 = self.classifier1.predict(
self.feature_vector[test])
classification2 = self.classifier2.predict(
self.feature_vector[test])
classification = []
for predictions in zip(classification1, classification2):
neutral_count = predictions.count(0)
positive_count = predictions.count(1)
negative_count = predictions.count(-1)
if (neutral_count == negative_count and
negative_count == positive_count):
```

```
classification.append(predictions[0])
elif (neutral_count > positive_count and
neutral_count > negative_count):
classification.append(0)
elif (positive_count > neutral_count and
positive_count > negative_count):
classification.append(1)
elif (negative_count > neutral_count and
negative_count > positive_count):
classification.append(-1)
classification = numpy.array(classification)
self.scores.append(self.score_func(y[test], classification))
def train_and_validate(self, cross_validate=False, mean=False,
serialize=False):
```

Appendix C: Bag of word model Code

```
import numpy as np
import re

def tokenize_sentences(sentences):
    words = []
    for sentence in sentences:
        w = extract_words(sentence)
        words.extend(w)

    words = sorted(list(set(words)))
    return words

def extract_words(sentence):
    ignore_words = ['a']
    words = re.sub("[^\w]", " ",  sentence).split() #nltk.word_tokenize(sentence)
    words_cleaned = [w.lower() for w in words if w not in ignore_words]
    return words_cleaned

def bagofwords(sentence, words):
    sentence_words = extract_words(sentence)
    # frequency word count
    bag = np.zeros(len(words))
    for sw in sentence_words:
        for i,word in enumerate(words):
            if word == sw:
                bag[i] += 1
    return np.array(bag)
vocabulary = tokenize_sentences(sentences)
```

```
from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer(analyzer = "word", tokenizer = None, preprocessor = None, stop_words =
None, max_features = 3000)

train_data_features = vectorizer.fit_transform(sentences)

vectorizer.transform(["sentence"]).toarray()
```

Appendix D: Sample Amharic Positive Sentiment Words

□□□□□ □□□□ □□□ □□□□□ □□□ □□□□□
□
□□□ □□□□ □□□ □□□□□ □□□□ □□□□□
□□□□ □□□□□ □□□□ □□□□ □□□□□ □□□□□
□□ □□□□□ □□□□ □□□□□ □□□□ □□□
□
□□□□ □□□□ □□□□ □□□□ □□□ □□□□
□□□□□ □□□ □□□ □□□□ □□□□ □□□□
□□□ □□□□□ □□□ □□□□□ □□□□□ □□□□
□
□□□□□ □□□□□ □□□ □□□□ □□□ □□□□□
□□□ □□□□□ □□□□ □□□□□ □□□ □□□□□
□□□□ □□□□□ □□□□□ □□□□ □□□□
□□□□□ □□□□□ □□□□ □□□□□ □□□ □□□□□
□

□□□□ □□□□□ □□□□□ □□□□□□ □□□□□□□ □□□□□□□ □□□□□
□□□□□□

Appendix E: Sample Amharic Negative Sentiment Words

□□□ □□□□ □□□□ □□□□ □□□□
□□□ □□□□ □□□□ □□□
□□□□ □□ □□□□ □□□□
□□ □□□ □□ □□ □□□□□
□□□ □□□ □□□ □□ □□ □□□□□ □□□□
□
□□□□□ □□□□□ □□□ □□□ □□□□□ □□□
□
□□□□□ □□□□□ □□□□□ □□□□ □□□ □□□□
□□□□ □□□□ □□□□ □□□□ □□ □□□□
□□□□□ □□□□ □□□□ □□□ □□□□
□□□□ □□□□□ □□□ □□□□ □□ □□□□
□□□□□ □□□□ □□□ □□ □□
□□□□ □□□□□ □□□□□ □□□□□ □□□□
□□□□□ □□□□ □□□□ □□□□ □□□□ □□□□
□□□ □□□□□ □□□□□ □□□□ □□□□
□
□□□□ □□□ □□ □□□□□ □□□□ □□□
□□□ □□□□ □□□□ □□□□ □□□□ □□□
□□
□□□ □□ □□□ □□□ □□ □□□□□ □□□□□
□□ □
□□□□ □□□□□ □□□□□ □□□ □□□□ □□□□
□
□□□□ □□□□ □□□ □□□□ □□□□ □□□

□□□□ □□□□□□ □□□□ □□□□ □□□□□□
□
□□□□□ □□□□□ □□□□□ □□□□ □□□□ □□□□
□□□□□□ □□□□□□ □□□□□□ □□□□□ □□□□□
□
□□□□□ □□□□□□ □□□□□□ □□□□□ □□□□□ □□□□□
□□□□ □□ □□□□□ □□□□□ □□□□□□ □□□□□□ □□
□
□□□□□ □□□□ □□□□ □□□□□ □□□□ □□□□□□
□□□□ □□□□□□ □□□□□ □□□□ □□□□□□ □□□□□
□□□□□ □□□□ □□ □□□□ □□ □□□□
□□□□□ □□□□ □□□□ □□□□ □□□□□□ □□□□
□ □
□□□□□ □□□□ □□□□ □□□□□□ □□□□□□ □□□□□□
□
□□□□ □□□□ □□□□□ □□□□ □□□□□□ □□□□
□□
□□□□ □□□□□ □□□□ □□□□□□ □□□□□ □□□□
□
□□□□□ □□□□ □□□□□ □□□□ □□□□□ □□□□□
□□ □□□□□ □□□□□ □□□□□□ □□□□□□ □□□□
□□□□□ □□□□□□ □□ □□□□□ □□□□□ □□□□
□□□□ □□□□□ □□□□□ □□□□□□ □□□□□
□□□□ □□□□□□ □□□□ □□ □□□□□□ □□□□
□□□□□□ □□□□□□ □□□□□ □□□□□ □□□□□□ □□□□□□
□□
□□□□ □□□□ □□□□ □□□□ □□□□□□ □□□□
□□□□□□ □□□□□ □□□□□□
□
□□□□□ □□□□□ □□□□□ □□□□ □□□□□□
□□□□ □□□□ □□□□ □□□□ □□□□□ □□□□□

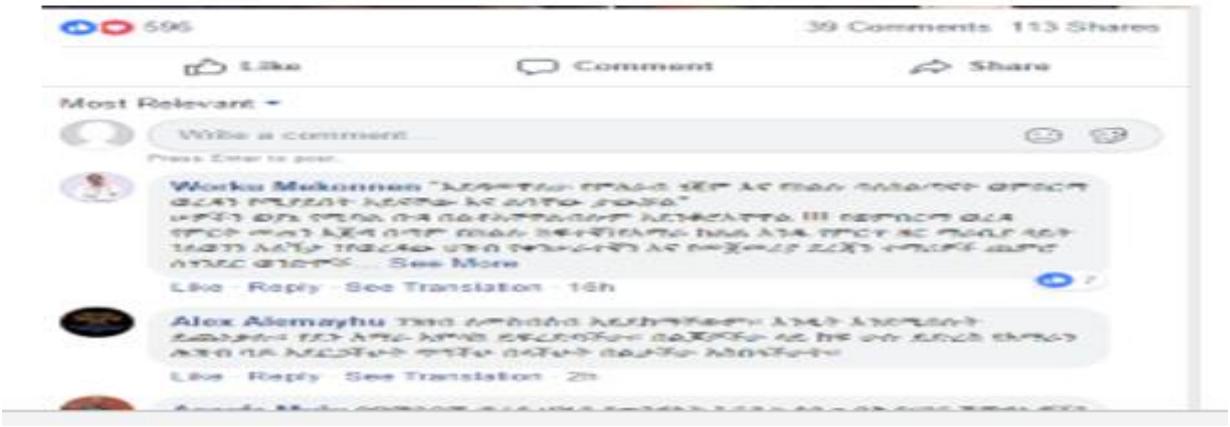
□□□□□□ □□□□□□ □□□□ □□□□□ □□□□□□ □□□□□□
□□□□□□ □□□□□ □□□□ □□□□ □□□□
□
□□□□□ □□□□□□ □□□□□ □□□□□□ □□□□
□□□□ □□□□□□ □□□□□□ □□□□□ □□□□□□
□□□□ □□□□□ □ □□□□□ □□□□□
□□□□□□ □□□□ □□ □□□□□□
□ □□□□□□ □□□□ □
□□□□□ □□□□□□ □□□□ □□□□
□□□□ □□□□□ □□ □□□□□□ □□□□
□□□□□□ □□□□□ □□□□□□ □□□□□
□ □□□□□□ □□ □□□□□□
□□□□□ □ □□□□□□
□□□□□
□□□□□□
□□□□□□
□□□□□□
□
□□□□
79

Appendix G: SERA Transcription System to Romanize Amharic Language using ASCI

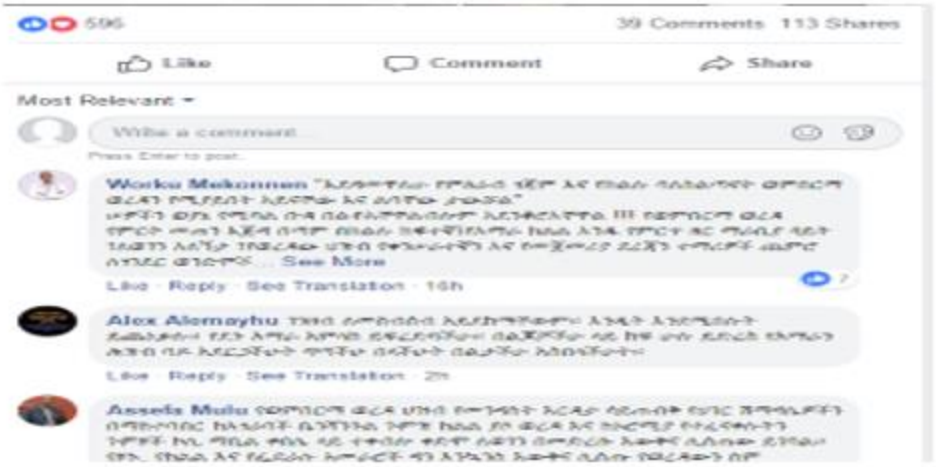
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
ha	hu	hi	ha	hE	h	ho
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
le	lu	li	la	lE	l	lo
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
Ha	Hu	Hi	Ha	HE	H	Ho
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
me	mu	mi	ma	mE	m	mo
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
^se	^su	^si	^sa	^sE	^s	^so
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ
re	ru	ri	ra	rE	r	ro
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
se	su	si	sa	sE	s	so
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ
xe	xu	xi	xa	xE	x	xo
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ
qe	qu	qi	qa	qE	q	Qo
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
be	bu	bi	ba	bE	b	bo
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ
te	tu	ti	ta	tE	t	to
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ
ce	cu	ci	ca	cE	c	co
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ
^ha	^hu	^hi	^ha	^hE	^h	^ho
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ
ne	nu	ni	na	nE	n	no
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ
Ne	Nu	Ni	Na	NE	N	No
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ
_a	_u	_i	_a	_E		_o
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ
ke	ku	ki	ka	kE	k	ko
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ
He	Hu	Hi	Ha	HE	H	Ho

we	wu	wi	wa	wE	w	wo
0	0	ዏ	ዏ	ዏ	ዕ	ዖ
`a	`u	`i	`a	`E	`	`o
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ
ze	zu	zi	za	zE	z	zo
ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ
Ze	Zu	Zi	Za	ZE	Z	Zo
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ
ye	yu	yi	ya	yE	y	yo
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ
de	du	di	da	dE	d	do
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ
je	ju	ji	ja	jE	j	jo
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ
ge	gu	gi	ga	gE	g	go
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
Te	Tu	Ti	Ta	TE	T	To
ጨ	ጪ	ጪ	ጫ	ጬ	ጭ	ጮ
Ce	Cu	Ci	Ca	CE	C	Co
ጰ	ጱ	ጲ	ጳ	ጴ	ጵ	ጶ
Pe	Pu	Pi	Pa	PE	P	Po
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ
Se	Su	Si	Sa	SE	S	So
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ
^Se	^Su	^Si	^Sa	^SE	^S	^So
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ
fe	fu	fi	fa	fE	F	fo
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ
pe	pu	pi	pa	pE	P	po
ሏ	ሗ	ሟ	ሧ	ሯ	ሷ	ሿ
lWa	HWa	mWa	sWa	rWa	sWa	bWa
ቷ	ቿ	ኋ	ኗ	ኟ	ኳ	ጏ
tWa	cWa	hWa	nWa	NWa	kWa	KWa
ዟ	ዧ	ዯ	ዷ	ዻ	ዿ	ፏ
zWa	ZWa	dWa	jWa	gWa	TWa	CWa
ፏ	ፗ	ፘ	፯	፰		
fWa	pWa	qWa	PWa	SWa		

Appendix H: Sample of data collection




Search
Chilote
Home
Find Friends
Chilote Dessalew
News Feed
Messenger
Videos on Watch
Shortcuts
Amhara mass
Explore
Pages
Events
Friend Lists
Games
Fundraisers
See More...
Create
Ad · Page · Group · Event
AMHARA MASS MEDIA AGENCY
አብመኤ
WWW.AMHARAWEB.COM


506
39 Comments 113 Shares
Like
Comment
Share
Most Relevant
Write a comment...
Press Enter to post.
Wosku Mekonnen
See More
Like · Reply · See Translation · 16h
Alex Alemayhu
Like · Reply · See Translation · 2h
Asseda Mulu