ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

HYBRID MODEL FOR AMHARIC SENTIMENT CLASSIFICATION

Girma Neshir Alemneh

A THESIS SUBMITTED TO IT DOCTORAL PROGRAM
ADDIS ABABA UNIVERSITY
PRESENTED IN FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY IN INFORMATION TECHNOLOGY
(INFORMATION RETRIEVAL)

ADDIS ABABA, ETHIOPIA
January 2022

Declaration Sheet

SCHOOL OF GRADUATE STUDIES
IT PhD Program
ADDIS ABABA UNIVERSITY

I hereby declare that this dissertation entitled **"Hybrid Model for Amharic Sentiment Classification"** was prepared by me, with the guidance of my advisers. The work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted, in whole or in part, for any other degree or professional qualification.

Student's Name	**Signature**	**Date**
Girma Neshir Alemneh		Jan. 31, 2022

(**Candidate Name**)

Approval Sheet

SCHOOL OF GRADUATE STUDIES

IT PhD Program

ADDIS ABABA UNIVERSITY

This is to certify that the dissertation prepared by Girma Neshir Alemneh, entitled **"Hybrid Model for Amharic Sentiment Classification"** and submitted in fulfillment of the requirements for the Degree of Doctor of Philosophy in Information Technology (Specialization in Information Retrieval), complies with the regularities of the university and meets the accepted standards with respect to the originality and quality.

Signed by Examining Committee:

Examiners' Name	**Signature**	**Date**
Prof. Michael Gasser		Jan. 30, 2022
(**External Examiner**)		
Mekuanint Beyene		Jan 27, 2022
(**Internal Examiner**)		
Andreas Rauber		Jan 28, 2022
(**Principal Adviser**)		
Dr. Solomon Atnafu		Jan. 27, 2022
(**Local Adviser**)		
Dr. Dereje Teferi		Jan 27, 2022
(**IR Track Coordinator**)		

Abstract

Amharic is a less-resourced language, which lacks a standard dictionary, stemmer, language detector, subjectivity detection, negation handling, Amharic sentiment lexicon and annotated Amharic corpora to carry out sentiment classification in social media texts. This research focuses on sentiment analysis of Amharic texts. The first part of the research is to generate most of these required resources and the second part of the research deals with enhancing performance of sentiment classification using proposed approaches.

In this research, four categories of corpora are prepared: general corpora (category I), annotated corpora (category II), lexical resources (category III) and pre-trained models (category IV). The annotated corpora, such as Facebook comments of GCAO (2,871), PMO (6,637), EBC (2,444) and Zemen YouTube Comments(1,440) are used for building and evaluating Amharic sentiment classification approaches. To remedy the problems of sentiment analysis of an under-resourced language (Amharic in this case), Amharic sentiment lexicons are generated using dictionary based and corpus based approaches. Using a dictionary based approach, SO-CAL (5,681) and SWN (13,677) are generated from English sentiment lexicons (using category III). Using corpus based approach, Amharic sentiment seeds are expanded to generate Amharic sentiment lexicon from Amharic corpora (using category I). At the threshold of 500, the generated lexicon has a size of 8,132. The generated lexicons are evaluated in terms of subjectivity detection, coverage, agreement and accuracy by comparison with the manual lexicon (baseline). The generated lexicons are used for subjectivity detection and negation handling. For sentiment classification (SC) of text on a topic, supervised, ensemble methods and BERT transfer learning are proposed, built, tuned and evaluated under small labeled observations (using category II). Finally, for enhancing the performance of Amharic sentiment classification, a hybrid model (i.e. voting, averaging and blender) is developed that combines the top performing classifiers of the earlier approaches. Experiments on the proposed hybrid models were done using category II annotated sentiment data sets. The results show that the proposed hybrid model (blender) has achieved performance gain as compared to SVM model (baseline) using the data sets. The complete sentiment classification system is showed in real-time and offline applications for detecting language, topics, subjectivity and prediction of sentiments of comments.

Keywords: *Amharic Sentiment Analysis . Subjectivity Detection . Amharic Sentiment Lexicon . Machine Learning . Negation Handling . Ensemble Learning . Transfer Learning . Hybrid Model*

Acknowledgements

First, I would like to thank my GOD, Jesus Christ, and his mother St. Virgin MARY. I would like to present my deepest appreciation to my supervisor Prof. Andreas Rauber, Information and Software Engineering Group (IFS), Institute of Information Systems Engineering (ISE), Vienna University of Technology, for his indulgent support for accomplishing this research. I wonder about his tremendous suggestions and comments that help me mature the research. He always pushes me to produce quality output from the research. I learned practical research experience which will keep growing my future career. I also would like to present my special thanks to my co-supervisor, Dr. Solomon Atnafu, Associate Professor of Computer Science, Addis Ababa University, for his closest guidance since the time of selection of my research topic. He is the best model showing me how to survey literature, present to colleagues, get feedback from colleagues, how I can get a topic for my research, playing a prominent role while offering me two advanced seminars in Information retrieval courses. He made it up-to-date to get practical research skills and experience for my research. I thank you again for your friendly and professional support. I also thank the professionals of the Ethiopian federal Government Communication Office Affairs (GCOA), Mr. Bekele, director of ICT, and his colleagues, specially Mr. Mehari Alemu and all social media section officers at Ethiopian Broadcasting Corporate (EBC) which provided me archived news and Facebook labeled comments at no cost. My special thank also goes to Ms. Meseret, Walta news archive officer, Mr. Gutema, Fana ICT director, and all archive officers in Ethiopian Reporter for their support in sharing the long years news archives. I present special thank Mr. Yacob Daniel, for his support to access SelamSoft and Amsalu Akilu Amharic-English dictionaries and for sharing diverse collections of Amharic lexical and corpora resources. I would also like to present my special thanks to Mr. Binyam Ephrem, lecturer and Ph.D. candidate, Addis Ababa University(AAU), for sharing resources, and providing me suggestions and comments on the generated sentiment lexicons. My thanks also go to Dr. Salehu Anteneh for writing up a number of support letters to various local news media and the embassy whenever I needed. I also thank Information Retrieval track coordinators Dr. Dereje Teferi for his comments of my research progress report. Last but not least, I would like to present my special thanks to my wife Meseret Bekele, my bigger daughter, Eden, and little baby, Arsema, for their understanding throughout my research, their embarrassment and love gives me the energy to work tirelessly, and for the success of my research.

Dedication

1. To my father **Neshir Alemneh**
   Abaye!: You were a good example by hard-work to the community. You passed too much challenges in your life. However, I lost you suddenly in 2015. Rest in Peace.

2. To my mother **Abesha Desie**
   Enaye!: You are always giving me a loving spirit that sustains within me. Long live!

3. To my wife **Mesert Bekele**
   Mesi!: I
   thank you so much for your understanding, be patient and keeping me forward. Long live!

4. To my daughters **Eden and Arsema**
   Buchu and Baby!: You are always becoming my energy and getting clean, refresh and get out of daily routines. *Edegulign!*

List of Abbreviations

BERT	Bidirectional Encoder Representations from Transformers
BOW	Bag Of Words
CM	Confusion Matrix
CNN	Convolutional Neural Network
CV	Cross-Validation
EBC	Ethiopian Broadcasting Corporate
ECV	Ethiopic-Constant-Vowel
ELMo	Embeddings from Language Models
GCOA	Ethiopian Federal Government Communication Office Affairs
GPT	Generative Pre-trained Transformer
HIS	High Intelligent System
IF	Information Fusion
LDA	Latent Dirichlet Allocation
LR	Logistic Regression
LSI	Latent Semantic Indexing
NMF	Non-negative Matrix Factorization
LSTM	Long Short-Term Memory
Masked LM	Masked Language Model
MCC	Matthews Correlation Coefficient
NB	Naive Bayesian
NER	Named Entity Recognizer
NH	Negation Handling
NLP	Natural Language Processing
NSP LM	Next Sentence Prediction Language Model
OM	Opinion Mining
PMI	Point-wise Mutual Information
PMO	Prime Minister Office
POST	Part-Of-Speech Tagger
PPMI	Positive Point-wise Mutual Information
RF	Random Forest
SC	Sentiment Classification
SERA	System for Ethiopic Representation of ASCII

SMOTE	Synthetic Minority Oversampling TEchnique
SO-CAL	Semantic Orientations CALculator
SVD	Singular Value Decomposition
SVM	Support Vector Machine
SWN	SentiWordNet
TF-IDF	Term Frequency-Inverse Document Frequency
DF	Document Frequency

List of Symbols

d	Document
D	Documents
w	Word
W	Words
$\leq$	Less or Equal to
μ	Centroid
Σ	Summation
AWV	Average Word Vector
TWA	TF-IDF Wieghted Average Word Vector
$softmax$	Softmax Function
$argmax$	Maximium among Set of Values
ϵ	An Element Of
h	Hypothesis
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
log	Logarithmic Function
$cosine$	Cosine Function
κ	Kappa Value Determing Intra-agreement
Po	Observed Agreements
Pe	Expected Agreement by Chance
$P-$	Probability of Being Negative
$P+$	Probability of Being Negative
ϵ	Is an Element Of
$\cdot$	Element-wise Multiplication
$+$	Addition
$-$	Subtraction Operator
$\times$	Multiplication Operator
$\sqrt{(}x)$	Square Root of x

Contents

List of Tables

List of Figures

1. Introduction

This research deals with the detection and classification of Amharic text sentiment in the situation where there are limited natural language resources such as sentiment lexicons, labeled data sets, part-of-speech tagger and named entity recognizer, just to name a few. Sentiment information is crucial for improving the quality of good governance and public services for citizens. It accomplishes this by delivering timely and summarized sentiment information in favor or against a particular topic or event activity at any instance.

1.1 Background

The amount of opinion-oriented data generation is increasing very fast on daily basis. This is facilitated by the emergence of the second generation web which came out as the platform for social media applications that enabled users to share their opinion in various formats such as texts, videos, photos, audios and graphics [14]. This research deals with analysis of opinion expressed in textual form.

According to R. Shanmugam [130] the textual content of the web accounts for 80% of the total formats. Textual contents are chosen because text format can easily allow most social media users to express their feelings, opinions and views towards an aspect. The process of identification of texts with either subjectivity or objectivity is called subjectivity detection. The subjectivity terms expressing opinions, feelings, emotions, affections and sentiments in text are usually understood as synonymous and are used interchangeably in NLP. However, they have differences in meanings and usage [97, 90]. Lack of consistency in terminology usage leads to confusion and difficulties of interpreting concepts or features involved in the text analysis.

According to Miriam-Webster dictionary [72], the subjective terms such as Affect, Feeling, Emotion, Attitude, Opinion and Sentiment are defined as follows:

"Affect": *"The conscious subjective aspect of an emotion considered part from bodily changes also a set of observable manifestations of a subjectively experienced emotion."*

Affect refers to non-conscious expression, which is the most complex and abstract expression to be fully represented using language. Thus, affect is more closely related to emotion [90]. Affect is not important for the purpose of this research as it is difficult to completely express using textual language.

"Feeling": *"An emotion state or reaction; often unreasoned opinion or belief."* Feeling refers

to the conscious expression of affects. For example, infants do not experience feelings because they lack language and biography. Therefore, infants' experience affect. Feeling is the term which is closely related and synonymous in meaning to sentiment. Besides, feeling is conscious activity which can be expressed using language.

"Emotion": *"Excitement; the affective aspect of consciousness; a state of feeling; a conscious mental reaction (as anger or fear) subjectively experienced as strong feeling usually directed toward a specific object and typically accompanied by physiological and behavioral changes in the body."* Emotion denotes the pre-conscious cultural or social expression of feelings or affects. From the previous example, the emotion of the infant is the direct expression of affect and/or feelings. For example, terms such as angry, sad, joyful, fearful, ashamed, proud, etc. are expressing emotions. Emotion is a non-conscious representation of physiological or bodily behavioral changes of expressions towards an object or issue. It is similar in meaning to sentiment. However, it is pre-conscious activity which is not very aligned to the purpose of this research.

"Attitude": Attitude is the tendency of acting with belief in a particular way towards an object or person [126]. An attitude is a view expressed consciously towards an issue or object. This term is closely aligned to the term that is needed in this research i.e. sentiment.

"Sentiment": refers *"to an attitude, thought, or judgment prompted by feeling; a specific view or notion."* Sentiment refers to the conscious emotional dispositions developed over time. For example, *"His criticism of the court's decision expresses a sentiment that is shared by many people."* This is the term that is needed for this research as the term sentiment is highly conscious activity of a person's judgmental view towards an issue or object, the consciously judgmental view of a person in favor of an issue (i.e. positive sentiment) or against an issue (i.e. negative sentiment).

Therefore, the term sentiment is used for this research as the intention is to measure or detect the judgmental views (expressed using written comment) towards the news posts on a national issue posted on Facebook page. This replicates the task performed by the professional recruited by the government to summarize the sentiment of Facebook users' comments for decision making.

"Opinion": refers to *"A view, judgment or appraisal formed in the mind about a particular matter; a belief stronger than impression and less strong than positive knowledge."* Opinion refers to the conscious personal interpretation of information related to emotion. Opinion is one's idea or knowledge regarding an entity. An opinion might change when new evidence is coming in. The term opinion is also very similar to sentiment as it is a personal and conscious judgmental view expressed towards an entity or object or issue. In this research, the term "opinion" can interchangeably be used with the term "sentiment". According to the authors in [97, 72], there is no unique definition for each of these subjectivity terms (affect, emotion, feelings, sentiment and opinion). This is due to the difference in usage and understanding of those terms across different disciplines such as psychology, neurology and sociology. For

instance, emotion has more than 92 different definitions in the literature and most of them varied in meanings [72].

Sentiment Lexicon: is a dictionary or a vocabulary that has been created for analyzing sentiment [122]. For example, SentiWordNet, AFINN, SO-CAL and MPQA subjectivity lexicon are some of sentiment lexicons in English language.

Sentiment Classification (SC): is the computational study of subjectivity detection and sentiment detection in data [94] towards an entity such as products, services, organizations, individuals, events, and their different aspects [40]. SC is also called Opinion Mining (OM). In other words, SC is the process of identification and extraction of polarity using different computational approaches. The task of sentiment analysis includes sentiment detection, transfer learning (cross-domain classification) and building resources [94]. In sentiment classification, the main task is to develop a system that is used to structure the opinionated text document into structured data by extracting different linguistic features from the text. Some of these text features include unigram or n-gram, part-of-speech, sentiment subtopic, semantic features, just to name a few [121].

Levels of Sentiment Analysis: This research computes sentiment at different linguistic levels. These include word-level, clause-level, sentence-level, document-level, and aspect-level sentiment analysis [94, 11].

Word-level: In word level sentiment analysis, one can detect sentiment, typically in adjectives, nouns and verbs. For example, the word "new" is an adjective and its sentiment is positive. The word "fail" is a verb and its sentiment is negative. And the word "pleasure" is a noun and its orientation is positive.

Phrase-level: This research can also analyze sentiment at clause (phrase) level, for example phrases such as "not sure at all" [negative sentiment], "not all sure" [negative sentiment] and "stand up and dance" [positive sentiment].

Sentence level: Sentence level sentiment analysis detects and classifies polarity expressed in each sentence. The sentence can be either simple or compound. The government officials (in GCOA) manually rephrase and summarize the sentiment of Facebook comments at sentence level. Accordingly, one of the labeled Facebook corpora (i.e. GCOA Facebook Comments, described in section 4.1.1) is sentence level labeled dataset. Sentence level sentiment classification of Facebook comments is also considered.

Document level: Document level sentiment analysis is the detection of the sentiment of the whole document that describes a topic/target. Sometimes, there is not much difference between sentence level and document level sentiment analysis as sentences can be treated as a small documents [94]. However, a document may contains multiple topics where the topics may be either different or opposite opinions towards a target. Facebook comments are varying in size from phrase level to document level. For example, two of the Facebook labeled corpora (i.e. EBC and PMO Facebook Comments, described in section 4.1.1, are document level, i.e.

having an average number of characters of 99 and 192, respectively). This research is carried out on document-level sentiment classification of Facebook comments.

Aspect-level: Aspect-based Sentiment Analysis is the identification of opinion of a specific feature or attribute (i.e. aspect) of a particular product or service [22]. For example, the sentence: "The picture quality of this phone is good, but the screen size is small" describes different aspects/features/attributes (image quality, screen size) of the same entity (phone). Users are giving their opinions about these features in an online product or service review. Extracting such opinion is very crucial to organizations so that they can take action about a feature or brand of their product or service.

Aspect based Sentiment Analysis is a challenging task that usually requires linguistic analysis such as Part-of-Speech (PoS), Named Entity Recognition (NER) and reference resolution. To extract aspects and their associated opinion words, PoS tags of words in a text are considered. The opinion words describing an aspect/feature are adjectives whereas aspects are nouns. For example, consider the sentence "The food was tasty and hot, but our waiter was not friendly" [22]. From this sentence, one can extract the polarity-feature pairs (tasty, food), (hot, food), (not-friendly, waiter). For aspect level SC, POS tagger is usually used to label and extract features and opinion words in a text.

For this research, the Facebook comments are feedback expressing opinions, views on national issues of different topics, including politics, news, culture, sport, art, and so on. To detect these topics before performing sentiment classification of Amharic, the approach used in [102] is used. As a result, this research does not consider aspect-level sentiment classification. This is because aspect-based sentiment classification is not common in Facebook comments of national aspects.

Nowadays, there are various approaches to sentiment classification. These include lexicon-based, machine learning-based and hybrid approaches. Lexicon-based approach relies on sentiment lexicon to obtain the sentiment score of a given text. Machine learning approach learns the text features to distinguish sentiment classes of text. Machine learning models can be either supervised, unsupervised or semi-supervised. The supervised approaches consists of conventional machine learning and deep learning algorithms. Ensemble learning is a supervised approach which combines multiple machine learning models (homogenous or heterogenous).

Transformers in NLP is a deep learning model which uses the attention mechanisms, representing and assigning different weights to an item (might be a word) depending on its significance in a sequence. The break-through in NLP is the development of transfer learning models. Currently, transfer learning in NLP refers to the imageNet version of pre-trained language models like ELMO, Open-GPT, and BERT. These models allow researchers to carry out downstream NLP tasks with minimal task-specific fine-tuning using less labeled data, less computational resources and faster development time. Hybrid model is the combination of multiple classifiers of the two or more of the previous sentiment classification approaches using different fusing

strategies like majority voting, averaging probabilities and stacking or blending models.

1.2 Challenges in Sentiment Analysis

The most common challenges in sentiment analysis are highlighted in the literature. These challenges include context, negation, sarcasm, irony and idioms, intensifiers, implicit opinions and sentiments, compound sentence, ambiguity and domain sensitivity [121]. Each of them are outlined as follows:

Context Sensitivity: An entry in sentiment lexicon could have different sentiment in the same domain. This is a serious problem of detecting semantic orientation of words in natural language text. For example, the word "large" has different polarities in the electronics domain:

1. The phone is too large to fit into a pocket. (Negative).
2. The memory card has large storage capacity. (Positive).

The opinion words that include short, large, long, small, etc. are context sensitive. That means, each of these words may opposite polarities in the same domain. Similar phenomena can happened in Amharic language.

Sentiment Shifters: Sentiment shifters, also called valence shifters, are words and phrases that are changing sentiment orientations in a text. There are different types of valence shifters including negation (such as "not", "never", "none", "nobody", "nothing", "nowhere", "neither", etc.). For example, "The voice quality of this phone is not good." In this sentence, the word "not" reverses the polarity of the opinion word "good". Deciding the scope of the negation in a sentence is not a trivial task. Unlike English, the negation of Amharic language may not be a single word unit. Negations are usually affixes to verbs and nouns. As Amharic is morphologically rich, a single Amharic verb holds a lot of linguistic information in the form of affixes. Some negative prefixes of Amharic verbs include **አል**-+verb, **አት**-+verb, **አይ**-+verb, **ባይ**-+verb and **ባት**-+verb. Some negative suffixes to Amharic abstract nouns such as noun+**-ቢስ**, noun+**-የለሽ**, noun+**-የለው**, etc. are negation markers. This issue is captured using negations handling for sentiment classification of Amharic [10]. Examples of negation words include **አይሆንም**(no), **የለም**(none) and **ምንም**(nothing), **አይችልም**(he cannot), **አትሂድ** (do not go/masculine/), and **አትሂጅ** (do not go/feminine/) are verbs with negation prefixes. Other sentiment shifters are intensifiers which are changing the degree of sentiment. Intensifiers are either amplifiers (that increase sentiment strength) or down-toners (that decrease the strength of sentiment) [90]. For example, "this phone is very good". The word "very" increases the strength of polarity of "good". On the other hand, the word "barely" in "The phone is barely any good" diminishes the strength of good to be less positive.

Domain Sensitivity: The task of sentiment analysis is affected by the domains from which sentiment lexicons are built or training data are collected. That means, an entry in a sentiment lexicon can have positive sentiment in one domain, while negative sentiment in another

domain. For example, the word “unexpected” has positive polarity in movie domain (e.g.“unexpected plot”) whereas negative polarity in automobile domain (e.g.“unexpected steering”) [121]. Similar challenges exist in Amharic sentiment analysis.

Implicit sentiment word: Sentiment lexicons cannot handle implicit sentiment words. These implicit words are obtained by deducing from the context of a sentence. For example,“the phone is too expensive to buy”. In this sentence, the target word is implicit (i.e. “price”). This word is rather implied from the sentence. The same cases happen in Amharic sentiment analysis.

Sarcasm, irony and idioms: Negation can be expressed using sarcasm and irony that is not simple to detect, even manually. For example,“What a great phone! Its camera stopped working in two days”. Amharic language also uses sarcasm, irony and idiom expressions in a text.

Incompleteness/lack of coverage: Sentiment lexicons do not cover all language constructs of sentiment words. For example, slang in social media will not exist in conventional sentiment lexicons. This affects the performance of sentiment analysis.

Lack of common-sense: Common-sense and multi-word sentiment words cannot be handled in the traditional sentiment lexicons. For example, sentiWordNet is used for lexicon-based sentiment classification whereas senticNet[1] is used for conceptual based sentiment classification [98]. The intuition behind senticNet is to consider common-sense knowledge to some extent as the concepts are free from restrictions imposed by part-of-speech affiliations of the language [121].

Ambiguity: Ambiguity is a situation where the same linguistic units are understood in different ways. At different levels of the linguistic unit, ambiguity is a major challenge in natural language processing. Lexical ambiguity is a result of different meanings of a word.

Example:

(1)"**ዳዊት ጉንፋን ስለያዘው ያስላል።።**" /David coughed due to the effect of flu./

(2)"**ዳዊት ስእል ያስላል።።**" /Someone is drawing a picture for Dawit./

(3)"**ዳዊት ቢላዋውን ያስላል።።**" /David is having his knife sharpened by someone./

In each of the above three sentences (1), (2), and (3), the same word "**ያስላል**" has different meanings (coughing, drawing, sharpening) with the same part-of-speech (i.e. verb). Resolving this ambiguity at different linguistic levels is one of the most challenging tasks in natural language processing. Unlike humans, computers lack common sense that is necessary for disambiguation [121].

This research addresses most of these above-mentioned challenges by cross-lingual, mono-lingual resource adaptations and transfer learning. First, Amharic sentiment lexicons are generated relying on both the context of terms in Amharic corpora and mapping an English sentiment lexicon into Amharic. The lexicon’s features are used to detect negation and subjectivity of Amharic texts. Text features for conventional machine learning algorithms and transformer (i.e.

[1]https://sentic.net/

dynamic language models) are investigated. The different approaches to Amharic sentiment classification are combined in the hybrid model and tested on Facebook comments.

1.3 Basic Facts about Amharic Language

Amharic is the official language of Ethiopia and the second most spoken language in Semitic family, next to Arabic language. Amharic is lingua franca of the country to unite the diverse populations and facilitate communication [47]. Amharic is historically descended from Ge'ez. Moreover, Amharic is a language with its own alphabet (or symbols) where each character represents a combination of consonant-vowel pairs [2] (See Appendix B). The alphabet was incorporated into Unicode in 2000 [46].
Amharic is morphologically complex [47]. Many words can be derived from the same root (stem) by affixations and a word can be formed by derivation which makes it to be challenging to carry out linguistic computation. Nowadays, more than 71% of Internet users are non-English speakers [86, 154, 85]. Amharic is one of these non-English languages. To-date, Amharic is showing more and more presence on the web. However, Amharic is a less-resourced language which lacks linguistic resources to perform sentiment classification. As a result, there are only few sentiment analysis research projects on Amharic texts. This is one of the reasons that initiated this research to carry out sentiment analysis for Amharic texts.

1.4 Research Problem

Problem Identification: Nowadays, sentiment analysis is a very hot topic which is widely carried out in the areas of natural language processing. Nevertheless, there is no universal solution to the above-mentioned universal challenges of sentiment analysis. There are also key universal problems to NLP such as (i) lack of NLP resources for most of the languages other than English, (ii) natural language understanding, and (iii) reasoning about large and multiple documents [3].
For the cases where the language under consideration is less-resourced, these problems are more serious when performing computational linguistic research. Less-resourced language problems are primarily associated with the lack of adequate and efficient tools for the processing of natural language: part-of-speech tagger, stemmer, lemmatizer, dependency parser, named-entity tagger, etc. The key approaches to resolving these challenges are to address the lack of annotated corpora to perform specific linguistic tasks [105, 76] including development of automatically labeled/annotated data, reliance on expert/linguistic knowledge, development of single language or "universal" solutions and/or resource creation [105].

[2] https://www.omniglot.com/charts/amharic.xls
[3] http://ruder.io/4-biggest-open-problems-in-nlp/

Another concern according to the reports of the authors in [63, 105, 76] is that half of the world's languages will vanish by the end of this century, unless proper attention is given by the concerned language professionals and the speakers of the languages. This report draws the attention of computational linguistic researchers to minimize the projected risk to the world's languages.

For the issues related to insufficiency of linguistic resources, there are attempts to develop possible techniques. These include creating more labeled NLP corpora, sentiment lexicon, leveraging resource-rich languages, using semi- or unsupervised methods, and rule-based methods [105]. For example, [49] manually created Amharic polarity lexicon with 1060 terms. Though the lexicon has correct sentiment annotation of entries, this lexicon has a number of drawbacks such as: limited coverage, no part-of-speech tags and no numerical sentiment strength information. For the challenges related to scarcity of labeled NLP corpora, some approaches use linguistic knowledge to seed unsupervised models. In addition, models/approaches of familiar languages are also leveraged to less-resourced languages [105, 76].

In [71], existing approaches to NLP are reviewed and adapted as necessary to less-limited languages like Amharic. Currently, research work on Amharic sentiment analysis is not adequate. Amharic sentiment analysis work can be categorized into lexicon based in [49, 139], machine learning based approaches [152, 36, 95, 141] and hybrid approach in [131].

This research investigates Amharic sentiment classification provided that there is scarcity of sentiment lexicons and insufficient labeled data sets. To properly address these problems, first the above-mention problems are formulated into research questions and research objectives. Six research questions are formulated as follows:

- RQ1: Given a sentiment lexicon in a resource rich language, to what extent can this research accurately build Amharic sentiment lexicons by applying cross-lingual resources with minimal manual annotation?
- RQ2: Given a monolingual corpus in Amharic, to what extent can this research accurately build Amharic sentiment lexicons from the corpus by expanding Amharic seeds with minimal manual annotation?
- RQ3: Given labeled documents and unlabeled documents, which machine learning algorithms perform best for classifying the unlabeled samples with its default hyper-parameter setting and different text feature configurations? Which features perform best for Amharic sentiment classification with the selected classifier?
- RQ4: Given the category of documents, to what extent does this research improve Amharic sentiment classification system relying on hyper-parameter tuning of the selected classifier in RQ3?

- RQ5: Under small size labeled corpus, does fine-tuning of BERT transfer learning improve Amharic sentiment classification system?
- RQ6: Given the fine-tuned textual features and the tuned models from answering RQ4, and RQ5, to what extent can this research accurately build a hybrid or ensemble learner that combines different approaches for detecting, classifying and predicting sentiment of Amharic text?

The response to the first research question enables the researcher to develop an efficient algorithm to automatically generate Amharic sentiment lexicons. This reduces the time and cost needed to develop the sentiment lexicon manually. The algorithm should be designed to map sentiment labels from sentiment lexicon of source language to target language. This approach is referred as dictionary based lexicon generation. Moreover, corpus-based sentiment lexicon generation algorithms would be developed to generate Amharic sentiment lexicon.
The research question (RQ3) is stated to investigate the salient text feature and the best machine learning for carrying out Amharic sentiment classification. Research question (RQ4) is formulated to test the improvement of sentiment classification by applying hyprer-parameter tuning of machine learning algorithm. Research question (RQ5) is formulated to test fine-tuning of BERT transformer for Amharic sentiment classification performance on small labeled samples.
Finally, research question (RQ6) is stated to test sentiment classification of hybrid model which combines two or more of top classifiers obtained in the earlier stages of the research questions.

1.5 Research Objectives

General Objective:

The general objective of the research is to build a hybrid model for carrying out Amharic sentiment classification.

Specific Objectives

To address the general objective, the following specific objectives will be accomplished.

- Collect and organize data sets and lexical resources.
- Develop algorithm for Amharic sentiment lexicon generation from resource-rich language (s) and mono-lingual corpora with minimum human effort.
- Evaluate the generated Amharic sentiment lexicons.

- Build Amharic text preprocessing pipeline (language identification, normalization, spelling correction, stop word removal, stemming, subjectivity detection and topic detection.
- Extract the possible text features from TF-IDF character ngrams, word level ngrams, word embeddings, TF-IDF weighted Word embeddings, LDA, NMF and compare their performance on Amharic SC.
- Build classifiers (e.g. supervised, stacked ensemble, fine-tune BERT) for small size labeled data.
- Develop a hybrid model which combines the top best classifiers (supervised, transfer learning, or other ensemble models) to effectively perform sentiment analysis of Amharic texts.
- Evaluate the performance of different approaches using evaluation metrics (accuracy, precision, F-measure and recall), and using statistical tests.

1.6 Applications of the Research Results

Sentiment detection of Amharic can have many applications in support of national issues and the users of the language. As Amharic is the working language of the Federal government of Ethiopia, analysis of summarized sentiment information related to governance is needed. This can help to take smart decisions or remedial actions or renew national policies in line with the needs of citizens. To date, the government of Ethiopia gets feedback from the citizen by traditional means. Manual summary report of the social media viewers' comments is generated. Sentiment information of customers is a measure of KPI (Key Performance Indicator) of an organization or business, or government institution. To increase the satisfaction level of customers/citizens, automatic detection of feedback is needed.

Nowadays, many user-generated data in social media (Facebook, YouTube, Instagram, etc.) is available that needs proper consideration. Availability of variety of data in different formats such as texts, image, videos, audios, etc. can be used to reduce costs, bring revenue, improve services, provide timely information for intelligent decision making, just to name a few. Research in the domain of sentiment analysis is critical for languages like Amharic. Nowadays, some social media like Facebook is restricting real-time data access through automated tools due to concerns on users' privacy. Nevertheless, in practice it is observed that sentimental expressions are causing conflicts, violence, etc. In such instances, developing countries like Ethiopia, with no system based solutions, has no other option but blocking the Internet. Making informed decision on such instances is a feasible solution.

This research should answer the research questions stated above. This leads to a need to design improved and effective solutions for sentiment classification of Amharic comments. Sentiment

analysis provides an understanding of the problem at hand to propose and execute alternative approaches in line with the stakeholders' needs.

1.7 Scope of the Study

Amharic is a language with scarce linguistic resources. One type of scarcity is lack of annotated Amharic sentiment corpora. This is one of the major challenges for researchers for carrying out sentiment classification tasks using supervised machine learning techniques. In this research, small Amharic sentiment corpora are annotated and an approach is taken including conventional supervised learning and transfer learning for Amharic sentiment classification.

This work deals with Amharic sentiment lexicon generation, and development of Amharic sentiment analysis. However, spam detection and hate speech detection are not the focus of this research.

The coverage of the generated Amharic sentiment lexicon is limited to the size of Amharic-English dictionary and the size of the Amharic corpus collection that is used while generating Amharic sentiment lexicon. Similarly, the accuracy and the reliability of sentiment ratings assigned to Amharic polarity words are affected by the quality and the size of the corpora, whether it covers sentiment terms in the text for expressing opinion. The Amharic-English dictionary used is generic and independent of any domain. Thus, the generated lexicon is also domain independent. In this research, Amharic Facebook comments (Facebook news reviews) are collected, prepared, and organized into categories. The domain of the Amharic web texts is usually general and related to news and events happened in the country.

The content of the opinionated Amharic corpora is related to events and news in the activities of Ethiopian government. The Facebook news posts and the corresponding Facebook users' comments are collected and organized to test and evaluate the performance of the proposed hybrid model.

To show how the research result can be applied, this research addresses the expectation of a stakeholder. The potential stakeholder is the Federal Government Communication Office Affairs under the Prime Minister Office Press Secretariat. This is office, where part of the sentiment labeled corpus is collected. A framework is developed to detect sentiment of Facebook comments (classified either positive or negative) under topic categories (sport, social, politics, culture, art, and so on).

Handling sarcasm and irony idiomatic expressions is beyond this study. Facebook comments containing hate speech, fake Facebook comments or spam content are also not considered in this research.

1.8 Research Contributions

The contribution of this research is highlighted as follows.

- The proposed approach for generating Amharic sentiment lexicon can highly reduce the cost and time of labeling terms manually.
- The approach developed for generating sentiment lexicon is language independent that can be adaptable to other less resourced languages.
- BERT transfer learning is developed for Amharic sentiment classification under situation where small labeled corpus is used. This could reduce the cost and time needed to annotate linguistic corpora.
- All [4] the generated resources (Amharic sentiment lexicons, word embeddings, corpora, source codes) are shared in a data repository[5]
- The proposed methods can be used by research communities in the areas of sentiment analysis and opinion mining. Besides, its applications include: detecting opinion towards market, collecting business intelligence, summarizing public opinion or feedback, assessing, monitoring government policies, measuring stances of the public before voting, just to name a few.

The key contributions of this work are associated with the artifacts generated in this research. Developed algorithms and approaches for Amharic sentiment lexicon generation are also part of the contribution. Generated resources include Amharic sentiment lexicons, pre-trained models, and annotated Facebook corpora.

1.9 Methodology

Choice of Research methodology: A suitable research methodology is very crucial to effectively address the research objectives. Figure 1.1 shows that design science has been chosen as a research methodology to be followed in this research [58]. The goal of design science is to create effective artifacts and utility to the expectations of stakeholders [146]. The reason for selecting design science is that it is suitable for designing IT artifacts to address specific problems in sentiment analysis domain in context of Amharic language texts. These artifacts include constructs, models, methods and instantiation as real life prototypical products. The other reason for following design science research methodology is that as it is used to address unsolved problems in new ways, and it is also used to find more effective solutions to already solved ones. [146].

[4] bilingual dictionary have copyright agreement restrictions.
[5] https://bit.ly/3HfJcxW.

Relying on the design science methodology, the contextual view of the research workflow to Amharic sentiment classification is presented in chapter 3.

1.10 Outline of the Thesis

The rest of this dissertation is organized as follows and a mapping to design science research process is presented in the diagram 1.1.

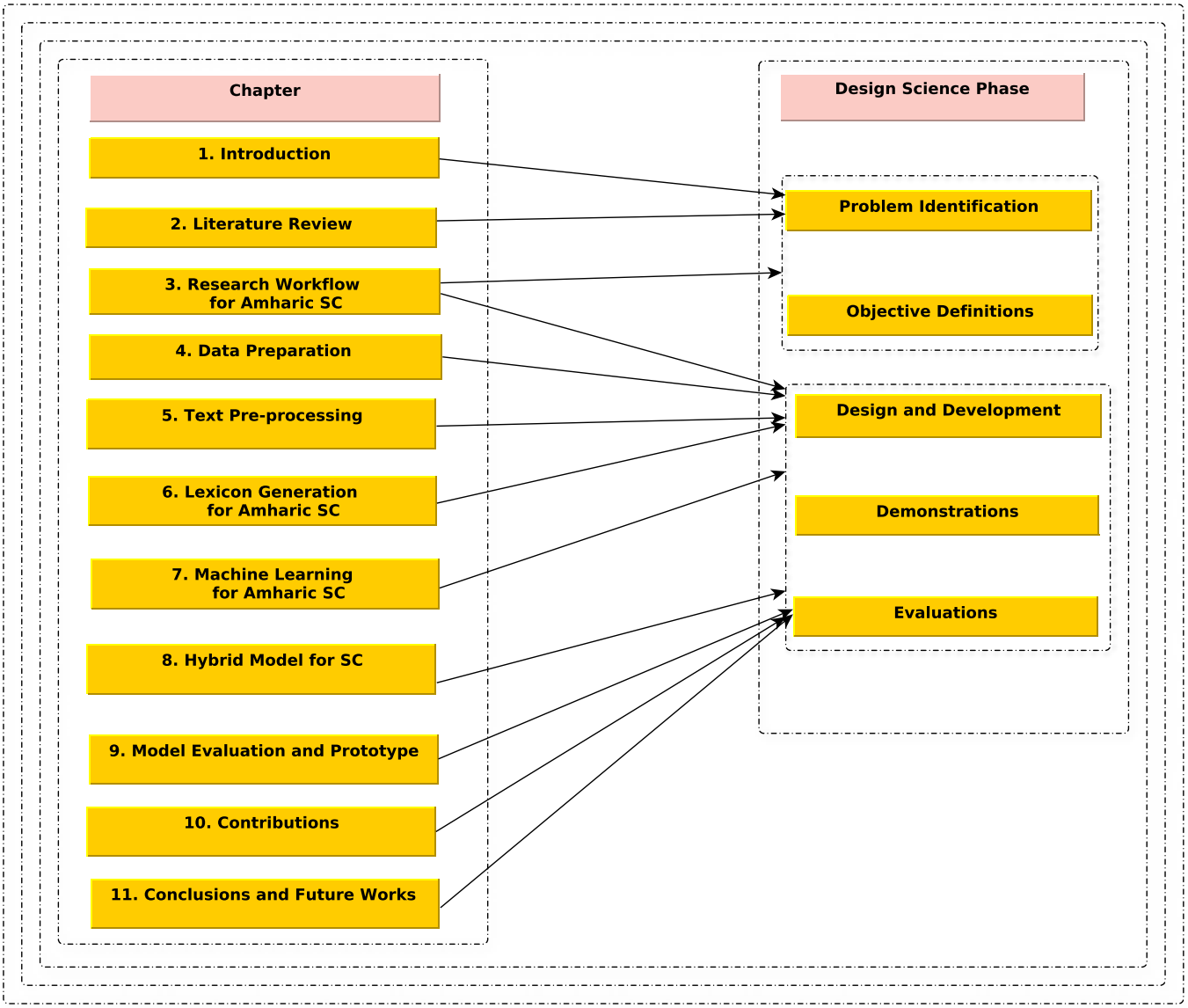


Figure 1.1.: Outline of this Thesis

Chapter two presents a review of related research works on sentiment classification. The chapter divides the review into three parts: (i) lexicon-based literature review, (ii) machine learning-based sentiment classification which is divided into subsections representing review of machine

learning-based, ensemble-based, BERT fine-tuning-based approaches and (iii) a hybrid model for Amharic sentiment analysis. The summaries of the related work that identifies and justify the research gaps in connection to the research questions and research objectives. Chapter three presents the research workflow for Amharic sentiment classification. The workflow reveals the high level view to carry out Amharic sentiment classification. This includes data preparation, preprocessing steps, machine learning based, and hybrid approaches.

Chapter four starts with the collection and preparation of the Amharic corpus. This is preprocessed as per the requirement of the experimental setting described in the subsequent chapters. Amharic text preprocessing operations are presented in chapter five. Chapter six presents the lexicon-based Amharic sentiment classification. This chapter starts by generating Amharic sentiment lexicons relying on dictionary-based and corpus-based. The generated sentiment lexicons are used for lexicon-based Amharic sentiment analysis. In chapter seven, a machine learning approach for Amharic sentiment classification is proposed. Models are tuned and the best model is also selected and tested. The model is the output of this chapter as an artifact that is used in the next chapter. It also presents an ensemble model of Amharic sentiment classification and BERT model to fine-tune for Amharic sentiment classification. Chapter eight presents the hybrid model for Amharic SC and the detail of the hybrid model to address the identified problems and the aim of this research. The ninth chapter presents an evaluation of the Amharic sentiment classification approach proposed in the earlier chapters. Sentiment classification performances of the proposed approaches in each module are evaluated for accuracy and reliability. The models with best performance results are used by the system for demonstration and are investigated against human judgments. The summary report for the research is presented in line with the research questions set out at the beginning of the research by proving the corresponding hypothesis.

Chapter ten reports the key contributions of the dissertation. The last chapter presents conclusions, and the future researches is also included.

2. Literature Review

This chapter organizes the literature review into five sections. The first section presents subjectivity detection related research. The second section presented the related work on lexicon-based methods. The third section presents related work on sentiment research which used machine learning algorithms. This section has subsections which present the review of machine learning approaches, such as conventional machine learning algorithm, ensemble methods, and transfer learning. The fourth section reviews the literature on hybrid approaches. The last section highlights the key points of the review related to this research.

2.1 Subjectivity Detection

K. Palshikar et. al (2016) in [106] developed meta-learners for subjectivity classification (i.e. either subjective or non-subjective). The meta-learners using stacking (MultilayerPerceptron, bagging, Logistic, AdaBoost) are tested and achieved an F-score of 71.9% using 1130 subjective sentences data sets. Feature sets used for this research include the number of adjectives, number of nouns, number of strongly subjective words, number of comparatives, number of superlatives, and number of adverbs. The author suggested the potential applications of subjective detection, including performance appraisals and scientific abstracts.
C. Sindhu et. al (2021) in [132] proposed machine learning approach for subjectivity detection using Twitter data sets. SVM outperformed with F-score of 84% using features from filtering tools such as TextBlob, Opinion Finder tool and SentiOutlook with an optimal threshold level of 0.5. Chaturvedi et. al (2017) in [31] presented the comparative investigation of the two approaches, specifically the handcrafted method and machine learning based methods for subjectivity detection. The authors also discussed the open challenges associated to annotating sentences, either subjective or objective. R. Satapathy et. al (2017) in [125] proposed reinforced deep CNN for detecting subjective sentences in nuclear energy tweets and the reported result has shown that 60% nuclear energy tweets were subjective.
As there are no labeled subjectivity data sets for Amharic to apply machine learning to this research proposes rule based subjectivity detection of Amharic texts. Subjectivity detection is used to weed out or filter objective (or factual) texts from subjective texts prior to carrying out Amharic sentiment classification.
In [129], authors developed lexicon based sentiment classification of Facebook comments in

Malay. English SWN Lexicon is used to get the score of each word of a comment and then the average score is used to decide the sentiment of a comment. The approach has shown better accuracy (63.41) of sentiment classification over term counting and term score summation method. However, the author suggested improving the accuracy of the adjective list, which does not exactly refer to the adjectives of Malay language as it propagated the sentiment information from English adjectives.

In [88], A. Mahoto et. al (2018) proposed lexicon based subjectivity detection of Twitter data. The result of the proposed method is significant compared to human evaluated data sets. The proposed method detects subjectivity of tweets by using SentiWordNet and labeled corpora. The score of the terms in a tweet is summed. If the score is greater than 0, then it is positive. If the score is less than 0, it is negative and if the score is zero; it is neutral. The proposed subjectivity detector (i.e either positive, negative or neutral) is evaluated using three data sets and also compared with expert opinions. Similarly, Aung et. al (2017) in [129] developed lexicon based approach to compute sentiment of students' feedback to evaluate the level of teaching performance. First, they manually built a lexicon for finding the sentiment score of each word in a student's comment. Then, based on the sentiment score, sentiment of the comment is assigned.

The proposed approach in this research is similar to the approach used in [88]. However, Amharic sentiment lexicon is needed for the purpose of detecting subjectivity of Amharic text.

2.2 Lexicon-based Sentiment Classification

Sentiment lexicons are generated using three approaches: manual approach, dictionary based and corpus based methods. The subsequent three subsections of this section present related work corresponding to the above methods of lexicon generation.

2.2.1. Manual based Lexicon Generation

This section briefly presents related work of manual lexicons for sentiment classification. For English language, there are various lexicons which were created manually for analyzing sentiment in a text. These include: AFINN (size of 2477) [103], MPQA (Multi-Perspective Question and Answering) subjectivity lexicon (size of 8,222) [151], SO-CAL (size of 10126) in [137], just to name a few. In Amharic language, below are research projects using manually built lexicon for Amharic sentiment classification.

Selama et. al [49] manually built Amharic sentiment lexicon of 1060 terms. They developed lexicon-based approach to classify sentiment at document level on Amharic movie reviews (data prepared by collecting from movie watchers and web resources) and on small news reviews relying on the manually built lexicon. Similarly, Tulu (2014) in [139] built a lexicon-based

feature-level sentiment analysis of Amharic blog texts in four domain reviews. The author used sentence patterns in which the rules specify the arrangement of feature terms, opinion terms, and negations in a sentence. The rules are manually extracted by identifying the features from the three domains—Hotel, University, and Hospital. Tulu [139] used this sentiment lexicon to identify the opinion words in the reviews. The performance of both works is reportedly encouraging.

In [138], the author developed lexicon-based sentence-level Amharic sentiment classification, achieving precision and recall of 92% and 82% for positive class, 94% and 96% for negative class and 90% and 90% for neutral class, respectively, on 800 annotated comments from social media. In [51], the author proposed trilingual (i.e. English, Amharic and Tigrigna) lexicon-based sentence level sentiment classification of 564 sentiment sentences collected from Facebook and YouTube, their approach obtaining average precision, recall and F-measure of 87.49% 84.78% and 85.99%, respectively.

Though, the manually built Amharic sentiment lexicon by [49] is more accurate, it has shortcomings. These include, (i) the polarity value of each term in this Amharic sentiment lexicon [49] was labeled either -2 or +2 without showing intermediate sentiment strength, (ii) this lexicon has no part of speech information, (iii) the effort required to further extend this sentiment lexicon manually is prohibitive, calling for automated methods, and the lexicon size is not sufficient to cover the sentiment words of Amharic.

To address the above problems, this research proposes automatic approaches to build Amharic sentiment lexicons. The subsequent sections review those approaches to lexicon generation.

2.2.2. Dictionary-based Lexicon Generation

Below are discussions of related work of dictionary based lexicon generations for sentiment classification of non-Amharic languages. Mahyoub et. al (2014) in [89] built an Arabic sentiment lexicon by exploiting Arabic WordNet. The authors started from seed list and then found negative and positive new polarity terms from synset relations of Arabic WordNet by using semi-supervised learning (i.e. propagation through the network synset relation). Finally, an Arabic sentiment lexicon of size over 800 positive, 600 negative and 6000 neutral polarity terms is generated. The generated Arabic sentiment lexicon was evaluated by incorporating it with a machine learning classifier and its classification accuracy achievement was 96%. The Arabic sentiment lexicon was very similar to SentiWordNet in that each term in the sentiment lexicon contains triplet polarity scores, POS tag, and gloss definitions for some terms. The Arabic sentiment lexicon does not contain different dialects, compound expressions for words, phrases, and proverbs.

Similarly, the work of Chen et. al [33] proposed to build high-quality sentiment lexicons for 136 major languages by exploiting and integrating a wide variety of linguistic resources i.e.

100000 most frequently used words in wiktionary and 12 sentiment lexical resources of 5 non-English languages including Chinese, German, Arabic, Japanese and Italian using a graph propagation algorithm. In addition, it also integrated Google translators, Transliteration Links, and WordNet. Amharic is one of these 136 languages. However, the number of terms in the generated Amharic lexicon is only 46, which are too few to cover the language of sentiment terms. Besides, most of these terms in the Amharic sentiment lexicon are incorrect.

K. Denecke in [35], exploited the sentiWordNet for multilingual sentiment analysis tested on IMDB's movie review and MPQA news articles data sets. First, a language model using LingPipe identifies the language for a document. If the language of the document differs from English, the document is translated to English using translation software called PROMT excellent Translation Technology. Finally, by employing a lexicon-based multilingual framework, the sentiment class of sentence-level and document level review polarity is determined by interpreting the polarity scores computed from n-grams of the text using SentiWordNet. The classifier's accuracy result shows it does not depend on the processed language and domain. Instead, a significant portion of errors in SentiWordNet affects the performance of sentiment classification. The errors could be due to lack of negation handling, translation errors, spelling errors, ambiguities, and different meanings in the synset. The author recommended enriching SentiWordNet by using WordAffect which will address some of the aforementioned errors associated with SentiWordNet. Analyzing topics and opinion holders in a multilingual context is suggested to be done further.

Similarly, SentiWordnet for Indian languages [34] was built relying on English SentiWordNet, English subjectivity list, and bilingual dictionary. First, English SentiWordNet and subjectivity lists are merged and duplicates are removed. Second, it transferred synsets of this merged lexicon to corresponding three Indian languages (i.e. Hindi, Bengali and Telugu) using language translator systems and using existing dictionaries. And next, Hindi/Bengali WordNet was used to expand the generated sentiWordNets using a dictionary-based approach. Then, an antonym generation method was applied to further ensure the inclusion of antonyms in the sentiment lexicon by using few handcrafted rules. And then, a corpus-based approach applied to capture language-specific and/or culture-specific words. For the corpus-based approach, selected seed words were provided to train Conditional Random Field (CRF) model. Finally, a gaming approach was proposed to validate and testify the credibility of the generated sentiment lexicons for Indian languages. Only Bengali SentiWordNet was evaluated for its coverage using Bengali news review and blog corpora. The lexicons in the rest of the languages were not evaluated for their coverage, as they lack corpora.

In [104], Nusko et. al (2016) built a Swedish sentiment lexicon using bootstrapping approach which allows one to expand six manually selected seed words from SALDO based on parent-child relation in the semantic network of SALDO. The format of the generated sentiment lexicon

includes polarity, strength, word forms and senses, POS, confidence, lemgram[1] frequency, and example[2]. The Swedish sentiment lexicon contains 175 seeds, 633 children, and 1319 grandchildren. One of the major drawbacks of the Swedish sentiment lexicon is that it does not contain terms that are outside of SALDO. The other weakness is that its aforementioned format differs from SentiWordNet's format. This does not easily allow Swedish sentiment lexicon to work with other languages to transfer sentiment information to/from other sentiment lexicons' standard format.

Vania et. al (2014) in [144] generated an Indonesian domain-specific sentiment lexicon automatically using small seed words (translated from English lexicon) relying on resources such as machine translation, user reviews, and part-of-speech (POS) taggers. The seed list is expanded by using user reviews based on handcrafted rules or patterns of POS combinations. The accuracy of the generated sentiment lexicon is 77.7%. The authors recommended using the approaches to any language having a POS tagger to generate domain-specific sentiment lexicon.

In the work of [56], Yulan et. al (2010) exploited three English lexical resources MPQA, Appraisal, and SentiWordnet, by translating them into three Chinese sentiment Lexicons using Google translator. The performance of these generated Chinese lexicons was compared with the Chinese NTUSD sentiment lexicon by implementing them to classify sentiment on Chinese product reviews using SVM and Naive Bayesian classifiers. Another setting, where polarity bearing lexicon's terms in each domain of product review were extracted from translated Chinese lexicon using LDA (Latent Dirichlet Allocation), repeated the experiment. Sentiment lexicons generated by LDA could recognize domain-specific polarity words that did not exist in the original sentiment lexicons. The authors carried the experimental results in different settings out. The combination of the translated MPQA and SentiWordNet performed better than the other combinations on the original Chinese product review. However, this experimental setting performed poorly on translated Chinese corpora because the MPQA subjectivity lexicon provides better sentiment prior information to the LDA for Chinese product review sentiment classification than the other English lexicons. LDA also provides overall better accuracy when compared to SVM and Bayesian networks. The major errors of the generated sentiment lexicon by the proposed system are because of the machine translation system, which can be affected by language gaps between the source and target languages. To address these issues, the authors recommended generating language-specific sentiment lexicons automatically from the large corpus of the language.

Taboada et. al [137] proposed a semantic orientations calculator (SO-CAL) for lexicon-based subjectivity classification. SO-CAL sentiment lexicon includes intensification and negation to calculate and assign semantic orientation of words in the text. The major strength of this

[1] A Lemgram in SALDO is a combination of a base form and an inflectional pattern

[2] Example refers to a sentence to describe the context how the term is used

work is that the sentiment lexicon outperforms consistently in most domains. SO-CAL is also used to extract word level/sentence level opinions in text. SO-CAL is annotated and evaluated using mechanical turk. SO-CAL performed well in sentiment analysis of user's blog postings. SO-CAL is used as one of the baseline sentiment lexicons for assigning sentiment values to newly translated terms.

Medagoda et. al in [93] built SentiWordNet 3.0 for the Sinhala language by mapping English SentiWordNet 3.0 relying on the online English-Sinhala dictionary. The English words in the dictionary were used as search keys to generate a sentiment lexicon for the corresponding translated Sinhala sentiment lexicon. If the translated Sinhala word is found, then it is inserted with its polarity value and POS tag into the Sinhala sentiment lexicon, otherwise search proceeds for the next English word in the dictionary. The final Sinhala sentiment lexicon contains 5973 adjectives and 405 adverbs. This lexicon was evaluated using 2,083 manually classified news article opinions collected from Sinhala online newspapers. Based on the different classification methods using this sentiment lexicon, the accuracy achieved was between 56-60%.

The handling of negation, multi-word expressions with negation and context sensitivity are suggested to be addressed to improve the performance of the lexicon generation. A similar approach is used to generate an Amharic sentiment lexicon, considering the idiosyncrasies of the Amharic language.

2.2.3. Corpus based Lexicon Generation

This subsection presents the key papers related to corpus-based sentiment lexicon generation for sentiment classification of non-Amharic languages.

L. Velikovich et. al (2010) in [145] developed a large polarity lexicon semi-automatically from the web by applying graph propagation methods. A set of positive and negative sentences are prepared from the web for providing clues to expansion of the lexicon. The method assigns a higher positive value if a given seed phrase contains multiple positive seed words, otherwise it is assigned negative value.

Both quantitatively and qualitatively, the web generated lexicon outperformed the other sentiment lexicons generated from manually annotated lexical resources like WordNet.

Another corpus-based work in [53] developed two domain specific sentiment lexicons from historical corpora of 150 years and online community data using word embedding with label propagation algorithm to expand small seeds. This approach achieved competitive performance with hand curated sentiment lexicons. Results revealed that there is sentiment change of words either positive to negative or vice versa through time. They construct a lexical graph using a PPMI matrix computed from word embedding. To fill the edges of two nodes (w_i, w_j), cosine similarity distance is computed. To propagate sentiment from seeds in a lexical graph, a random walk algorithm is used. The generated sentiment lexicon in [53] from domain specific

embedding performs very well when it is compared with the baseline and other variants.
The proposed corpus-based approach is closely related to the work of Lucia et. al in [108]. Lucia et. al (2015) generated an emotion-based lexicon by bootstrapping a corpus using word distributional semantics (i.e. using PPMI). The proposed approach differs from the work of [108] in that the approach generates a sentiment lexicon rather than an emotion lexicon. The approach of propagating sentiment to expand the seeds is also different. Cosine similarity distance between the seed mean vector and the corresponding vectors in the PPMI matrix entries are used.
In this research, corpus-based lexicon generation is proposed to generate lexicons from corpora using co-occurrence counts such as frequency-based methods (i.e. PPMI). Basically, Levy et. al [80] demonstrated that word embedding can implicitly be factorized into a context PMI matrix.

Table 2.1.: Summary of Related Work on Lexicon Generation and Lexicon-based Sentiment Classification

Authors	Methods Used	Average Accuracy	Languages	lexicon Size	Domain
Selama(2010) [49]	Manual to generated lexicon and Rule based Sentence Level SA	94.5% on Movie and 91.4% on News	Amharic	1060	254 movie reviews and 49 News Articles
Tulu (2014) [139]	uses Lexicon in [49] and Rule-based aspect-level SA	85.8% (Average F-score)	Amharic	1001	204 Hotel Review, 100 Hospital Review and 180 University Review
Teshome et. al (2019) [138]	Created Lexicon size of 1298 and Lexicon-based Sentence-level SA	88.9% (Average F-score)	Amharic	1000	800 Facebook comments
Gebremedhin et. al (2018) [51]	lexicon-based using tri-lingual lexicon	average F-score of 85.99%	English-Amharic-Tigregna		564 Facebook and YouTube comments
Mahyoub et. al (2014) [89]	Arabic SentiWordNet generated from Arabic WordNet using bootstrapping by exploiting synset relations and Support Vector Machine was used for SA	96%	Arabic	7400	500 Movie Review and 990 Book Review
Denecke (2008) [35]	English SentiWordNet + machine learning/rules are used as lexical resource for multilingual SA relying on language translation software.	Accuracy on German movie review is between 58%-66%.	Multilingual settings(e.g. German, English, etc.)	100000	535 MPQA news and 2000 IMDB's movie review
Medagoda et. al (2015) [93]	Uses Bilingual dictionary to map English SentiWordNet 3.0 to Sinhala lexicon and Uses ML such as SVM, Naive and J48	Naive Bayes outperformed with 60% accuracy	Sinhala	6378	2,083 manually classified Sinhala online newspaper
Vania et. al (2014) [144]	Built Indonesian domain specific lexicon using small seed sets, online user review, English sentiment lexicon and POS tagger.	Accuracy of 77.7%	Indonesian	3553	1368 TripAdvisor, 3850 OpenRice and 11816 Twitter
Yulan et. al (2010) [56]	Used English MPQA, Appraisal, SentiWordnet and machine translator for Chinese product review SA using weakly supervised approach	Accuracy of 83%	Chinese	varies	5484 product reviews
Taboada et. al (2011) [137]	Built Semantic Orientation CALculator(SO-CAL) sentiment calculator by extracting text	80% on product review (epinions.com)	English	varies	400 product reviews, 1900 movie review and 2400 camera review
Kundi et. al (2014) [75]	Used existing English sentiment lexicons to perform SA of Tweet reviews	92% Binary class and 87% Multi class	English	varies	152647 Tweets of three product reviews
Musto et. al (2014) [99]	Used SentiwordNet,Word Affect,MPQA using lexicon based approach	74.65% with SentiWordNet on STS data sets	English	varies	Stanford Twitter sentiment (STS)
Ding et. al (2008) [40]	Developed holistic lexicon-based SA	Average F-Score is 0.90	English	-	benchmark product review

2.3 Machine Learning Approaches

This section presents sentiment classification related work which employed machine learning approaches (i.e. conventional, ensemble learners and transformers). The key related work of sentiment classification relying on machine learning approaches in both English and other languages are reviewed.

2.3.1. Machine Learning for Sentiment Classification

Recently, the number of research projects on sentiment classification of Amharic and other Ethiopian languages using machine learning have been increasing. First, the existing sentiment research in local languages is reviewed. Then, machine learning-based sentiment classification related work in other languages is reviewed.

In [44], the authors investigated the effect of preprocessing (i.e. emojis feature) improving accuracy by 0.55% than using only sentiment lexicons on 9138 Amharic YouTube comments. The authors also tested the effect of stemming on sentiment classification using LSTM and NB models, the performance of the models is reduced when stemming is applied on YouTube comments. The author also discovered that automatic labeling using sentiment lexicon and emoji helps to reduce the required effort and time of the annotator. In [141], the authors tried to develop sarcasm detection of Amharic texts using SVM, Neural Nets and Random Forest relying on unigram features using information gain, yielding an accuracy of 80.6%, 80.1% and 79%, respectively, on Facebook comments having two categories: 400 sarcasm and 400 non-sarcasm.

In [152], the authors proposed multi-scale (-2,-1,0,+1,+2) Amharic sentiment classification using NB relying on unigram, bigram and hybrid variants, achieving an accuracy of 43.6%, 44.3% and 39.5%, respectively, on 608 labeled blog reviews. Similarly, in [36], the authors tried to develop feature-level Amharic sentiment classification using supervised machine learning algorithms (i.e. SVM and NB) which are trained using 1200 labeled Facebook comments collected from Facebook page of Amhara Mass Media. Using Bag of word features, the precision, recall and F-score of SVM and NB are (84%, 80% and 81%) and (87%, 82% and 84 %), respectively.

In [95], the authors attempted to develop Awngi sentiment classification using SVM and NB by extracting features based on chi-square and tfidf. SVM (acc. of 79%) outperformed NB (acc. of 75%) using Awngi music reviews datasets comprising 865 positive reviews, 393 negative reviews, and 242 neutral reviews. In [150], the authors proposed deep learning approaches (CNN and LSTM) for sentiment classification of 1452 Afaan Oromo Facebook comments, achieving an accuracy of 89% and 87.6%, respectively.

In [30], Chaovalit et. al (2005) investigated opinion mining using a machine learning approach

(corpus-based) compared to an unsupervised approach (semantic orientation). Their results in both approaches showed they performed comparably or better than existing methods and that preprocessing and feature selection affected the result of machine learning. The authors recommended TFIDF weighting feature as input to the supervised approach since TFIDF weighting reduces the number of features and sparseness of feature values. The authors also suggested identifying opinionated training datasets by removing factual datasets using sentiment lexicon and part-of-speech linguistic features. To improve semantic orientation, instead of using the opinion words "excellent" and "poor" as seed for movie review opinion mining, it is better to select more representative opinion words related to the movie domain. In addition, two-word phrases can also enhance the performance of semantic orientation.

Below are key related work relying on machine learning approaches in non-local languages. In [133], authors investigated state-of-the-art machine learning classifiers (Naïve Bayes, J48, BFTree and OneR) for optimization of sentiment analysis on two datasets in English texts. The first dataset was collected from Amazon and the second is assembled from IMDB movie reviews. They compared the four above classifiers. Naive Bayes is the fastest learning algorithm, whereas OneR outperformed the others with the accuracy of 91.3% in precision, 97% in F-measure and 92.34%.

In [128], Shamantha et. al (2019) developed a machine learning classifier for sentiment classification of English Twitter (i.e. either positive, negative or neutral). They selected word features as input to classifiers (NB, SVM and RF).

In [43], the authors proposed machine learning approaches (i.e. NB, SVM, KNN) for predicting sentiment of Arabic user generated 2591 tweet comments which are labeled either positive, negative or neutral with crowd-sourcing tools. With TFIDF features, the precision (i.e 75.25) of SVM is relatively better than the other classifiers. In [140], the authors compared performances of sentiment classification (i.e. positive or negative) relying on machine learning algorithms (NB and SVM) on 2000 labeled polarity movie datasets. With TFIDF features, SVM achieved an accuracy of 0.94 with 10 cross-fold validations.

Table 2.2.: Summary of Related Work of Machine Learning-based Sentiment Classification

Authors	ML approach	Average Accuracy	Languages	Domain
Fikre et. al (2020) [44]	ML-based: Investigated the effect of emoji and stemming on sentiment classification	With emoji 86.98% with LSTM and 91.74% with NB and Without stem 82.36% with LSTM and 90.7% with NB	Amharic	9,138 YouTube Comments
Agidaw et. al (2019) [141]	Proposed SVM, Neural Net. and RF on sarcasm classification	With unigrams, SVM, NN and RF has acc. of 80.6%, 80.1% and 79% resp.	Amharic	800 Abebe Tolla's "Mitsetoch" book and Comments
Philemon et. al [152]	Proposed NB for multi-scale sentiment classification	With bigram, NB performed accuracy of 44.3%	Amharic	608 labeled blog reviews.
Mengesha et. al (2019) [36]	Machine Learning: NB and SVM	With BoW features, NB and SVM yield F-score of 84% and 81%, respectively	Amharic	1200 labeled Facebook comments
Mihret et. al (2019) [95]	ML approaches i.e. NB and SVM	With chi-square and tfidf, SVM and NB achieved accuracy of 79% and 75%, respectively	Awngi	1500 labeled music reviews
Rase (2020) [150]	DL approaches i.e. CNN and LSTM	CNN and LSTM models achieving accuracy of 89% and 87.6%, respectively	Afaan Oromo	1452 Facebook comments
Singh et. al (2017) [133]	ML approaches i.e. NB, J48, BFTree and OneR	OneR achieving accuracy of 91.3%	English	Amazon reviews and IMDB movie reviews
Chaovalit et. al (2005) [30]	Proposed machine Learning (ML) based and using Semantic Orientations (SO) of movie review domain	85.54% with ML and 77% with SO	English	1400 IMDB's movie reviews

2.3.2. Ensemble Learning

In this subsection, some related work on ensemble learning is presented.

In [100], the authors proposed ensemble methods for the problem of subjectivity and sentiment classification of Arabic texts. The ensemble classifiers combine base learners (Rochio, SVM and NB). The results show that the ensemble classifier performed better than the performance of base learners.

In [70], Khalid et. al (2020) proposed an ensemble learning that combines gradient boosting and support vector machines using a voting classifier. They called it Gradient Boosted Support Vector Machine (GBSVM) for sentiment classification. With term-frequency and TFIDF (uni-, bi-, and tri-gram) features, the approach is evaluated on a dataset of 64,295 Google App user reviews, which are labeled either positive, negative or neutral. The proposed GB-SVM is achieving the highest accuracy of 0.93 with term-frequency feature compared to the other variant TF-IDF features and compared to individual classifiers (i.e. GB, SVM, LR,RF) performances.

In [148], Wan et. al (2015) developed an ensemble of NB, SVM, Bayesian Net, C4.5, and RF using majority voting for sentiment classification of airline customer service feedback. The proposed ensemble outperformed with F-score of 84.2% for three class dataset and 91.7% for two class dataset using 10 fold CV of 12864 tweets. The performance of the ensemble on sentiment classification improves as compared to the individual classifiers.

In [149], Wan et. al (2014) assessed and compared the three most popular ensemble approaches (i.e. Bagging, Boosting, and Random Subspace) relying on base learners (i.e. NB, Maximum

Entropy, Decision Tree, KNN, and SVM) for sentiment classification using BoW features of ten publicly available datasets. According to 1200 comparative group experiments, the results proved that ensemble learners improve performance of base learners for sentiment classification. Random Subspace has shown the best comparative performance of all the three ensemble learners.

In [12], Alnashwan et. al (2016) proposed an ensemble approach that is designed to improve sentiment classification relying on base learners (SVM, NB, LR and RF). The base learners take sentiment lexicon features for the three Twitter datasets as input for training. The ensemble reduces error rate by using the lexicon resource. This avoids selection of poorly performing base learner. The performance of the proposed ensemble is better than the individual learners; however, the proposed ensemble (i.e. accuracy of 83.2%) has no significant difference with the performance of decision random (i.e. accuracy of 83.4%).

In [55], Hassan et. al (2013) developed a bootstrap ensemble framework for sentiment classification of English Tweets. The intention of the proposed framework is to tackle the challenges of Twitter sentiment classification, such as class imbalance, sparsity, and representational richness issues. In the framework, first text features are extracted, then seven different machine learning algorithms are trained, finally selected models are combined. This proposed approach has shown more accurate and balanced prediction of sentiment of Twitter compared to other algorithms.

Yet another similar approach was done by [17] to integrate two or more unsupervised approaches using meta-classifiers for Spanish movie review sentiment classification. Stacking classifiers (SVM, Naïve Bayesian, Logistic Regression) were used as the meta-learners to combine the unsupervised models. The integrated system using stacking (Naïve Bayesian) performed better than the individual models.

Table 2.3.: Summary of Related Work of Ensemble-based Sentiment Classification

Authors	ML approach	Average Accuracy	Languages	Domain
Khalid et. al (2020) [70]	Ensemble-based: GBSVM i.e. ensemble learner combines gradient boosting and SVM using voting classifier for sentiment classification.	With term-frequency, GBSVM outperformed with accuracy of 93%.	English	64,295 Google App user reviews
Wan et. al (2015) [148]	Ensemble Learning: NB, SVM, Bayesian Net, C4.5, and RF	The results of F-score: 84.2% for three class dataset and 91.7% for two class dataset	English	12864 tweets of Airline customer feedback

2.3.3. Transfer Learning Approaches

This section presents a review of transfer learning for sentiment classification.

In [107], transfer learning is defined as "the method that uses the similarity of data, data distribution, model, task and so on to apply the knowledge already learned in one domain to the new domain." A model is constructed in one domain and used in a new domain to increase the

annotation of unlabeled data in the target domain. For sentiment transfer learning, three types of transfer learning are applied: parameter transfer methods (that transfers knowledge from one source to target domain through trained model parameters), instance-transfer methods (that shares data from both source and target domain) and feature-representation transfer methods (that transforms data from both source and target domain to same feature space) [83].

Using parametric transfer learning for sentiment classification, parameters are leveraged by sharing from the trained model in the source domain to the models in the target domain where there is scarcity of labeled training data. This is how transfer learning is proposed to address the problem of scarcity of labeled sentiment corpora to carry out Amharic sentiment classification. The simplest parameter transfer method applied in NLP tasks is word2Vec embeddings pretrained model which can transfer the knowledge stored in parameters of the first layer of a model [96].

Word embeddings are a representation of words linked to numerical vector representations that can capture the syntactic (i.e. arrangement of words) and semantic (i.e. meanings of words) relying on the language modeling objective function which can be used to discriminate correct from incorrect words in left and right contexts of a target word [96, 37].

This addresses the drawback of bag-of-word models which can be used to capture the semantics of words in a text [73]. Various word embedding methods have been developed that maps unigrams into vector representations for input to machine learning algorithms. The most common word embedding models have successfully used deep learning algorithms for NLP applications including Word2Vec [96], GloVe [109], and FastText [109].

The word-level embedding approaches have been extended to coarser granularity of a text including sentence embeddings and paragraph embeddings. These word embeddings are also called static word embeddings. The representation in static word embeddings can not handle the varied meanings of a given word in a text. The word representation captures only one meaning of a word as its unique vector representation. Thus, static word embeddings can not address polysemy (i.e. the different meanings of a text) [73].

To date, more expressive contextualized language models have been developed to address the aforementioned problem of static word embeddings. The meaning of a word in a text depends on both the word identity and its surrounding words at that moment. This means, a given word has dynamically changing representations relying on context at a particular time. This approach is developed with the objective to model the probabilistic distribution of word sequences of a particular language. This is called language modeling. Thus, given a context, a language model can predict the probability of a word in that context. Some of contextualized language models which can be applied pre-trained language models to downstream NLP applications include ELMo [110], BERT [37] and GPT [116].

In this research, BERT transfer learning for Amharic sentiment classification is proposed. The reason is that a BERT pre-trained language model is suited for down stream NLP tasks like

sentiment classification in small size labeled data set As Amharic has limited labeled dataset for SC and BERT has not been built for Amharic language yet. Below is a review of BERT for NLP tasks (particularly, sentiment classification).

In [41], the authors present an overview of BERT and show how to create and fine-tune BERT for sentiment classification and illustrated the effectiveness of BERT on 500,000 Amazon fine food reviews labeled either positive or negative. With three epochs, the model performed with an accuracy of 98%.

In [87], the authors presented a BERT model that relies on transfer learning for Twitter sentiment classification for disaster management for proper usage to emergency response using different disaster related data (e.g. CrisisNLP). The customized BERT approach outperformed the baseline (Glove + LSTM) by 3.29% in terms of F-score using CrisisNLP 74346 labeled Tweets in English language. Ambiguity and noise in the texts on disaster twitter data degrades the classification performance.

In [113], Pota et. al (2021) proposed BERT-based Italian Tweet sentiment classification in a two-step pipeline. The first step is building a BERT base small pre-trained Italian language model using Italian OPUS corpora (of 13GB). The second step is training the pre-trained model in step one by using the labeled Italian Twitter dataset, and its sentiment classification performance is compared with other existing solutions. This approach shows improvement by average F-score of 3% over the baseline ALBERTO model (the overall average F-score of the new model is 75% whereas the F-score of ALBERTO is 72.23%) using SENTIPOLC 2016 labeled Twitter dataset.

In [65], Karimi et. al (2019) fine-tuned the pre-trained BERT model for sentiment classification of Persian texts. The performance of the fine-tuned pre-trained model for sentiment classification has gained accuracy improvement compared to the performance of classifier built from scratch.

In [15], the authors extended the BERT-based model to Arabic BERT (AraBERT) for Arabic language understanding. The pre-trained AraBERT has achieved and worked efficiently in most NLP tasks (such as Sentiment Analysis, Named Entity Recognition, and Question Answering) as compared to the multilingual BERT from Google as baseline. As Amharic is a Semitic language and has the same linguistic characteristics as Arabic, this research is proposed to extend the BERT-base model for developing pre-trained BERT for Amharic language.

Table 2.4.: Summary of Related Work of BERT-based Sentiment Classification

Authors	ML approach	Average Accuracy	Languages	Domain
Doug et. al (2020) [41]	Fine-tuned BERT for sentiment classification	98%	English	500,000 Amazon fine food reviews
Ma et. al (2019) [87]	BERT-based transfer learning for tweeter disaster sentiment classification	–	English	CrisisNLP 74346 labeled Tweets
Pota et. al (2021) [113]	BERT-based transfer learning for Italian tweet sentiment classification	F-score of 75%	Italian	SENTIPOLC 2016 labeled tweeter dataset
Antoun et. al (2020) [15]	BERT for Arabic language understanding (e.g. Sentiment classification)	–	Arabic	cover different genres, domains and dialects (e.g. ASTD 10,000 Egyptian dialect tweets).

2.4 Hybrid Approaches

In this section, a review of hybrid based sentiment classification is briefly presented.
In [131], the authors developed a hybrid sentiment classification for Amharic book reviews. In this work, the authors tried to combine a lexicon-based classifier, i.e. using count of opinion words and a machine learning based classifier, i.e. using feature selection methods such as chi-square, mutual information gain, and Galavvotti-Sebastiani-Simi coefficient. Three supervised algorithms (NB, LR, and SVM) were tested on 600 Amharic book reviews, which are labeled either positive or negative. With this dataset, the lexicon-based classifier, NB with information gain (i.e. accuracy 93.33%) performed the best as compared to the lexicon-based (i.e. accuracy of 74%) and hybrid classifier (i.e. accuracy of 87%).
Hsu [62] analyzed theoretically why hybrid ensembles work to enhance performance using different algorithms. The author also experimentally tested whether combining NB and decision tree showed an accuracy gain as compared with base algorithms and provided a theoretical foundation for building and applying hybrid ensembles. Hsu showed that diversified classifiers in a hybrid ensemble are superior or comparable to non-hybrid ensembles. Diversity among algorithms in a hybrid ensemble leads to gain in performance as the base classifiers complemented the errors made by themselves while working together.
In [7], the authors proposed a hybrid approach for sentiment classification of the Universiti Utara Malaysia (UUM) Facebook comments. Their approach, which combined lexicon-based and machine learning-based (i.e. NB and SVM) approaches outperformed the individual approaches. In particular, the lexicon-based approach with SVM is the best of all the approaches with accuracy of 90% as compared to the individual approaches i.e. SVM with accuracy of 80% and lexicon-based with accuracy of 85%.
In [8], Aldayel et. al developed a hybrid scheme for Arabic tweet sentiment classifications, combining unsupervised (semantic orientations) and supervised approaches (SVM). The hybrid is formed by feeding outputs of the lexicon-based approach as input features to the SVM classifier. The accuracy of the hybrid approach is improved by 16.41% from the individual lexicon-based approach, i.e. with overall accuracy of 84%. In addition, the authors investigated

the effects of preprocessing steps (cleaning, normalization, stop word removal, negation and light stemmer) that revealed the increase of accuracy of lexicon-based sentiment classification of Arabic tweets. However, the effect of unigram, bi and tri-gram word level features did not show significant performance change on the Arabic tweet classification.

In [6], Al Amrani et. al proposed a hybrid approach combining RF and SVM for sentiment classification, i.e. to classify into one of the classes—negative and positive. The hybrid classifier outperformed the base classifiers on 1000 product reviews (i.e. labeled 500 positive and 500 negative examples) with 10 fold cross-validation (CV) by an increase of accuracy of 1% (with an overall accuracy of 83.4%).

In [124], Sarkar (2020) developed a hybrid of three heterogeneous classifiers for sentiment polarity detection of Bengali and Hindi tweets by augmenting external sentiment lexicon features with word and char ngrams. The first and second base classifier is the same (i.e. NB), but the input for the first one combines traditional word n-gram features and sentiment lexicon features, the second one takes char ngram and sentiment lexicon features, and the third classifier is linear SVM taking input features—unigram and sentiment lexicon features. The author used three deep learning models—Convolution Neural Network (CNN), Long Short-term Memory Model (LSTM) and Bidirectional (BiLSTM) as baselines. With 10 folds CV, the proposed model was tested on the SAIL 2015 dataset (1500 Bengali tweets and 1760 Hindi tweets) with ensemble combination strategies—majority voting, averaging probabilities and max rule. The hybrid classifier with majority voting outperformed the BiLSTM by a gain of 1.8% accuracy on the Bengali dataset (i.e. overall accuracy of 57.53%) whereas with the Hindi dataset, the hybrid approach gained 2.03% accuracy over the BiLSTM by using averaging probabilities (i.e. overall accuracy of 62.63%).

In [54] Handhika et. al combined a lexicon-based approach using SenticNet and machine learning-based approaches that select the best performing homogeneous algorithms—Bagged Decision Trees, Logistic Model Tree, and Bagged Multi-layer Perceptron algorithms. The output of the lexicon-based approach is the input to the homogeneous classifier. Among the homogeneous algorithms, RF had the highest average accuracy of 98% on the Bali tourist attractions review dataset.

Table 2.5.: Summary of Related Work of Hybrid based Sentiment Classification

Authors	Methods Used	Average Accuracy	Languages	Domain
Shikur et. al (2019) [131]	Hybrid that combined lexicon-based (i.e. term counting) and machine learning (i.e. LR, NB, SVM)	87% (Best)	Amharic	600 Book reviews
Kamil et. al (2020) [7]	hybrid approach for sentiment classification	90%	Malay	UUM Facebook comments.
Aldayel et. al (2016) [8]	Combined lexicon-based and machine learning approaches for Arabic tweet sentiment classification	84%	Arabic	Tweets.
Al Amrani et. al (2018) [6]	Combines RF and SVM for Arabic tweet sentiment classification	83.4%	English	1000 product reviews
Handhika et. al (2019) [54]	Hybrid approach for sentiment classification	98%	English	Bali tourist attractions review dataset.
Kennedy et. al (2006) [69]	Hybrid that combined lexicon based (i.e. term counting) and machine learning (i.e. SVM)	86.2% (Best)	English	2000 movie reviews (IMDB)
Teresa et. al (2012) [91]	Hybrid which combined lexicon based (i.e. Semantic Orientation) and machine learning (i.e. SVM).	88.28% (Best F-score)	English-Spanish	3878 movie reviews (MuchoCine)
Balahur et. al (2012) [20]	Proposed supervised approach for Multilingual SA relying on machine translation services.	66.6% (best bagging for Spanish)	English-French, German Spanish	6,165 sentences (MOAT) and 6,223 sentences (MOAT)
Martinez et. al (2014) [17]	Integrated two or more unsupervised using meta-classifiers	64.68% (best in the combined approach)	Spanish	3878 movie reviews(MuchoCine)

2.5 Summary of the Review

In summary, as discussed in section 1.2 of chapter 1, one of the major challenges which is identified in less-resourced languages is lack of a large-scale sentiment lexicon [121].
In Amharic, there is only the manually built sentiment lexicon by Selama Gebremeskel [49] for lexicon based sentiment classification. The size of this lexicon is 1060. This lexicon has its own pros and cons. To mention a few, the lexicon is relatively accurate as it is manually built and can be used as a baseline. However, it has also short-comings: (i) it is too small to cover all sentiment words in the lexicon, (ii) it is labeled either positive and negative, thus does not show the strength of sentiment words by using numerical values, (iii) it does not contain glosses, and part-of-speech tags, and (iv) it is partly domain specific, for example, covering movie reviews. To remedy these problems, dictionary based and corpus based Amharic sentiment lexicon generation is proposed. This is connected to the research questions RQ1 and RQ2. And the proposed solutions, experiment, results, and evaluations are presented in chapter 6.
To highlight research gaps in the above related work employing supervised machine learning approaches [36, 95, 150, 141], for example, Mengesha (2019) [36] prepared 1200 Facebook comments and used it to train NB and SVM with BoW features to classify sentiment of Amharic texts. The author in [36] reported that NB performed better than SVM with 3% accuracy. The author did not report why SVM and NB is chosen and why BoW feature is selected over other feature sets.
This research proposes to test which of the text feature sets are best and which of the machine learning approaches are best at Amharic sentiment classification. This is associated with the stated research question RQ3. details in chapter 7.1.

Text Features Selection for Amharic SC: Among the text feature sets (i.e. Bag-Of-Words (BOW), TF-IDF char/word ngram feature set, average word2Vec, TF-IDFWeighted word2Vec, LDA, LSI and NMF), the best possible features for machine learning algorithms for Amharic sentiment classification have not been evaluated. Investigation of the best feature set for distinguishing the sentiment of Amharic texts can be taken as research gaps. The discrimination power of various text features sets on detecting sentiment of Amharic texts is further covered in this research. This is related to research question RQ3. The corresponding experiment and results are reported in chapter 7.1.

Machine Learning Algorithm Selection for Amharic SC: We can take machine learning selection for Amharic SC as research gaps which covered for sentiment in Amharic texts. In line with machine learning for Amharic SC, the research gaps are as follows.

- The machine learning algorithms for sentiment classification in recent years have been surveyed by Ebru (2016) et. al in [18]. The most widely used algorithms include SVM, NB, RF, LR, etc. The type of machine learning algorithms that best suit modeling Amharic sentiment classification have not been studied and identified.

- Hyper-parameter tuning is used to find the best hyper-parameter value(s) of a machine learning algorithm at which the machine learning algorithm achieves high accuracy of classification [50, 5]. However, hyper-parameter tuning of the machine learning algorithms for Amharic sentiment classification were not investigated in the existing works. This is linked to research question RQ4.

Improving Amharic SC: Though some studies achieved a high success rate in improving the accuracy of sentiment classification, accuracy of all sentiment analysis techniques is the primary problem in almost in all languages [18]. The performance of sentiment classification techniques of the existing works needs an improvement.

Below are the research gaps which need to be addressed for improving the performance of sentiment classification of Amharic texts under less-resourced settings.

- In several sentiment classification projects on other languages like Arabic in [118, 100] and English in [117, 70, 148, 149, 12, 55], ensemble methods which combine different classifiers for sentiment classification tasks have reported better performance than the individual classifiers. However, ensemble learning for Amharic sentiment classification has never been studied. This is a research gap to be answered in relation to research question RQ6.

- BERT based fine-tuning for different tasks of NLP and sentiment classification of different languages has shown dramatic performance gains, as in [37, 135, 41, 87] for English and in [15] for Arabic, just to name a few. However, BERT based Transfer learning has never

been used for sentiment classification of Amharic texts. That is why research question RQ5 is stated.

- A hybrid approach to sentiment classification has reported performance improvements in different projects like in [62, 7, 8, 6, 54] for English and in [124] for Hindi language. However, a hybrid approach (voting, averaging and blending) to Amharic sentiment classification has never been seen in existing research. This is a research gap to be addressed by the research question RQ6.

The main objective of this research is building an effective hybrid approach relying on a combination of machine learning approaches, contextualized language models, and stacking ensemble for improving the accuracy of Amharic sentiment classification.
As discussed earlier, Muisa (2019) [131] used a 600 book review dataset and used the manual sentiment lexicon of Selama [49] to label unlabeled corpora and use them for input to a machine learning algorithm for sentiment classification. The author combined a lexicon-based classifier i.e. using count of opinion words and machine learning based classifiers (such as NB, LR, and SVM) using features selected by methods such as chi-square, mutual information gain, and Galavvotti-Sebastiani-Simi coefficient. With this data set, the lexicon-based classifier, NB with information gain (i.e. accuracy 93.33%) performed the best as compared to the lexicon-based (i.e. accuracy of 74%) and hybrid classifiers (i.e.accuracy of 87%).
One of the sources of the errors in the hybrid approach is that the lexicon based approach performs poorly and it propagates the errors to the hybrid approach. The other reason might be that the manually built sentiment lexicon is too small to work and might not contain sentiment words used in the book review domain. The hybrid approach performs below the individual approaches, i.e. lexicon based and machine learning based classifiers.
However, the intention of building hybrid approach is to enhance the performance of the classification task. A new hybrid approach is proposed by combining multiple classification approaches. In addition, the proposed hybrid approaches, such as voting, averaging and blending have never been tested for Amharic sentiment classification. This is linked to research question RQ6 (Discussed in chapter 8).
Almost all work was carried out in the social media domains (Facebook, You tube and blogs). This shows the development of the area of sentiment analysis on opinionated texts expressed in Ethiopian languages and the increasing presence of these languages in social media. As these languages are less-resourced in terms of lack of availability of comprehensive sentiment lexicons and sufficient labeled sentiment corpora, there is yet a challenge to apply sentiment classification since it is difficult to make generalizations.
In summary, the research gaps which are identified based on the existing related work are as follows.

- The Amharic sentiment lexicon needs to be extended to cover more sentiment words of Amharic language.
- The Amharic sentiment lexicon needs to contain sentiment words with numerical sentiment strength rather than a binary label.
- The Amharic sentiment lexicon needs to contain sentiment words with PoS.
- Feature set and machine learning selection, hyper-parameter tuning for Amharic SC have not been carried out.
- Ensemble learning, BERT transfer learning and hybrid approach have not been tested for Amharic SC.

The above research gaps confirm that research questions (RQ1,RQ2, RQ3, RQ4, RQ5, and RQ6) are not answered in the earlier sentiment classification research Amharic.
The general research workflow to address the formulated objectives corresponding to these research gaps is presented in the next chapter.

3. Workflow to Amharic Sentiment Classification

This chapter is structured into five sections. The first section overviews the workflow for Amharic sentiment classification. The second section highlights data preparation and text preprocessing is introduced in the third section. The fourth section states the lexicon generation for Amharic language. The fifth section overviews machine learning.

3.1 Overview

In this section, the research workflow to Amharic sentiment classification system pipeline that shows the high level view of the proposed solution based on the design science methodology is depicted. The intention of the workflow is to facilitate performing sentiment classification of Amharic texts generated in social media, particularly Facebook. The workflow is a high level conceptual view showing the stages of sentiment classification of Amharic. However, the research workflow is not necessarily to automate the entire process of sentiment classification [67]. It is composed of different number of components in each stage for Amharic sentiment classification (SC) system as it is shown in Figure 3.1.

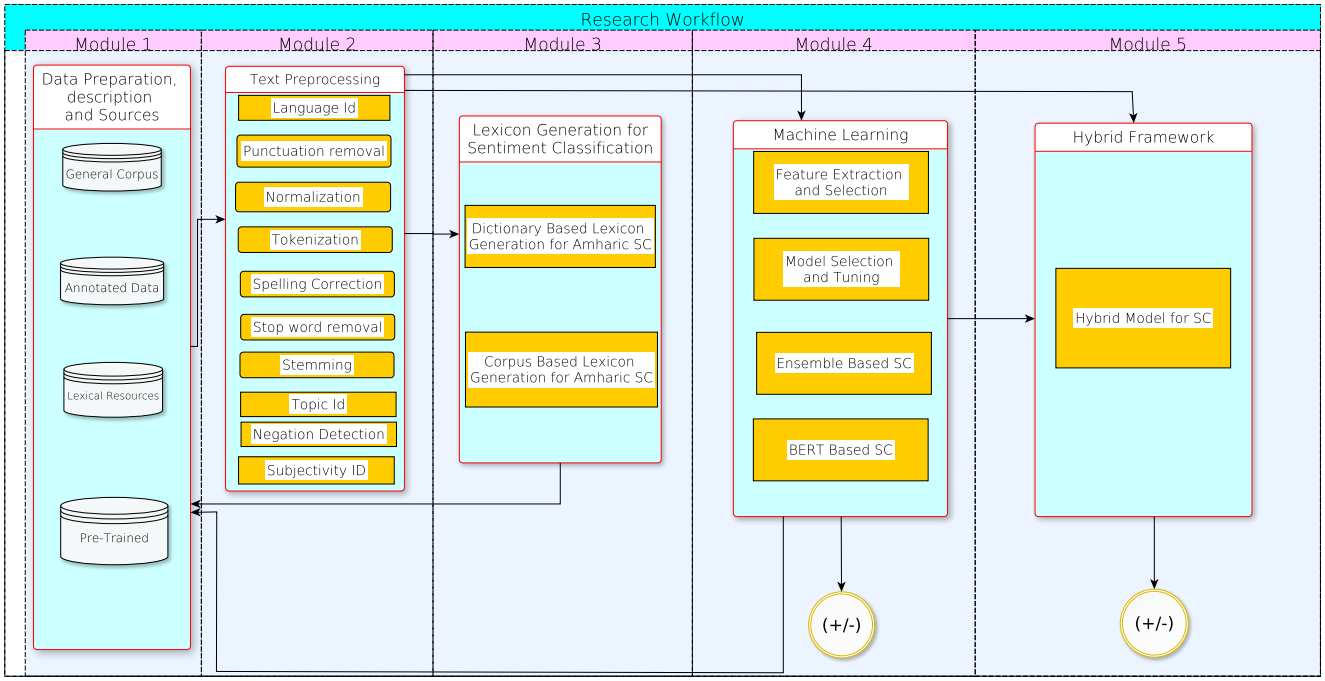


Figure 3.1.: The research workflow of proposed approaches to Amharic sentiment classification

The components in the workflow of Figure 3.1 are represented in modules as: Amharic Data Preparation (Module 1), Amharic Text Preprocessing (Module 2), Lexicon Generation for Amharic Sentiment Classification (Module 3), Machine Learning for Amharic Sentiment Classification (Module 4) and Hybrid Approach for Amharic Sentiment Classification (Module 5) which are described as follows.

3.2 Data Preparation

This module consists of data preparation such as collecting raw data from sources like Facebook. Different corpora, lexical resources and pre-trained models are described in this module. The resources described are used in further modules (stages) of the research workflow.

3.3 Amharic Text Preprocessing

The data sets which are inputs to the workflow at each stage are usually web collection of Amharic texts. The data sets needed at different stages of the workflow require different preprocessing operations. Depending on the type of task in each module, different preprocessing operations are performed. For example, input feature representation for conventional machine learning like SVM for sentiment classification is different from the input feature set needed by BERT. The preprocessing applied to the raw texts for these two models (e.g. SVM needs TF-IDF or BoW features whereas BERT needs word-piece positional embeddings). The type

of preprocessing applied depends on the nature of input vector representation for the machine learning algorithm. The preprocessing operations like punctuation mark removal, stop-word removal, subjectivity detection etc. of that particular language (i.e. Amharic) are applied. The preprocessing module (i.e. Module 2) represents the preprocessing pipeline of Amharic including language identification, punctuation mark removal, character normalization, spelling corrections, tokenization (word and sentence/clause), stopwords removal, stemming, topic detection and subjectivity detection. For each of the tasks in this pipeline, algorithms are developed and implemented. The details of the preprocessing tasks will be presented (chapter 5).

3.4 Lexicon Generation for Sentiment Classification

In module 3 of the workflow, Amharic sentiment lexicons are built by using dictionary-based and corpus-based approaches. The first task of module 3 generates Amharic sentiment lexicon through dictionary based approach from English SentiWordNet and English SO-CAL by using Amharic-English bilingual dictionary. The corpus based approach is also used to generate Amharic sentiment lexicon directly from the collection of Amharic text corpora, by expanding from seeds of Amharic. This task by itself can be a nested design science research cycle and it, in turn, carried out iteratively. Generation of Amharic sentiment lexicons can be built by following the design science research process (starting from awareness of problem to conclusion). The artifact of an iteration can be the Amharic sentiment lexicons, and evaluation data sets. The generated sentiment lexicons is also used for subjectivity detection and negation handling. The details of these Amharic sentiment lexicons generations, algorithms, evaluations and discussions are presented (chapter 6).

3.5 Machine Learning for Sentiment Classification

As feature extraction, machine learning algorithms and evaluation metrics are used in subsequent sections, highlight of basic concepts of machine learning, feature extraction methods and evaluation metrics are presented as follows.

3.5.1. Overview of Machine Learning

Machine Learning is a subfield of artificial intelligence which deals with developing models by learning the patterns and features of training data. Machine learning is also defined as “optimizing a performance criterion using example data and experience” [29]. In other definition, machine learning is defined as a computer program which can be specified by learning experience E, regarding a task T and its performance measure P if its performance is improved with E on T measured by P [115]. It has a wide range of applications in various problem domains,

including medicine, face recognition, pattern recognition, character recognition, advertising, business intelligence, engineering, marketing and manufacturing. In machine learning, models and algorithms are confusing terms that are interchangeably used. However, model is the specific representations learned from data whereas algorithm is learning the feature representation from data [24].
A particular algorithm takes in a set of data as input and yields a model. Symbolically,

$$Model = Algorithm(Data) \tag{3.1}$$

Data plays a significant role in machine learning model and the algorithm is used to uncover the hidden knowledge and feature representations in the data samples. The learning and the prediction performance of a model is determined by the quality and the quantity of the data sets [24]. There are two categories of data, labeled data and unlabeled data. Suppose D denotes a data set with N samples. Each sample is denoted by a feature set(X). The nth data sample element of X is denoted by x^n where each sample has a d-dimensional vector. Mathematically,

$$x^{(n)} = [x_1^{(n)}, x_2^{(n)}, x_3^{(n)}, ..., x_d^{(n)}]^T \tag{3.2}$$

where, $x^{(n)}$ is called a feature vector or feature sample where each dimension of a feature vector is called an attribute, feature, variable or element. If the data set is annotated, it has label class. Assume Y is the corresponding labeled set. Each element of Y corresponds to the label of the feature vector x. The label of the n^{th} feature vector is represented by $x^{(n)}$ and the corresponding target class is represented by $y^{(n)}$. The notations for labeled and unlabeled data sets are as follows.

- (i) Labeled data D:= $\{x^{(i)}, y^{(i)}\}_{i=1}^N$
- (ii) Unlabeled data D:= $\{x^{(i)}\}_{i=1}^N$

In machine learning, the goal is to develop a model by training the properties or knowledge of the training set. Finally, it is tested using testing sets. This research investigates to what extent the model predicts the unseen data samples. The training and testing data sets are sampled from universal data set in the problem domain [29].
Machine Learning is roughly categorized into four groups. These are introduced as follows.
1. Supervised Learning: Supervised learning finds relationships between feature sets and label sets relying on the labeled data sets. In supervised learning, the input variable X is mapped to an output variable by a function f. Mathematically,.

$$Y = f(X) \tag{3.3}$$

The goal is to approximate the mapping function to correctly predict its output when new

input is provided. The algorithm is trained by the training data iteratively until acceptable performance is obtained. Supervised learning is classified into classification problem (if output variable is categorical) and regression problem (if the output variable is real number). Examples of supervised learning include Linear Regression (LR), Random Forest (RF), Support Vector Machine (SVM), Naïve Bayes (NB), just to name a few. For this research, supervised approaches are proposed for Amharic sentiment classification, where there are insufficient labeled data sets.

2. Unsupervised learning: Unsupervised learning finds common features and patterns from feature sets relying on unlabeled data sets. Unsupervised learning learns input variable with no output variable. The goal of unsupervised learning is to learn the distribution or the underlying structure of data. Unsupervised learning is divided into association problems (finds rules that describe the data) and clustering problems (groups data relying on inherit similarity of data). Unsupervised approach is also used for building BERT pre-trained model for Amharic language, which will be covered in section 7.3.

3. Reinforcement learning: Reinforcement learning learns a policy from its environment relying on provision of reward and punishment towards achieving the goal.

4. Semi-supervised learning: Semi-supervised learning learns features relying on insufficient labeled data and large unlabeled data). Semi-supervised learning learns in both supervised and unsupervised manner.

The most popular machine learning algorithms are presented next.

Learning Algorithms: For this research, the most common machine learning algorithms are selected for sentiment classification with considerable effectiveness in the past. These include: linear algorithms (e.g. linear and logistic regression) and non-linear algorithms (e.g. SVM, NB and Trees) providing the same input text features in a data set. The algorithms' performance is compared independently. These algorithms are briefly introduced.

Linear regression (LR): is the simplest linear algorithm borrowed from the field of statistics [24]. Linear regression is the most popular machine learning, based on assumptions that there is a relationship between input variables and output variable. It is denoted by,

$$y = c + m_1 \cdot x^i + m_2 \cdot x^i + ... \tag{3.4}$$

provided that input x^i is the i^{th} attribute of x which is either numeric or binary and the output y is numeric and $c, m_1, m_3..$ are coefficients. The algorithm learns a set of parameters provided that the sum of square error $(y_{actual} - y_{estimated})^2$ is minimized. This min. cost error is sensitive to outliers. The solution is to remove outliers and retrain it.

Logistic regression: The term logistic is given as it uses logistic function(also called sigmoid function) as its core method where the function transforms an input value to a real number

between 0 and 1. The function is given by

$$p(y) = \frac{1}{(1 + e^{-y}))} \tag{3.5}$$

If a linear regression function is $y = c + m \cdot x$, its logistic counter part becomes $\frac{1}{(1+e^{-(c+m\cdot x)})}$, that is:

$$\frac{e^{(c+m\cdot x)}}{(1 + e^{(c+m\cdot x)})} \tag{3.6}$$

See its s-shaped graph in Figure 3.2 which shows the difference between linear and logistic regression.

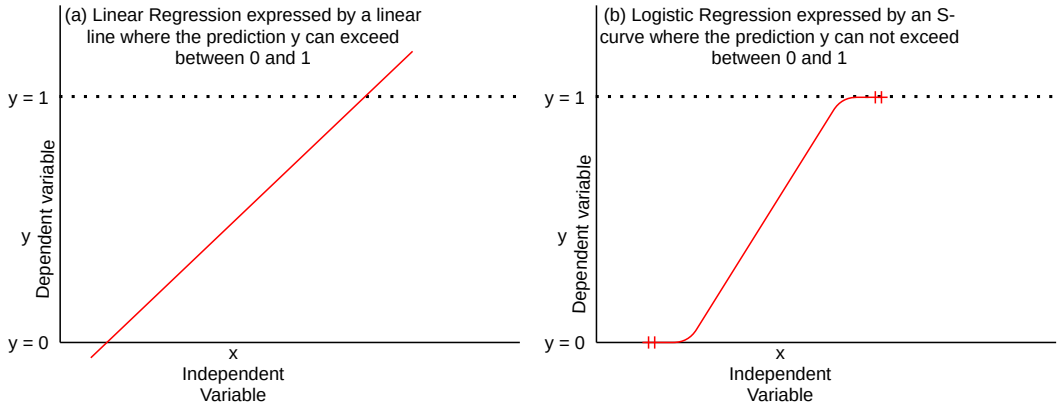


Figure 3.2.: Linear Regression (left) vs Logistic Regression (right)

This algorithm is potentially used to test the classification task prior to testing non-linear algorithms. This helps to understand to the extent to which the problem is complex. The coefficients of this algorithm are estimated from training data relying on maximum likelihood estimation, which is very common in a variety of machine learning algorithms. Logistic regression is used for binary classification. During prediction by this algorithm, the best coefficient is close to 1 (for one class) or close to 0 (for the other class). For example, in gender classification, application prediction is 0 if $probability(person = male) < 0.5$, or prediction is 1 if $probability(person = male) >= 0.5$.

The drawbacks of logistic regression includes (i) only used for binary classification, (ii) bad if data is noisy or has outliers, (iii) assumes a linear relationship of input and output, (iv) likelihood estimation may not converge and (v) it does not work for highly correlated cases.

Naive Bayes (NB): is a probabilistic approach that relies on Bayes' rule where the input features are assumed to independently determine the output variable. Even though this approach worked well in many problems, this independence assumption is rare in reality. The other strength is that it can learn incrementally, scalably and update its probability distribution [84, 59, 24]. This research uses multinominal Naive Bayes, which models the same probability. It is given

by:

$$P(y|x) = \frac{P(x|y) * P(y)}{P(x)} \tag{3.7}$$

where P(y|x) is posterior, P(x|y) is likelihood probability, P(y) is class prior probability and P(x) is predictor prior (marginal probability). For maximized P(y|x),

$$P(y|x) = P(x_1|y) * P(x_2|y) * P(x_3|y) * ... * P(x_n|y) * P(y) \tag{3.8}$$

It is noted that the maximized probability is obtained by discarding the divisor probability as it is the same for both tags (i.e. positive and negative class)
Support Vector Machine (SVM): is the most popular method which; it tries to divide each input features using a hyper-plane by maximizing the distance between data points of positive and negative outputs. Besides maximizing margin distance, it minimizes errors using objective function. It considers a possible hyper-plane, which divides the outputs but optimizes and selects the best hyper-plane with maximum distance from data points of either positive or negative class. A linear problem finds the hyper-plane at

$$c + m_1x_1 + m_2x_2 + .. + m_nx_n = 0 \tag{3.9}$$

This is the decision boundary which used to classify samples (or data points). It is re-written as:

$$W^TX = 0 \tag{3.10}$$

Using this equation, one can find the class of input sample x depending on whether it is above the line or below the line. As depicted in Figure 3.3, all samples less than the negative hyper-plane ($W^TX = -1$) are classified as negative class whereas all samples greater than positive hyper-plane ($W^TX = +1$) are classified as positive class. Samples on either negative hyper-plane ($W^TX = -1$) or positive hyper-plane ($W^TX = -1$) are said to be support vectors. The distance between positive and negative hyper-plane is the margin. The objective function of SVM is to maximize the margin. Terms of SVM such as margins, support vectors, decision boundary and hyper-planes for binary classification tasks are shown.

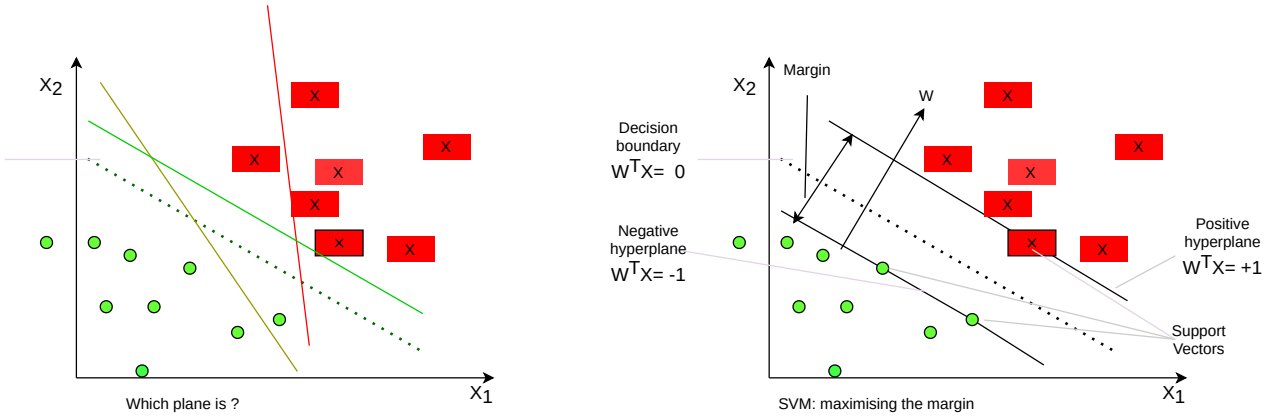


Figure 3.3.: Linear SVM with Hyper-parameters, the green circles and red squares in the plots (left and right) are data samples with negative and positive classes, respectively.

SVM makes use of quadratic programming for optimization of margin distance of data points from hyper-plane. If a dataset is not lineally-separable, SVM kernels (e.g. polynomial, radial) map to higher dimensional space, equivalent to original linear case. These are called kernel tricks that is done depending on the data points. Transformation of data is done in two steps: (i) optimal tuning parameters are found, and (ii) SVM is trained using such optimal parameters. SVM contains the important parameter C, which defines the number of violations in the allowed margin. If C = 0, there is no data point within the margin (i.e. on the opposite side of the hyper-plane). That means no violation. The larger the C, the number of violations allowed. The number of violations affects the placement of hyper-plane. The smaller the value of C, SVM is more sensitive to training data(=low bias and high variance). The larger the value of C, SVM is less sensitive to the training data (= high bias, lower variance). Therefore, there is a balance of bias and variance by tuning C, depending on the training data.
For tuning hyper-parameters of SVM, the kernel parameter is tuned to take “linear”,“poly”,“rbf”, etc. The default kernel for SVM algorithm is “rbf”, which refers to the Radial Basis Function (rbf) kernel. The rbf kernel is a kernel used by SVM to identify non-linearly separable data points. With a kernel, the gamma value is tuned by setting the “gamma” parameter which controls the coefficient of either of kernels. The higher the gamma, the more over-fitted the model. The value of C has been tuned by the “cost” parameter which controls the balance between bias and variance trade-offs relying on training data.
Random Forest(RF): is an ensemble of multiple decision trees (i.e. bagging) with different configurations. This overcomes the limitations of decision trees, which do not update themselves when new training sample is provided. Random forest is robust as it combines multiple tree classifiers relying on n subset of input features in the training set. Finally, it decides by voting for predicting new sample. Figure 3.4 shows how random forest classifier is built.

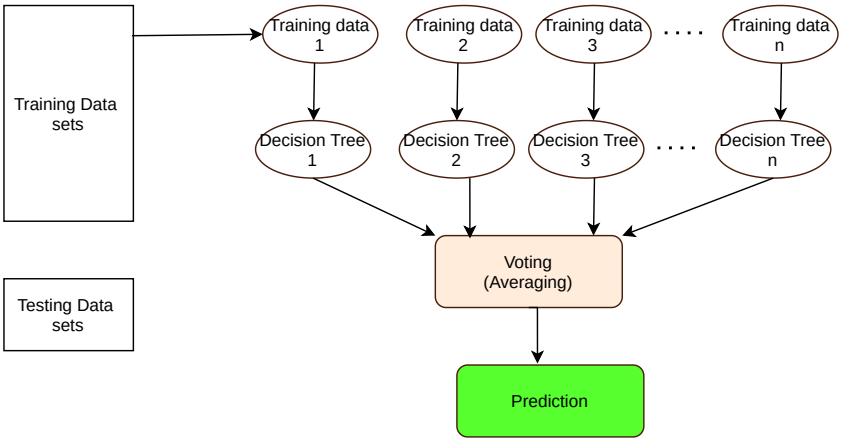


Figure 3.4.: Random Forest Classifier

Random forest classifier is built in two phases: create N number of decision trees, and get predictions with those trees for test sets, and finally combine their predictions by majority voting (averaging). Random forest is a popular bagging model which minimizes a cost function for evaluating performance of trees. Extremely Randomized Trees (ExtraTree) is more randomized "bagging" of multiple trees, which decreases variance with cost of slight increase of bias. Among hyper-parameters of random forest, the first one is the number of estimators (default is 10) that controls the number of trees in the model. The larger the number of estimators, the more over-fitted the model. The second one is criterion which is a function (default is entropy) determining the accuracy of the split.

In [3], challenges of text classification including high dimensionality of text (i.e. as number of features increases, classifiers performance decreases) and some features might be irrelevant or redundant or noisy (i.e. these misguide classifiers). To reduce the level of these problems, different methods are used.

Dimensionality reduction methods include PCA, LDA, NMF, LSI, SVD, etc. [123, 23]. These feature transformation methods are used to reduce feature vector sizes from higher dimensions to lower dimensions, rather than selecting features from feature sets.

3.5.2. Feature Extraction

As discussed earlier, text data is represented using different text feature sets. For the task of sentiment classification, the most popular features include Bag-Of-Words (BOW), Term Frequency-Inverse Document Frequency (TF-IDF), ngram (character/word) language modeling feature sets, topic modeling feature sets and word embedding feature sets.

For feature extraction, pre-processing is applied on the text. Basic pre-processing includes punctuation mark removal, stop word removal, stemming, numeral removal and non-Ethiopic

character removal. More about preprocessing is discussed in chapter 5. Below are the proposed feature sets for sentiment classification of Facebook comments under resource-constraint settings.

BOW: the terms in a text document are grouped into ngram words in contiguous order to take account of the words, phrases or collection of words which appear in a sequence. The BOW ngram models include uni-gram, bi-grams, tri-grams and so on. Regardless of how frequently they appeared in a document, the presence or absence of BOW N-gram features are taken into account. For machine learning and natural language processing tasks, up to order of tri-gram words is sufficient. Studies revealed that bigram words feature performs better than unigram word feature for text classification [60]. Similar research reported that BOW word feature is showing better performance for distinguishing text categories compared with other types of text feature sets such as TF-IDF [60] and word embedding.

TF-IDF: the TF-IDF ngram word feature is computed by multiplying the term-frequency with inverse-document frequency of a word. Term-Frequency (TF) of a term is the ratio of number of times the term t appears in a document d divided by the total number of words in document d. The TF of a term is computed for a sample of Amharic corpora to identify rare words and highly frequent words (stop words) in the text collection. TF is denoted as $TF(t,d)$, which is represented in equation 3.11.

$$TF(t,d) = \frac{f_{t,d}}{\sum_{t' \epsilon d} f_{t',d}} \tag{3.11}$$

where $f_{t,d}$ if the raw count of term in document d and $\sum_{t' \epsilon d} f_{t',d}$ is the sum of the raw frequency of a term t' in document d, and the inverse document frequency of a term t is given by:

$$IDF(t,D) = log(\frac{|D|}{|d \epsilon D : t \epsilon D|}) \tag{3.12}$$

Where, |D| is the total number of documents and $|D_t|$ is the same as $|d \epsilon D : t \epsilon D|$ referring to the number of documents containing term t [127].

According to Manning et.al in [127], the term-frequency inverse document frequency ($TF - IDF(t,d)$) is given by multiplying $TF(t,d)$ and $IDF(t,D)$. i.e.

$$TF - IDF(t,d) = TF(t,d) \cdot IDF(t,D) \tag{3.13}$$

Term frequency is used to inform the degree of importance of the terms in a sample of corpora. Unlike term frequency, TF-IDF includes inverse document frequency of a term to penalize non-informative highly frequent words (i.e. stop words). In this case, TF is a normalized term frequency which is the local weight of a term in a document whereas IDF refers to a global weight of the term in the collection. When TF is multiplied by IDF, it used to get the weight

of a term in a document. The higher the TF-IDF of a term, the more discriminant the term is.
Ngram Character Feature: In computational linguistics, Ngram refers a contiguous sequence of n items from a given sample of text or speech. However, in this case, the N-gram character features of either bag of word (BoW) or TF-IDF are used as feature vectors.
Topic Modeling Feature: For Amharic sentiment classification, Latent Dirchlet allocation algorithm is applied to categorize Facebook comments into topics relying on the underlying probabilistic distribution of words in the collection. After repeated observations of the dataset, Facebook comments are categorized into 9 topics. Relying on this topic model, each training and testing Facebook comments are transformed into LDA feature vectors. This section presents the LDA used as a feature set for input to training machine learning for sentiment classification. The details of its performance on sentiment classification is depicted in Table 7.2.
Word Embedding Feature: The pre-trained Amharic Word Embedding in [52] is used to transform the training and testing sets into word embedding feature vectors.
Averaged Word Embeddings Feature: The trained Amharic word embeddings are used to calculate the averaged word vector representation of a document d with words w_1, w_2, w_3, w_n. It can be represented mathematically as:

$$AWV(d) = \frac{\sum_{i=1}^{n} wv(w_i)}{n} \tag{3.14}$$

where n is the number of words in document d.
TF-IDF Weighted Averaged Word Embedding Feature : The TF-IDF weighted averaged word vector for each document d containing words w_1, w_2, w_3, w_n. Mathematically,

$$TWA(d) = \frac{\sum_{i=1}^{n} wv(w_i) \cdot TF - IDF(w_i)}{n} \tag{3.15}$$

This research tries to investigate the features extracted from Facebook comments. The features which are prominent for detecting the sentiment category are selected.

3.5.3. Evaluation Metrics

To measure the performance of a classification system, evaluation metrics are required. The most popular metrics include accuracy, precision, recall and F-score. Before discussing each of these metrics, confusion matrix is defined. Confusion matrix is an N by N matrix used for performance of a classification algorithm, given that N is the number of classes. The purpose of the confusion matrix is to compare the actual target values with predicted values by the classifier model. This matrix reports the summary of what the model predicted correctly and what is not. Let's see in more detail by using a 2 by 2 confusion matrix which is a binary classification task. The 2x2 confusion matrix is represented in Table 3.5.3.

Table 3.1.: The confusion matrix for evaluating a binary classification system, where TP = True Positive, FP = False Positive, FN = False Negative and TN = False Negative

		Actual Values: Positive	Actual Values: Negative	Total
Predicted Values	Positive	TP	FP	TP + FP
	Negative	FN	TN	FN + TN
	Total	TP + FN	FP + TN	TP+FP+FN+TN

In the Table 3.5.3, the columns represent the actual values i.e. Positive and Negative whereas the rows represent the predicted values of the target class. The most important terms, TP, TN, FP and FN, are:

- (i) True Positive (TP) is the number of positive class members that are correctly predicted by the model.
- (ii) True Negative (TN) is the number of negative class members that are correctly predicted by the model.
- (iii) False Positive (FP) is the number of positive class members that are wrongly predicted by the model. This is also called Type I Error.
- (iv) False Negative (FN) is the number of negative class members that are wrongly predicted by the model. This is also called Type II Error.

Now, it is time to define the model's performance evaluation metrics which are presented below:

- (i) Accuracy is the percentage of correctly predicted samples. i.e.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.16}$$

- (ii) Precision (P) is the evaluation metric that measures the correctly predicted samples that actually turned out to be positive i.e.

$$Precision(P) = \frac{TP}{TP + FP} \tag{3.17}$$

  Note that precision is a metric that measures the reliability of the model.

- (iii) Recall (R) is a measure of the number of actual positive samples which are correctly predicted by the model i.e.

$$Recall(P) = \frac{TP}{TP + FN} \tag{3.18}$$

  Recall is also called sensitivity. Note that precision is more important to tell when the model predicted more false positive samples than false negative samples. On contrast, recall is more important metric to tell if the model predicts more false negative than false positive samples.

- (iv) F-score: is calculated from precision and recall i.e.

$$F - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3.19}$$

Practically, as the precision of the model increases its recall declines and vice-versa. Therefore, to address this trade-offs, F-score measures the balanced performance of the model i.e. it measures both trends in a single value.
Now coming to module 3 in Figure 3.1 of the workflow, it specifies the different machine learning algorithms which are proposed for Amharic sentiment classification. The possible text features such as BOW, TF-IDF at word and/or character level ngrams, word embeddings and text-based embeddings enrichment (average word2Vec and TfIDFWord2Vec) are generated. From these feature sets, the best features are selected. First, these features feed to the machine learning algorithms (i.e. Logistic Regression, Naive Bayes, Support Vector Machine, Random Forest), and models' hyper-parameters are tuned and finally, the best models are used to build the next module. This module includes feature selection, supervised classifier selection and hyper-parameter tuning, effects of varying training size on sentiment classification performance. Similarly, the best feature sets and the best base classifiers from the previous stage of this module are combined using ensemble based method.
In BERT based transfer learning, first, Amharic wordPiece BERT tokenizer and BERT model for Amharic is built using a large general collection of Amharic text collection. BERT is fine-tuned to perform Amharic sentiment classification using small sentiment labeled Amharic corpora. The corresponding details of these methods are presented in section 7.1.

3.6 Hybrid Approaches for Amharic Sentiment Classification

In module 5, a hybrid approach for Amharic sentiment classification is built. The detailed algorithms, results and discussion of this approach are presented. The hybrid approach combines the best classifiers built in earlier stages of the pipeline. Sentiment classification system is developed by using the combination of the top classifiers using selected salient features. The best performing classifiers are combined using different hybridization strategies such as voting

method, averaging and bending learners. This hybrid approach is illustrated in detail using UML diagram as shown in Figure 3.5.
The UML design is implemented using python libraries. The experimental results of the above-mentioned hybrid approach are discussed and reported in chapter 8.

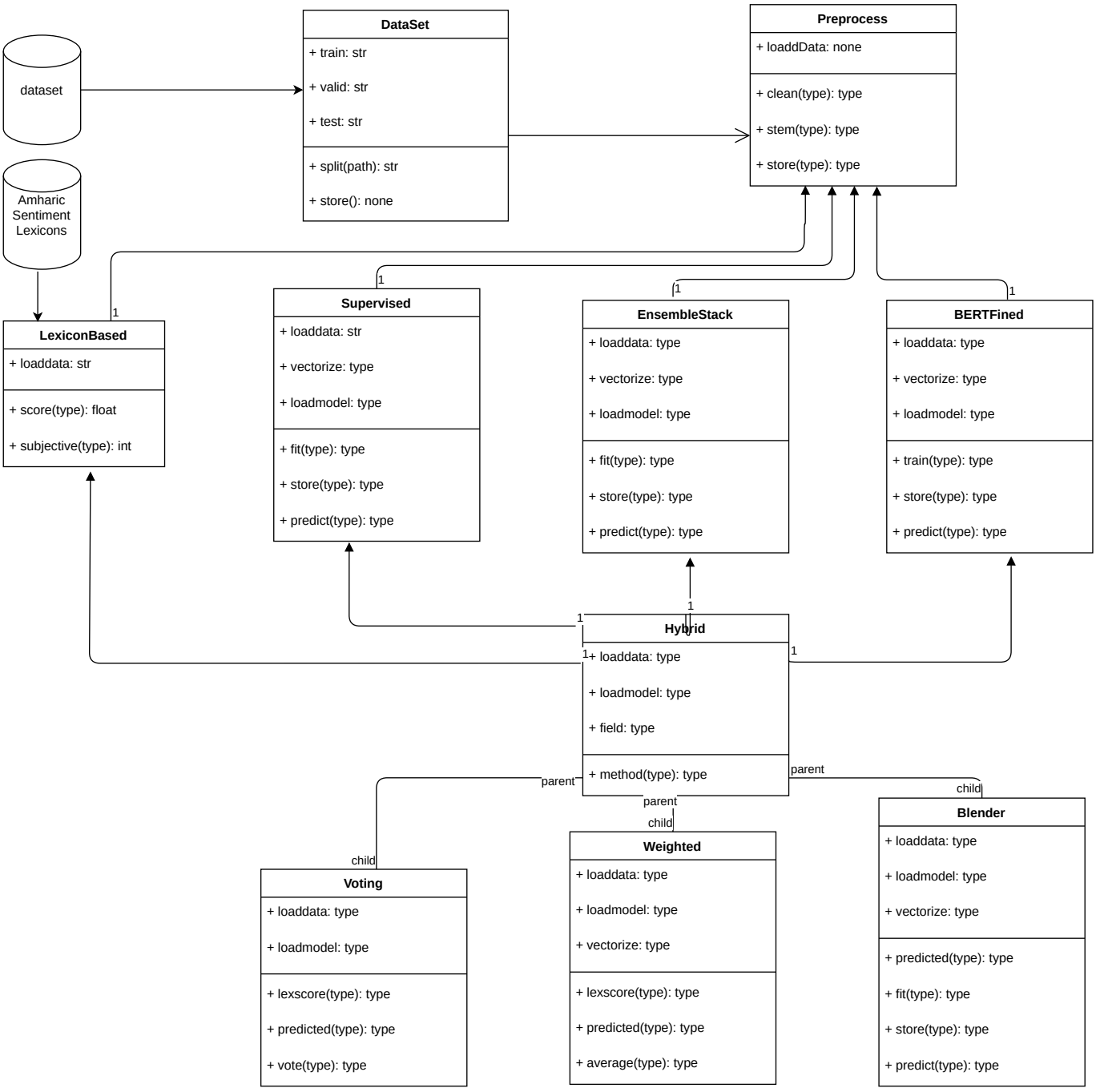


Figure 3.5.: UML Design for the Proposed Hybrid Sentiment Classifiers

Figure 3.5 shows the system design for hybrid approaches represented using UML diagram. The UML diagram contains a set of related classes including dataset class, preprocess class,

lexiconBased class, Supervised class, EnsembleStack class and BERTFined class, Hybrid class, Blender class, Voting class and Weighted class. A lexicon object from lexiconBased class creates lexicon object and preprocessing object is created from Preprocessing class. Similarly, supervised object, ensemble object and BERT object are created from the corresponding classes. The objects in turn are rendered by the hybrid object. The Hybrid class has three sub-classes such as Voting class, Weighted and Blender class. The corresponding attributes, methods and associations are also specified in the diagram. The high level architectural design and its logical flow and operations' description, evaluation and discussions of those three design alternatives of the hybrid approach are presented in chapter 8.

Each of the aforementioned modules in the research workflow are discussed in subsequent chapters of this thesis.

4. Data Preparation

This chapter describes the data preparation, description of the categories of the corpora, data annotation, lexical resources and pre-trained models, hardware and software tools needed at different stages to Amharic resource generation and Amharic Sentiment Classification.

4.1 Data Preparation, Description and Annotations

4.1.1. Dataset Collection

Before pre-processing is performed, a description of the list of corpora, lexical resources and pre-trained models is presented. The general corpora, labeled corpora, lexical resources and pre-trained models are grouped under four categories: category I, category II, category III and category IV, respectively. Each of these categories is further described as follows.

Category I: The datasets in this category are used for various tasks such as building Amharic sentiment lexicon using corpus-based approaches, evaluating the coverage of the generated Amharic lexicons, building or tuning Amharic word embeddings and BERT language model.

Table 4.1.: Category I Corpora Description

Dataset	#Articles	#Sents	Avg.Len	Size	Source	Year
Walta-I	18,467	19070	1184	75MB	Archive	2011-15
Walta-II	1,064	7619	1049	4MB	Geez.org	2001
REPORTER	25,673	14018	4294.79	255MB	Archive	2011-17
GCOA	4,013	4010	1090.38	37.5MB	Archive	2014-17

Category II: This category contains four sentiment labeled Amharic corpora. The labeled data set comprises the collections of annotated social media contents, in this case Facebook and YouTube comments. The reason behind choosing this domain is that: (i) social media user generated content usually contains subjectivity, expressing opinions towards or against an entity, place, event, issues. Facebook corpora were collected from local news media archives, from official Facebook pages of these government media centers and from Zemen Drama YouTube viewers' comments.

These datasets are collected from two social media domains namely, Facebook comments and YouTube movie comments (Zemen Drama[1]). The Facebook comments collected from: (i) Facebook page of GCOA[2], (ii) the official Facebook page of PMO[3] and (iii) the Facebook page of EBC[4]. Facebook comments (obtained from GCOA and EBC) are labeled by their own professionals into binary class (i.e. either positive and negative). The professional manually collected Facebook viewers' comments of highly viewed counts, i.e. ranges from 17,000 to 450,000 for these corpora when it was collected from their official Facebook pages. On the other hand, the PMO Facebook viewers' comments are collected from their official Facebook page of the year 2017 using Facebook scraper tool called Facepager[5]. Two annotators preprocess and label the PMO Facebook comments, which will be discussed shortly.
With the second domain, Zemen Drama is a weekly TV drama series which also posted in its official YouTube channel, having over 400,000 subscribers. The drama has 245 parts. By the time of collecting viewers' comments, all comments from part 1 to part 184 are collected. These comments are cleaned and filtered (i.e.1440 comments).
The statistical summary of the above four datasets is depicted in Table 4.2 and it is also available in Zenodo[6] data sharing repository.

Table 4.2.: Category II Social Media Dataset Statistical Description

Dataset	#Raw	#Processed	Avg.Len.	#words	Size	#POS	#NEG	Year
FB-GCOA	2871	2871	41.31	8.34	2871	1728	1143	2016-17
FB-EBC	3610	2446	99.75	20.41	2444	1707	737	2017
FB-PMO	10500	6637	192.03	39.03	6637	4589	2048	2018
YT-ZEMEN	2087	1440	63.21	13.42	1440	490	850	2016-18

4.1.2. Dataset Annotation of Unlabeled Data Under Category II

Zemen Drama Comments: The final 1440 comments have been annotated by two human annotators, who have been employed to label the comments relying on the annotation guideline in Appendix A. To compute the agreement between the two raters, Cohen Kappa formula shown in equation 4.1 is used. According to the authors in [92], Cohen Kappa formula is used for finding the inter-annotator agreement of two annotators. Cohen Kappa is a measure of inter-rater

[1]https://www.youtube.com/channel/UCzfrWFpc5sgVyybMHp5b5sQ
[2]https://www.facebook.com/gcao.ethiopia
[3]https://www.facebook.com/PMOEthiopia/
[4]https://www.facebook.com/EBCzena
[5]https://github.com/strohne/Facepager/
[6]

https : *//doi.org*/10.5281/*zenodo*.5005968

reliability (i.e. the agreement between only two raters on the same number of items). Cohen Kappa is given by:

$$\kappa = \frac{Po - Pe}{1 - Pe} \tag{4.1}$$

where, Po is observed agreements and Pe is expected agreement by chance. The κ value determines the agreement between two raters on the datasets. If the value of κ is 0, there is no agreement between raters. If the value of κ is 1, there exists complete agreement between annotators. According to the authors in [92], the following ranges of values are used to decide the agreement.

If κ is	less than 0.00,	the raters have **poor** agreement.
If κ is within	0.00 - 0.20,	the raters have **slight** agreement.
If κ is within	0.21 - 0.40,	the raters have **fair** agreement.
If κ is within	0.41 - 0.60,	the raters have **moderate** agreement.
If κ is within	0.61 - 0.80,	the raters have **substantial** agreement.
If κ is within	0.80 - 0.89,	the raters have a **strong** agreement.
If κ is within	0.81 - 1.00,	the raters have **almost perfect** agreement.

Authors in [92, 77] used different scales (or the κ agreement intervals). This difference is because the original creator of KAPPA tool did not mention in their report why and how the intervals of agreement are fixed.
In this study, Cohen Kappa is used, as two annotators are employed. The annotators have been oriented following the annotation guidelines in Appendix A. The raters' annotations of 1440 Zemen Drama YouTube comments are represented as confusion matrix shown in Table 4.1.2.

Table 4.3.: Annotation Summary of Two Annotators (A and B) of Zemen Youtube Comments

		Annotator B		
		Positive	Negative	Total
Annotator A	Positive	639	211	850
	Negative	148	442	590
	Total	787	653	1440

The observed agreement (Po) is computed as follows:

$$Po = \frac{(A+, B+) + (A-, B-)}{Total} = \frac{639 + 442}{1440} = 0.75 \tag{4.2}$$

The expected agreement Pe is computed as follows.
$Pe = P+ + P-$
Where P refers to the probability of being positive, $P+$ and probability of being negative, $P-$.

$$P+ = \frac{((A+, B+) + (A+, B-))}{Total}.\frac{((B+, A+) + (B+, A-))}{Total} = \frac{(639 + 211)}{1440}.\frac{(639 + 148)}{1440} = 0.32 \tag{4.3}$$

Similarly, probability of being negative, $P-$ is calculated as:

$$P- = \frac{(A-, B+) + (A-, B-)}{Total}.\frac{((B-, A+) + (B-, A-))}{Total} = \frac{(148 + 442)}{1440}.\frac{(211 + 442)}{1440} = 0.18 \tag{4.4}$$

The sum of $P+$ and $P-$ is computed, i.e.
$Pe = P+ + P- = 0.32 + 0.18 = 0.50$
Using equation 4.1, Kappa coefficient κ is given by:

$$\kappa = \frac{Po - Pe}{1 - Pe} = \frac{0.75 - 0.50}{1 - 0.50} = 0.49 \tag{4.5}$$

The inter-annotator rating agreement is 0.49. This shows that the agreement between annotators is moderate.

GCOA Facebook Datasets Preparation: The Facebook comments datasets collected by the Ethiopian Federal Government Communication Office Affairs (GCOA) are manually collected, filtered and finally labeled by professionals, in two classes, ignoring neutral comments. The EBC's professionals, while preparing Facebook comments, follow the same procedure.

PMO Facebook Datasets Preparation: The datasets on the PMO's Facebook page comments are collected, filtered and described in the Table 4.2. These comments are labeled by two professional annotators. In all cases similar annotation guidelines listed in Appendix A is used. According to Cohen Kappa formula, the agreement between the two annotators is computed. The annotators (A and B) rated 6637 Facebook comments.

The two annotators ratings arereported in the confusion matrix, depicted in Table 4.1.2.

Table 4.4.: Annotation Summary of Two Annotators of PMO Facebook Comments

		Annotator B		
		Positive	Negative	Total
Annotator A	Positive	1900	148	2048
	Negative	480	4109	4589
	Total	2380	4257	6637

The observed agreement (Po) is computed as follows:

$$Po = \frac{(A+, B+) + (A-, B-)}{Total} = \frac{1900 + 4109}{6637} = 0.90 \tag{4.6}$$

The expected agreement Pe is computed as:
$Pe = P+ + P-$
Where P refers to the probability of being positive, $P+$, and probability of being negative, $P-$.

$$P+ = \frac{((A+, B+) + (A+, B-))}{Total} . \frac{((B+, A+) + (B+, A-))}{Total} = \frac{(1900 + 148)}{6637} . \frac{(1900 + 480)}{6637} = 0.11 \tag{4.7}$$

Similarly, probability of being negative, $P-$ is calculated as:

$$P- = \frac{(A-, B+) + (A-, B-)}{Total} . \frac{((B-, A+) + (B-, A-))}{Total} = \frac{(480 + 4109)}{6637} . \frac{(148 + 4109)}{6637} = 0.44 \tag{4.8}$$

The sum of $P+$ and $P-$ is computed, i.e.
$Pe = P+ + P- = 0.11 + 0.44 = 0.55$
Using equation 4.1, Kappa coefficient κ is given by:

$$\kappa = \frac{Po - Pe}{1 - Pe} = \frac{0.90 - 0.55}{1 - 0.55} = 0.78 \tag{4.9}$$

The inter-annotator rating agreement is 0.78. This shows the agreement between annotators is substantial.

Category III: Under this category, two popular English sentiment lexicons i.e. SWN 3.0[7] which of size 72092 and SO-CAL[8] of size 10126 are selected for transferring sentiment labels to corresponding Amharic sentiment lexicons using Amharic-English dictionary. The Amharic-English dictionary used is compiled from dictionaries by Amsalu Aklilu with a size of 16230, by

[7]https://github.com/aesuli/SentiWordNet/tree/master/data
[8]https://github.com/sfu-discourse-lab/SO-CAL/tree/master/Resources/dictionaries/English

Selam Soft a size of 2,069, and by NSU[9] with a size of 12701 words. Manually labeled Amharic lexicons in [49] are used as baseline for evaluating the generated Amharic sentiment lexicons. The details of Amharic sentiment lexicon generation are discussed in chapter 6.

Category IV: This category includes pre-trained models like static word embedding (i.e. wiki.am.vec 300) for word embedding and TF-IDF weighted averaged word embedding feature sets, and other pre-trained models built and used in the subsequent sections and chapters (chapter 7.1). These include language detector, topic detector, conventional machine learning model, ensemble model, and BERT pre-trained model for Amharic Sentiment Classification.

4.1.3. Hardware and Software Tools

Python programming language of version 3.7 is installed with Spider 3.8 IDE running on Anaconda virtual environments. The most popular machine learning framework of python libraries is used. These include, pandas 1.2.4, numpy 1.20.3, scikit learn 0.24.2, Keras 2.4.3, Pytorch 1.0.2, and Tensorflow 2.5.0. The running operating system is Linux Mint 19.4, 64 bits installed on ProBook HP Core i7. For BERT pre-training Amharic Language model experiment, Jupyter notebook is used in google colab running on TPU of 25GB RAM and 99GB storage, which is allocated for google Gmail account.

[9]https://github.com/geezorg/data

5. Text Pre-processing

The pre-processing tasks are used for cleansing and standardization of Amharic texts, which helps the analytical system to extract quality features and get higher accuracy. Preprocessing is the fundamental step in achieving higher performance. If the pre-processing task is required for a particular experimental setup, the corresponding preprocessing library interface is imported. It also investigates the effects of pre-processing operation on performance of Amharic sentiment classification. The last section presents the summary.

5.1 Text Preprocessing

Text pre-processing is a technique to cleanse noise, eliminate non-textual symbols, remove punctuation marks and remove stop words from raw text to produce text features that are used for tackling the target NLP task. After pre-processing is carried out, NLP techniques apply to extract and select features to detect sentiment in a text. For feature extraction and selection, pre-processing pipeline is carried out on Amharic text. The pre-processing pipelines enrich linguistic that is used for structuring the unstructured raw texts. The flowchart showing the Amharic text pre-processing steps is depicted in Fig. 5.1.

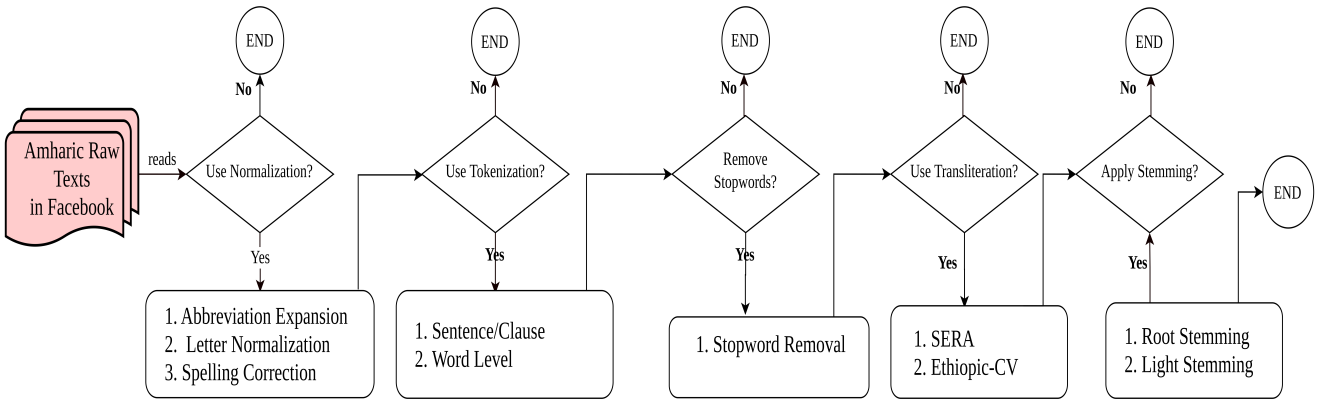


Figure 5.1.: Steps of Amharic Text Pre-procesing Operations

The pre-processing techniques on Amharic texts are presented below.

5.1.1. Language Identification

The starting of the pre-processing pipeline is the identification of language. That means, the pre-processing pipeline is assumed to begin by detecting a language of input texts. As the sentiment classification task is relying on user generated content from social media text, the first step is identifying the name of a language, whether it is Amharic.

5.1.2. Transliteration

To simplify further analysis of Amharic corpora, first Amharic text needs to transliterate into unified representation based on SERA (System for Ethiopic Representation in ASCII) [46]. Amharic has its own 34 base characters. Each of the Amharic characters (i.e. 34 consonant symbols) has seven orders (i.e. variations) depending on the combination of vocal sounds (i.e. vowels) such as a̱, u, i, a, e, ə and o as the complete alphabet shown in the Appendix B. The characters in the sixth column of the alphabet refer to the basic sounds of the language. The alphabet has 238 characters (34 base characters times 7 orders). Transliteration technique is developed to convert each character of Amharic texts into Ethiopic-consonant-vowel forms using Amharic characters. This can equivalently replace SERA (latinized consonant vowel transliteration). It easily done the encoding and decoding the texts without requiring transliteration and bringing back to Amharic texts using Latin alphabet. It is noted that the sixth column represents consonant characters. The ə sound is used either implicitly or explicitly, depending on the context of nature of the consonant letters in the text. For example, **መንግስት**/'mengst'/ to mean "government". In this word, the consonant n, s and t has no vowel ə following them. As a result, it is not mandatory for the sixth order letters to have a vowel ə explicitly [79].

5.1.3. Sentence/Clause Tokenization

This is splitting each document into a list of sentences. In English, full stop markers (.) followed by space is used. However, full stop marker is used for other (e.g. abbreviations of Doctor (Dr.)). In Amharic text, Amharic sentences are ended by pairs of colon (::) followed by space. This marker is used to split a document into sentences. If the text contains compound sentences, it is broken into simple sentences. The sentence terminal punctuation marks such as exclamation mark, question mark, four dots(::) are used to segment document into sentences.

5.1.4. Word Tokenization

Word tokenization is the technique of splitting or segmenting a sentence into constituent words. The words are used to reconstruct the sentence. Similar to texts in English, Amharic words are separated by space. However, in an old version of Amharic texts, Amharic punctuation marks

such as word separator (፡), semicolon (፤) and double colon (፥) are used. Word tokenization is done prior to NLP tasks including stemming, lemmatization, word level feature extraction, etc. Word tokenization is done before or after removal of special characters and punctuation marks depending on the task.

5.1.5. Abbreviation Expansion/or Normalization

Amharic Abbreviations or Acronyms differ from English in that Amharic text uses slash (/) between characters and sometimes dot (.) is also used. For example, Doctor and Addis Ababa are abbreviated as Dr.(ዶ/ር) and A.A.(አ/አ or አ.አ) respectively. Normalization of numeric values and abbreviated texts is also required. For example, if the text contains 10 and ten, replace 10 by ten and if there is T.V. and TV in the text, normalization is done by replacing T.V. by TV. For such abbreviations, dictionaries for abbreviations in the selected domain is prepared. For expanding abbreviated texts in Amharic, a long list of abbreviation dictionaries is developed where it contains expanded terms as key and list of abbreviations as values.

5.1.6. Punctuation Removals

Amharic has its own punctuation, which is depicted in Table 5.1. However, most of the punctuation marks are rarely used in modern writing systems. Sometimes English punctuation marks like dot(.) or backslash(/), question mark(?), exclamation mark(!), etc. are used in Amharic texts. The punctuation marks, including full stop (::), question marks (?) and exclamation marks (!) are used at the end of Amharic sentence or phrase.

Table 5.1.: Amharic Punctuation Marks

Punctuation Mark[1]	Purpose
※	section mark
፡	word separator
።	full stop (period)
፣	comma
፤	semicolon
፥	colon
፦	preface colon
፧	question mark
፨	paragraph separator

Abbreviation expansion is carried out before punctuation mark removal. In Amharic modern writing system, Ethiopic numerals **፩, ፪, ፫, ፬, ፭, ፮, ፸,፹,...** and the Ethiopic punctuation marks **፡,።,፣,፤,፥,፦,፨,፧** are rare to find in user generated Amharic texts in Facebook comments. The punctuation marks and the Amharic numerals are not considered for sentiment classification. In this research, it is removed from the text as part of pre-processing.

5.1.7. Letter Normalization

In Amharic language alphabet, there are some characters with the same sound in the Amharic alphabet. For the sake of speeding up the processing time of the NLP tasks in Amharic texts, normalization is done by replacing the various characters of same sound by a single character of its category. Similarly, Amharic characters with vocal variant sound such as **ቈ, ቖ, ቒ, ቋ, ቖ, ኈ, ጐ, ጐ, ሯ,ሻ, ፏ...** shown in Appendix B which are transformed to letters of the same sound as part of pre-processing. Table 5.2 represents the characters of Amharic which has same sound.

Table 5.2.: Amharic Characters Representing the Same Sound

Amharic character	Replaced with character	Pronouncation
ሀ,ሃ,ሐ,ሓ,ኀ,ኃ,ኻ	ሀ	ha
ሰ,ሠ	ሰ	Se
ፀ,ጸ	ፀ	Tse
ዐ,አ,ኣ,ዐ,ዓ	ዐ	a
ቆ,ቈ,ቐ	ቆ	Qo
ቁ,ቍ	ቁ	Qu
ኮ,ኰ	ኮ	ko
ጎ,ጐ	ጎ	go
ኋ,ኺ,ኼ	ኋ	hwa

Ethiopian script contains redundant symbols representing the same sound. Besides this, some of Amharic words have been written in various forms because of spelling variations used by different people. For example, the word ቴሌቪዥን ('television') can be written as ቴሌቭዢን, ቴሌቭዥን, ቴሌቪዥን [68]. In this work, the Amharic texts containing characters of the same sound is replaced by characters of the same sound. This normalization step is carried out before Amharic text transliteration.

5.1.8. Stemming

Stemming is reducing a word to its stem form by removing affixes. For example, አልመጣሁም(I did not come) and its stem is -መጣ-(come)) and its root is -ም-ጥ-. In the literature, there are works related to stemmers for Amharic language. For example, Argaw et. al [16] developed a rule based Amharic stemmer using dictionary look-up. Genet Fikremariam in [45] develops corpus based approach using successor variety algorithm with peak and plateau method to stem Amharic texts. The recent work in (Gasser, 2017) builds a morphological tool for three languages (Amharic, Oromo and Tigreygna) and can take Amharic surface words into its root form[48]. This work tries to provide open source morphological analysis tools for the aforementioned languages. This tool is taking Amharic word returns its root form, including linguistic features. However, it is computationally expensive to use it.
A light stemmer of Amharic language is developed which can preserve semantic information

by removing frequent prefixes and postfixes of the input word. The proposed approach is like the approach in [9] because it is affix removal. The authors in [9] developed the first Amharic stemmer relying on an iterative approach for hoping to improve performance of Amharic text retrieval system. This stemmer tries to handle prefixes and suffixes and it is evaluated with accuracy of 95% on 1221 words from specific of domains. However, this tool is not publicly accessible to test its performance for sentiment classification. This is the other reason the researcher inspired to develop light stemmer rather than root(heavy) stemmer. One of the aim of this research is to test the light stemmer for sentiment classification while comparing it with the root stemmer (benchmark) [48]. The light stemmer is relying on a set of rules using the regular expressions to remove prefixes and suffixes. During stemming, if Amharic word has a match to the pattern of regular expressions, the word is reduced to its corresponding stem by removing the matched suffix or/and prefix. For example, for the Amharic words such as **'ቆንጆው'**/the one who is handsome/masculine/, **'ቆንጆውን'**/someone who is handsome/masculine/, object form/, **'ቆንጆዎቹን'**/those who are beautiful, **'ቆንጆዎች'**/those who are beautiful, **'ለቆንጆዋ'**/to the beautiful one/feminine/ are reduced to the common stem by converting the symbols to corresponding consonant of Amharic/ **'ቅ-ን-ጅ'**/ by a proposed stemmer in [101].
See results in Table 6.4 that the accuracy of sentiment classification is improved relying on proposed lightweight stemmer.

5.1.9. Stop Words Removal

Stop words (also called common words) are the list of words which are the top frequent or redundant words in the term distribution of a language. Using Facebook comments corpora, category II above, the top 500 most frequent terms relying on the power law are plotted where it shows the relationship between two parameters (a term and its rank within the frequency distribution) in Figure 5.2. This distribution shows a long-tail, which refers to a small proportion of terms which are less dominant in distribution of occurrence in the collection. Zipf's Law, which states that for a sample of document collections, empirically expressed, the frequency of any word in the corpora is inversely proportional to its rank in the frequency table. Zipf's law has shown to identify the distributions of words in many languages [21].

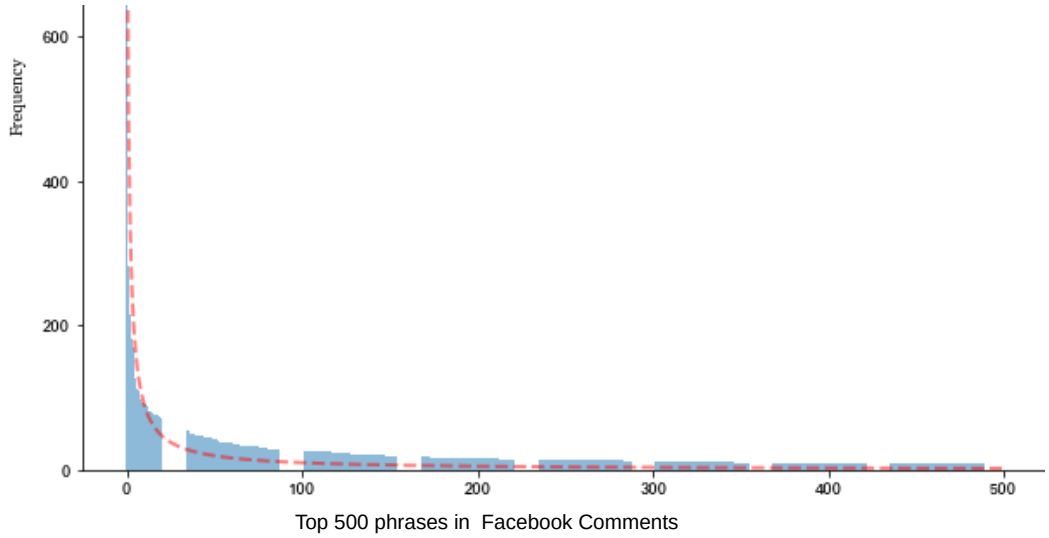


Figure 5.2.: Zipf's Law of Words Distribution for the Sample of Amharic FacebookComments

In this work, the top 20 percent of the terms are taken as the most frequent words which carry no information to discriminate documents in the collection. This is because these words appear in almost everywhere in the collections. These terms are taken as common words or stop words. Once the common words are identified, they are used to removing stop words while pre-processing the text. As some redundant words are opinion words or negation words, these words are not part of stop words. For example, the top 14 frequent terms in the positive comments. However, the terms **ጥሩ፣ ይገባል፣ በጣም፣ ደስ፣ መልካም፣ ሁሉም፣ ሰላም፣ ትኩረት፣ ለውጥ፣ ተጠናክሮ፣ እናመሰግናለን፣ በርቱ፣ እድገት፣ ተስፋ** whose meanings are /good, deserves, very, happy, good, all, peace, attention, change, strengthen, thank you, be strong, progress, hope/, respectively, are directly or indirectly affecting the sentiment of a comment. If these terms are taken as part of stop word list, it is difficult to detect the sentiment of the comments as these positive sentiments words are not present in the input comments. Even though they are highly frequently appearing in the corpora. These terms are not included as part of stop words. The same is true in the negative comments corpus. The terms such as **አይደለም፣ እንጂ፣ ምንም፣ ችግር፣ የለም፣ ሳይሆን፣ ውሸት፣ ችግሮች፣ የለበትም፣ በጣም** whose meanings are /no, but, nothing, problem, none, might not, lying, problems, nothing, very/, respectively, are also opinion or negation shifter terms which appeared in the top 50 most frequent terms in the negative comments corpora. However, some words such as **አይደለም**/it is not/, **ምንም**/nothing/, **የለም**/none/, **ሳይሆን**/not happened/, **የለበትም**/has nothing in it/ are negative words. Removing these words as stopword list affects negation handling for sentiment classification. These words are not included as part of stop word list. These words are not removed as part of stop word lists.

In this work, besides the 20 percent top most frequent terms taken from the corpora of category I, stop word list are extended by taking stop words which are already identified in the sample of general Amharic corpora in [52]. Depending on the NLP application, the list of stop words

requires revision of the terms in the list by checking whether the terms are important for the target application or not.

5.1.10. Spelling Correction

Spelling Correction is the method that tries to correct spelling errors in words of a language. The level of spelling errors is mostly too frequent in user generated texts from social media such as Facebook, YouTube, Instagram, and Twitter. Unless such errors are minimized, the performance of NLP tasks such as sentiment classification will be very low. In this research, to reduce the level of spelling errors of Amharic texts, both n-gram char embedding and edit distance are proposed to correct errors in Amharic Facebook comments.

5.1.11. Negation Handling

In sentiment analysis, negation words or clues are inverting the sentiment polarity of a text under consideration. This research uses negation handling approach implemented in [10].

5.1.12. Subjectivity Detection

Subjectivity detection is the technique that identifies whether a text contains either subjective or factual (or objective) information [82]. This is performed before processing sentiment classification. Because, sentiment classification is carried out if the text contains subjective information. Texts with subjective information contain either positive or negative sentiment. In this research, subjectivity in Amharic text is detected relying on the Amharic sentiment lexicons generated (chapter 6). The algorithm for subjectivity detection is specified in Algorithm 5.1.

Algorithm 5.1: Amharic Subjectivity Detection Algorithm

Input: *InputTokenized*: tokenized list of text
Output: *Subjectivity*: returns subjective if score either +/- otherwise objective
`/* Load from files */`
Data: *AmharicSentiLexicon* ←load sentiment lexicons generated (from chapter 6)
Data: *Negation* ←load negation handling library [10]
`/* Initialize Variables */`
1 *NegationCount* ←0
2 *SentimentScore*←0
3 *IsSentimentWord*←0
`/* compute sentiment score for a text or word using lexicons */`
4 **foreach** *word in InputTokenized* **do**
5 | *SentiemntScore* + ←*AmharicSentiLexicon*[word]
6 | if *SentiemntScore*! = 0 *IsSentimentWord* + ←1
7 | *NegationCount*+ ←*negation*(word)
8 **if** *SentiemntScore !=0 and NegationCount% 2==1* **then**
9 | *Subjectivity* ←"SUBJECTIVE"
10 **else if** *SentiemntScore ==0 and IsSentimentWord>0 and NegationCount% 2==1* **then**
11 | *Subjectivity* ←"SUBJECTIVE"
12 **else if** *SentiemntScore !=0 and NegationCount% 2==0* **then**
13 | *Subjectivity* ←"SUBJECTIVE"
14 **else**
15 | *Subjectivity* ←"OBJECTIVE"
16 return *Subjectivity*

Description: Algorithm 5.1 is described as follows. Before start of processing the algorithm, the generated Amharic sentiment lexicons from chapter 6 and negation detection library interface from [10] are loaded. Lines (1-3) initialize the variables to store the count of negation in the text i.e. *NegationCount*, the sum of the sentiment score of the text i.e. *SentimentScore* and counter *IsSentimentWord* increments by one if a word is present in the lexicon. From lines (4-7), the algorithm iterates for computing sentiment score for each token. Specifically, the sentiment score is updated (line 5) and if the word is in the lexicon, *IsSentimentWord* is incremented by one (line 6). Negation count is updated for each token (line 7). If the sentiment score of the text not zero and the negation count by modulo is 1 (line 8), the text is subjective (line 9). If the sentiment score is zero and *IsSentimentWord* is greater than zero and the negation count by modulo is one (line 10), the text is subjective (line 11). The sentiment score is zero, in that the positive and negative sentiment might cancel out. If sentiment score is not zero and negation count by modulo is zero (line 12), the text is subjective (line 13), otherwise the text is objective (lines 14-15). The output of the algorithm is the subjectivity of the input text is returned (line 16).

In this research, subjectivity detection is performed before performing sentiment classification. Texts without sarcasm, irony, or proverbs, neutral/objective texts are weeded out. Particularly in real applications, if the raw text is not subjective, no more sentiment classification is carried out. In order to detect subjectivity of a text whether an input text expresses facts (i.e. objective) or opinion (i.e. subjective) of the text [82], the generated Amharic sentiment lexicons (in chapter 6) and negation handling interface in [10] are used.

5.1.13. Topic Detection

Topic detection is the detection of topics for a domain or text. For proper topic prediction, merging of the title with its comments is used to predict the topic category of the input text. In this study, the topic detection approach in [102] is used.

5.2 Summary

To get the salient features from input social media texts for sentiment classification, basic preprocessing such as identification of languages, normalization, spelling correction, transliteration, tokenization, abbreviation expansion, punctuation and numerals removal, stop word removal, subjectivity detection, negation handling are essential machine learning algorithms to best learn from the input features for the sentiment classification task.

6. Lexicon Generation for Amharic Sentiment Classification

A sentiment lexicon contains a list of opinion words where each term has been assigned positive and negative values, and sometimes, part of speech and glosses are included. This chapter is organized into four sections. The first section provides the background of lexicon-based sentiment classification. The second section presents dictionary-based approaches to Amharic sentiment lexicon generation. This section has subsections which provide proposed approaches, results and discussions, and summary about the dictionary-based approach. The third section presents corpus-based approaches to Amharic sentiment lexicon generation. This section also has subsections which provide details of corpus-based approach, proposed methods, results and discussion, and summary. The fourth section reports the human evaluation of the generated Amharic sentiment lexicons.

6.1 Overview

Lexical resources are generated for Amharic sentiment analysis research by exploiting existing monolingual and cross lingual resources with minimal cost and time. In the literature [28, 136, 81],lexicon-based methods are alternatively referred to with terms such as unsupervised, semantic-based, knowledge-based or rule-based. Lexicon-based approach is a technique for calculating sentiment of the semantic orientation of texts using sentiment lexical resources or a set of linguistic patterns got from linguistic corpora [28]. This unsupervised approach relies mainly on the presence of sentiment lexicons. This technique is used in difficult situations when there are no sufficient labeled data sets in the language. The major drawback of this approach is poor recognition of polarity when terms are out of the dictionary or polarity recognition fails if the polarity is associated with a particular linguistic context or if it is used out-of-domain.
There are several ways to build sentiment lexicon resources for lexicon based sentiment analysis. The most common approaches are manual, semi-automatic(dictionary-based) and corpus-based [94, 1, 11, 2].
In manual approach, the polarity values (positive, negative and/or neutral) of each lexicon in the dictionary are assigned manually. The drawback of this approach is that it is labor intensive and time-consuming. For example, MPQA (Multi Perspective Question and Answering) is

comprehensive manually built English subjectivity lexicon [93] and General Inquirer is also manually labeled English polarity lexicon.
Gebremeskel [49] manually built Amharic polarity lexicon (total number of entries is 1060). They labeled the polarity value of entries of this Amharic sentiment lexicon either negative (-2) or positive (+2), which does not show intermediate sentiment strength. This Amharic sentiment lexicon [49] has several shortcomings. For example, the lexicon is too small to cover all subjectivity terms in Amharic language [1]. This lexicon might have low precision not only in a general sentiment detection application but also in domain specific sentiment detection.
Dictionary-based sentiment lexicon is based on bootstrapping approach, where first carefully selected and annotated seed words are provided to a particular bootstrapping algorithm (weak learner) and then the learner extends the lexicon automatically. A dictionary-based approach starts with small seed words that are high polarity terms (i.e. usually adjectives and adverbs) and are selected manually. These seed lists are automatically propagated to online resources (e.g. SentiWordNet, WordNet [93, 34]), and the similarity of a new term with a term in the seed list is computed. If the new term is so close to the term in seed list, the new term is added to the sentiment lexicon. This process is iterated until a specific stopping condition is satisfied [93].
Corpus-based approach is like dictionary-based approach because it uses seed list terms, but unlike dictionary-based approach, corpus-based approach uses large corpora from which opinion words of similar co-occurrence patterns with the seed list terms are searched based on specific strategies used in the algorithm. For instance, if the new terms frequently co-occur with the same context to seed list, they probably have the same semantic similarity. The opinion terms' similarity values greater or less than some threshold settings are added to the sentiment lexicon.

The approaches to sentiment lexicon generation are shown in Fig. 6.1.

[1] http://sentiment.christopherpotts.net/lexicons.html

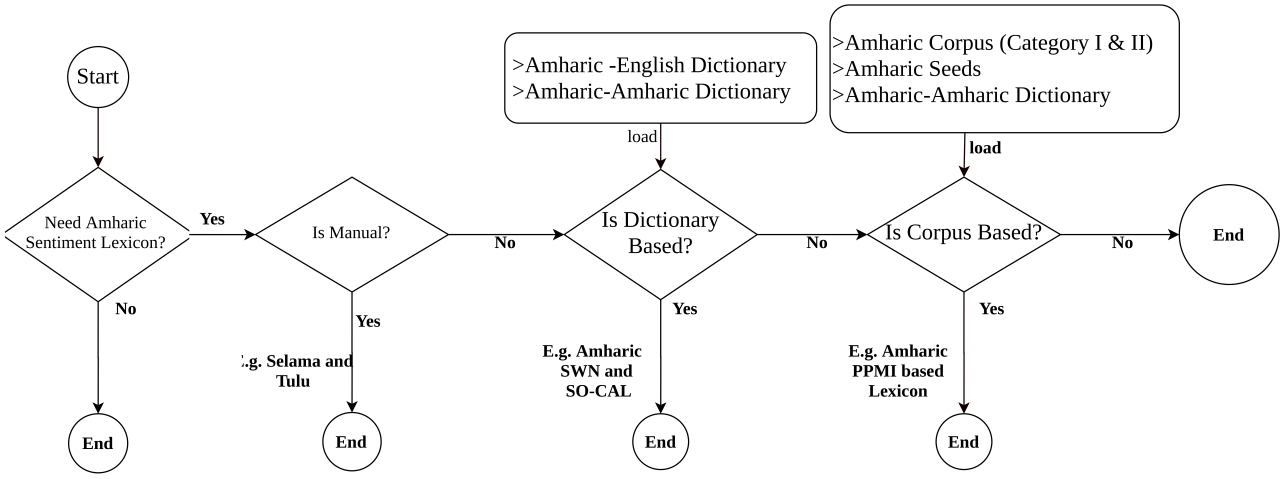


Figure 6.1.: The Approaches to Amharic Sentiment Lexicon Generation

Corpus-based approach is recommended to handle language and culture specific words in the sentiment lexicons [11]. However, it is impossible to handle all the words in the language as the corpora is a limited resource in almost all under resourced languages like Amharic. But still it is possible to build sentiment lexicons in particular in domains where a large number of Amharic corpora are available. Due to this reason, corpus-based sentiment lexicon is usually used for lexicon-based sentiment analysis in the same domain from which it is built.

6.2 Dictionary-based Lexicon Generation

6.2.1. Methods

Data Sets and Lexical Resources: This subsection describes the main data sets and lexical resources used for building and evaluating Amharic sentiment lexicons. (a) English SentiWordNet 3.0: SentiWordNet is automatically built based on synsets of wordnet version 3.0 [19]. The sentiment lexicon contains 72,092 terms with part of speech, id number, PosScore, NegScore, SynsetTerms and Gloss. The pair (POS, ID) uniquely identifies a WordNet (3.0) synset. The values PosScore and NegScore are the positivity and negativity scores (in the range of 0 to1) assigned to each entry of the synset. The objectivity score is computed as: ObjScore = 1 - (PosScore + NegScore). Synset-Terms column reports the terms, with sense number, belonging to the synset (separated by spaces). The sentiment lexicon of source language is ported to the sentiment lexicon of Amharic language

(b) The English SO-CAL Polarity Lexicon: SO-CAL refers to the Semantic Orientation CALculator, a tool to extract sentiment from text. It has a long history of development. The manually built sentiment lexicon [49] is used as a baseline. SO-CAL contains 10126 words with polarity value ranges from -5 to +5. The sentiment lexicon is categorized into adjectives,

adverbs, nouns, verbs and intensifier word lists. The sentiment lexicon has been extensively tested showing good performance in different domains.
(c) Amharic-English Dictionary: This dictionary works in one direction (i.e. Amharic to English). For each Amharic word, it contains part-of-speech tag(s) and corresponding meanings in English. The English text will have either a single word, phrase, or list of synonyms. The total size of the dictionary is 31000 terms which are obtained by merging Amharic-English dictionary (12700 Amharic words) [2], Aklilu's Amharic-English Dictionary (16231 words) and Amharic-English dictionary by SelamSoft Private Company (2,069 words). The final dictionary serves as a bridge to propagate sentiment from English sentiment lexicon to Amharic sentiment lexicon.
(d) Amharic-Amharic Dictionary: From more than 30 thousand entries of Amharic lexical word list in [3], Amharic-Amharic dictionary of size 33965 is automatically built, where this dictionary serves as a gloss source for terms in the generated Amharic sentiment lexicon.
(e) Facebook Comments Dataset: The data set consists of 2871 sentence/phrase level sentiment annotated Facebook users' comments collected from the Government Communication Office Affairs (GCOA) between 2016 and 2018. News that has highest number of viewers' comments were selected as "hot topics" and these Facebook comments were labeled by professionals into either positive or negative sentiment.
(f) Amharic Web Corpora: This is a crawled collection of web documents which consist mainly of news, politics and religion documents. It was collected and tokenized, automatically part-of-speech tagged and published under the HaBiT project [52]. The file is a single xml file of size 421MB. It contains 20 million tokens. These corpora are used for evaluation of the sentiment annotations.

6.2.2. Automatic Sentiment Score Calculation

To build an Amharic sentiment lexicon relying on bilingual dictionaries, subjectivity or polarity information is transferred from polarity lexicon of resource rich language (i.e. English sentiment lexicon) to resource limited language (i.e. Amharic). Specific to this approach, an algorithm is developed to transfer labels from English SentiWordNet 3.0 and SO-CAL to Amharic sentiment lexicon. As shown in Figure 6.2 below, the Amharic-English bilingual dictionary is employed as a bridge between the two languages. The proposed approach of building Amharic sentiment lexicon is similar to what is done for the Sinhala language in [93]. However, one of the differences in the proposed algorithm of this research is that it searches for each Amharic word as key rather than English words in the Amharic-English bilingual dictionary. That is why Amharic-English dictionary is used instead of English-Amharic dictionary. The

[2]https://github.com/geezorg/data
[3]https://github.com/geezorg/data

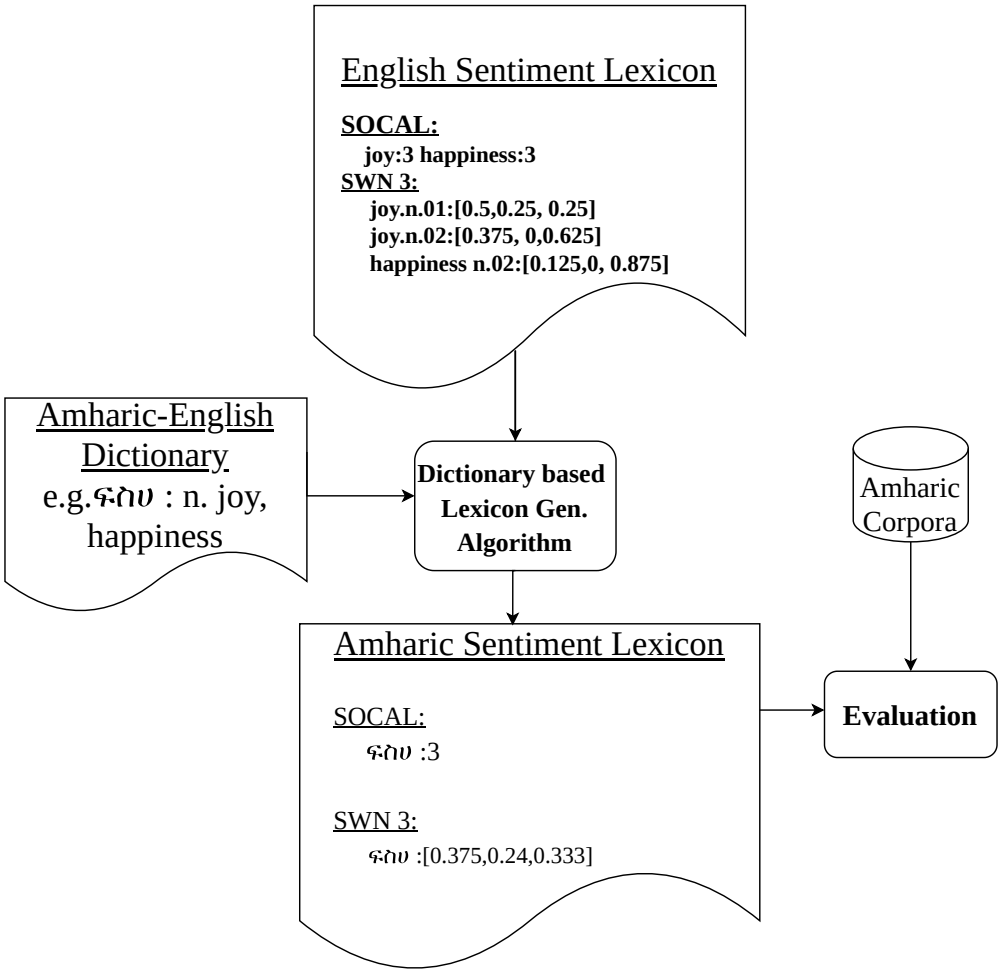


Figure 6.2.: Dictionary-based Amharic Sentiment Lexicon Building Process

key contribution of this subsection of the research is as follows: (i) the proposed algorithm uses average weighting of synonyms of English to tokens to assign the sentiment score for the corresponding Amharic term in the bilingual Amharic-English dictionary, and (ii) the approach in this work can also handle negation words that appear in the Amharic text. These findings in the proposed work make it different from the work of [93]. Consider Amharic word w_{ai} in the Amharic-English Dictionary d_{ae}, which has corresponding meanings in English word(s), w_{eij}. The Amharic sentiment lexicon is denoted by s_a, the English language sentiment lexicon by s_e. As part of preprocessing, all Amharic words are normalized in the Amharic-English dictionary by replacing varied alphabets of the same sound with identical symbols. Moreover, a stemmer is applied during evaluations. The sentiment lexicon building algorithm from SO-CAL is presented in listing 6.1.

Algorithm 6.1: Build Amharic Sentiment Lexicon

Input: s_e: English SWN or $SO-CAL$
Output: s_a: Generated Amharic SWN or $SO-CAL$

```
d_ae ← loadAmharic − EnglishDictionary
s_e ←load English sentiment lexicons: either SWN or SO − CAL
foreach w_ai ∈ d_ae: do
    Search its meaning words: ← w_ei1, w_ei2,. w_eij., w_ein−1, w_ein
    foreach synonym word w_eij ∈s_e, search its score in s_e do
        if found then
            return all the polarity values (+,-,objective) of the synsets
        Keep them in a list L
    Find length of +,-, objective list in L
    Compute the average weighted polarity in L using equation (6.1)
    Assign the average weighted polarity, part of speech, and gloss to w_ai in s_e
    That is, s_a[w_ai] = [Average weighted polarity, part of speech, gloss]
Return s_a
```

Algorithm Description: In listing-6.1, for each Amharic word in the bilingual dictionary, the aggregated sentiment values of corresponding English term(s) in Senti-WordNet 3.0 (or SO-CAL) are assigned to the Amharic word if terms are found in the source sentiment lexicon. Finally, the Amharic sentiment lexicon contains each Amharic term, its average weighted polarity values, its corresponding part of speech and gloss definitions [35].

For each Amharic word w_{ai} in the Amharic-English dictionary, there is a list of English words(w_{ei1}, w_{ei2},. w_{eij}., w_{ein-1}, w_{ein}) in the dictionary. The assumption is that the first English word w_{ei1} has more closer in meaning to Amharic word w_{ai} than the second English word w_{ei2} and in turn w_{ei2} is closer to w_{ai} than w_{ei3} and so on. As a result, more weight is given to the first word w_{ei1} and least weight is given to the last word w_{ein}. On the basis of this assumption, the weighted average sentiment of English words is computed and then the resulting sentiment value is assigned to the corresponding Amharic word w_{ai}. The sentiment score (w_{ai}) is given by:

$$w_{ai} = \frac{\Sigma_{j=1}^{n} score_{eij}(w_{eij}) \cdot (n - j + 1)}{n} \tag{6.1}$$

where n is the number of English words w_{ei1}, w_{ei2},.., w_{ein-1}, w_{ein}, and $score_{eij}$ are in turn refers to:

$$score_{eij} = \begin{cases} sentimentScore(w_{eij}) \;\; if\, the\, lexicon\, is\, SO-CAL \\ sumofSentimentScore(synset(w_{eij})) \;\; if\, the\, lexicon\, is\, SWN \end{cases}$$

Most Amharic sentiment words in the generated lexicon obtain their contextual meaning from another Amharic–Amharic dictionary. However, some terms are without context definitions as they were not found in the Amharic–Amharic dictionary [4].

[4]https://github.com/geezorg/data

The Amharic sentiment lexicons are generated and used under a few assumptions. (1) The senses of words in the bilingual dictionaries of the two languages are considered to be same. The sentiment score of an English word in sentiment analysis of English text is assumed to be identical to the sentiment score of an Amharic word in Amharic texts. (2) The parts of speech in both languages in the bilingual dictionary are assumed to be equivalent [93].
From English SentiWordNet 3.0 and SO-CAL, two corresponding Amharic sentiment lexicons are automatically generated. The sizes of these newly built Amharic sentiment lexicons are 13677 term entries and 5681 terms, respectively. The experimental results and evaluations are discussed in the subsequent section.

6.2.3. Results and Discussion

In this section, the experimental results relying on the generated Amharic sentiment lexicons are discussed. Each of the generated Amharic sentiment lexicons is evaluated in two ways using external data as well as intrinsic evaluation and mutual comparison of agreement. Finally the usefulness of the Amharic sentiment lexicons is tested in lexicon based sentiment classification of Amharic comments. To increase the string matching accuracy in generating the evaluation scores, basic preprocessing: tokenization, stopword removal, punctuation mark removal and stemming are applied, which are discussed in the preprocessing section.
External to Lexicon Evaluation: In this research, external evaluation is carried out in two ways. (a) subjectivity detection: Each term in the sentiment lexicons is counted only if it is present in a given Amharic comment. If more terms of the sentiment lexicon appear in the Amharic comment, the comment is subjective, otherwise objective. The usefulness of automatically generated sentiment lexicons are evaluated by their accuracy of detecting subjectivity of Amharic texts. These sentiment lexicons are used to distinguish subjective (positive or negative) Amharic Facebook comments from objective (neutral) Amharic comments. The accuracy of subjectivity detection increases as stemming is applied along with the sentiment lexicons. As Amharic language is morphologically rich, the variations of words derivations are normalized to the same stem. Usually, the resulting stem can be in root form. This improves computational efficiency by decreasing morphological variation of Amharic texts during its text processing by decreasing text comparison mismatch as many surface words are reduced to a common stem. The effect of stemming on performance of the lexicon-based sentiment classification is depicted in Table 6.1.

Discussion: Table 6.1 shows that the generated Amharic sentiment SentiWordNet lexicon outperforms the other lexicons, resulting in 99.33% correct subjectivity detection. The rate of detecting the subjectivity of Amharic Facebook comments using Amharic sentiment SO-CAL, in turn, outperforms the manual Amharic sentiment lexicon (93.56%). This subjectivity detection result shows that each of the Amharic sentiment lexicons outperforms the subjectivity

Table 6.1.: Lexicons' Subjectivity Detection Rate of Facebook Comments

Sentiment Lexicons	Detection withNoStem	Detection withStem
Amharic Manual(Baseline)	31.99	93.56
Amharic SO-CAL	31.28	96.65
Amharic SWN	75.83	99.33

detection result of the baseline sentiment lexicon (the manually generated Amharic sentiment lexicon) by 5.77%. One of the reasons for this might be that SWN is larger than the manual sentiment. The degree to which the entries of SWN appear in the Amharic Facebook comments is higher than the other sentiment lexicons.

Evaluation of Lexicons by its Coverage Count (or in Percent): This way of evaluating the generated sentiment lexicons is to measure the size of the sentiment lexicons in terms of their coverage count in general corpora containing 20 millions tokens and 2871 Amharic Facebook comments.

Table 6.2.: Lexicons' coverage (positive/negative count and in percent) on 2871 Amharic Facebook comments and 20 million tokens of Amharic web corpora

Lexicons	2871 Amharic Comments		20 millions Amharic tokens	
	coverage(+,-) count	in%	coverage(+,-) count	in %
Manual	[4399, 2995]	31.95	[2713167, 2161501]	25.01
SO-CAL	[5738, 3953]	41.87	[4169817, 3391213]	38.88
SWN	[9447, 4803]	61.57	[6645592, 4006072]	54.77

Discussion: The results in Table 6.2 verify that the coverage of the sentiment lexicon based on SWN is larger than the other sentiment lexicons on both Amharic corpora. The other aspect that this analysis verifies that the number of positive and negative opinion words are balanced in sentiment lexicons generated from SWN and SO-CAL. However, with SOCAL, the positive and negative count coverage on Facebook comments is not balanced. In addition, the coverage of the Amharic sentiment lexicons in both Amharic corpora is below the average benchmark semantic lexical coverage in other languages Welsh (25%) to Arabic (88%) [111].

Internal to Lexicon Evaluation: Internal evaluation of the lexicon is the process of counting common positive and negative opinion terms in the two sentiment lexicons generated from SWN and SO-CAL. The number of common terms in the two sentiment lexicons shows the extent of

the agreement (overlap) between the generated sentiment lexicons and the manual sentiment lexicon (baseline). To evaluate the results of the proposed approach, the agreement between the generated sentiment lexicons and the baseline lexicon are computed.

Table 6.3.: The Agreement Between Lexicons

-	Amharic Sentiment Lexicons	Agreement (%)
1	SO-CAL and Manual Lexicons	70.80
2	SWN and Manual Lexicons	59.49
3	SO-CAL and SWN	66.40

Discussion: Table 6.3 depicts that the Amharic sentiment lexicon from SO-CAL generated through dictionary approach has agreement or overlap of 70.80% with manual Amharic sentiment lexicon and it overlaps 66.40% with Amharic sentiment generated from SentiWordNet. On the other hand, Amharic sentiment from SWN and the manual Amharic sentiment lexicon have 59.49% of the total terms found in both lexicons. Although the size of Amharic sentiment from SWN is much larger (more than double) than Amharic sentiment from SO-CAL, the latter is more consistent in its sentiment scores with the other sentiment lexicon from SWN.
The agreement level of English sentiment lexicons are compared in symposium[5].In this comparison, the agreement level of SWN is better with Harvard General Inquirer (77%) than other English sentiment lexicons (MPQA, Opinion Lexicon, LIWC). Amharic generated sentiment lexicon is below the agreement levels reported for the English language resources. The reason for this is that it is very difficult to compare Amharic sentiment lexicons with sentiment lexicon of other language as the languages are very different in morphology apart from difference in cultural connotations.
Lexicon-based Sentiment Classification: Besides the evaluations in the earlier subsections, the usefulness of the generated lexicons and their combinations for sentiment classification using Amharic Facebook comments dataset are evaluated. Prior to sentiment aggregations of Amharic texts, basic text preprocessing (tokenization, punctuation mark removal, normalizing Amharic script symbols, stopword removal, spelling corrector, stemming, etc.) are applied. The use of stemming and negation detection technique on Amharic text is shown to improve the accuracy of lexicon-based Amharic sentiment classification.

[5]http://sentiment.christopherpotts.net/lexicons.html

Table 6.4.: The Accuracy (in percent) of Sentiment Lexicons for Sentiment Classification

Lexicons	Accuracy(%)		
	NoStem+NoNeg.	Stem+NoNeg.	Stem+Neg.
Manual(baseline)	16.7	42.9	52.7
SO-CAL	14.6	46.3	50.8
SWN	30.9	50.1	54.7
Manual +SO-CAL	37.2	63.4	72
Manual +SWN	49.9	65.9	74.62
SO-CAL +SWN	43.7	66.6	73.5
Manual+SO-CAL +SWN	53.7	75.8	86.2

Discussion: The usefulness of the sentiment lexicons is evaluated in terms of their accuracy of classifying sentiment of Amharic Facebook comments as shown in Table 6.4. In general, the results in Table 6.4 also reveal that stemming and negation handling on Amharic texts achieve performance gains of sentiment classification. The automatically generated sentiment lexicon from English SWN outperforms the performance of the manual sentiment lexicon (baseline) for classifying sentiment of Amharic texts. On the other hand, the manual sentiment lexicon (with stemming and negation handling) in turn outperforms the automatically generated sentiment lexicon from English SO-CAL. However, by applying stemming and without negation handling, Amharic sentiment lexicon from English SO-CAL relatively performs better than the Amharic manual sentiment lexicon in classifying sentiment of Amharic Facebook comments. The combination of generated sentiment lexicon (SWN) with manual sentiment lexicon (baseline) outperforms (with accuracy of 74.6%) the other combinations of sentiment lexicons for sentiment classification of Amharic texts. Similarly, the generated sentiment lexicons (SO-CAL + SWN) performed well (with accuracy of 73.5%). Yet another combination of the three sentiment lexicons (SO-CAL + SWN + Manual) performs the best (with accuracy of 86.2%) of all the other individual sentiment lexicons or their combinations.

Error Analysis: Unfortunately, it is challenging to trace the sources of errors in the automatically generated sentiment lexicons. Some of the causes of generated errors in the generated Amharic sentiment lexicons are pointed out. The identified errors are presented in this subsection:

(a)On-spot Analysis of Errors Detected in Sentiment Lexicons: In the generated sentiment

lexicons, the errors along with sentiment labels are transferred from source language terms to target language terms due to the following issues: (i) Mistranslation of Amharic terms are identified in SWN by the bilingual dictionary. For example, in Amharic SWN, the word **ሸጋ**/'nice'/ is wrongly assigned with negative sentiment. The reason might be that there is error in the bilingual dictionary or in the algorithm that propagates sentiment values from English, (ii) Correctly translated terms are obtained where they are assigned opposite polarity to source terms. For example, in Amharic SWN, the word **ቆራጥ**/'courageous decisive in manner'/. The translation is correct, but wrong sentiment value is assigned. (iii) A few terms are identified that they are correctly translated, with the same sentiment polarity but different sentiment strength. For example, **ሰይጣን**/ "devil"/ in Amharic sentiment lexicon from SWN. It is correctly translated and assigned a negative sentiment value but the sentiment strength is small. (iv) some terms are detected in SWN where these terms are correctly translated but different sentiment is assigned due to cultural connotations. For example, the term **እርካሽ**/"cheap"/ is assigned positive sentiment. In English, this might be correct, but in Amharic the word **እርካሽ** has negative connotation in that it refers to something which is sinful and lower in quality. This can also be positive depending on context. This can be addressed by incorporating context-dependent lexicon generated from Amharic corpora which leads to another venue of future research.

(b) Analysis of incorrectly Detected Amharic Comments: The associated reasons why the sentiment/subjectivity of Amharic comments are wrongly detected are identified. The subsets of reasons which cause wrong subjectivity/sentiment detection include: (i) Some Amharic comments are identified where these terms contain sarcasm and idioms. The nature of these comments is difficult to detect relying on ordinary sentiment lexicon. For example, **"የራሷ አሮባት የሰው ታማስላለች"**/solve your own problem before talking about others/ is an Amharic proverb that cannot be translated directly relying on ordinary dictionary. That is why it is wrongly classified by the sentiment lexicons. (ii) Besides handling negation, further formulations of linguistic rules (e.g. intensifiers, contrast rules, conjunction rules) are required to handle wrongly detected Amharic Facebook comments. For example, **"ጋይሌ ገ/ስላሴ በሩጫ ጎበዝ እንጂ ስለኢትዮጵያዊ መንፈስ ሊሰበክ አይችልም"**/H/G/Silasie is the best runner, however, he cannot preach about Ethiopianism/. In this example, the sentiment computation fails as the semantic orientation of the text is diverted by contrasting word **እንጂ**/however/. The text next to this contrasting word has a more dominant sentiment than the phrase before it. (iii) Some Amharic Facebook comments are identified where these comments are wrongly annotated by human annotators. For example, **"በጣም አሳዛኝ ነው"**/it is very tragic/. The sentiment of this comment is labeled wrongly as positive by the human annotator. The labeled data should be reviewed for correction. (iv) Some Amharic Facebook comments are also identified where these comments are detected wrongly as their meanings are implicit where their interpretations are connected to pragmatics. Such context dependent text connotations are difficult to handle by explicitly

using the generated sentiment lexicons. For example, the Amharic Facebook comments **"ፍትህ ለአማራ"**/justice to Amhara/ is assigned negative sentiment by annotator, but lexicon based classifier wrongly detected it as positive. The reason is that the lexicon based classifier is limited in handling the contextual meaning of the comment in place. This comment is primarily connected to the context and implicitly associated to the meaning behind the original news post.

6.2.4. Summary

Amharic sentiment lexicon is one resource required for Amharic sentiment analysis. Yet, extensive lexical resources are expensive to build manually. To remedy this problem, a dictionary-based approach is proposed for generating Amharic sentiment lexicon. It requires a bilingual dictionary to propagate polarity information from sentiment lexicon of source language to target language. This approach is used to generate large-scale sentiment lexicons. The dictionary-based approach generates two Amharic sentiment lexicons: which are called afterwards "Amharic SO-CAL" (5681) and "Amharic SWN" (13677).
The lexicons generated using a dictionary-based approach are evaluated by agreement (internal), coverage (external), subjectivity detection rate (external) and its performance in lexicon-based Amharic text sentiment classification. The Amharic sentiment lexicon generated from SWN is not only good in internal evaluation (i.e. it has an acceptable agreement rate with both the manual and SO-CAL lexicons) but it also has higher coverage in both test corpora than the other two sentiment lexicons. The lexicon-based sentiment classification of Amharic sentiment lexicon from SWN outperforms the other Amharic sentiment lexicon (including the baseline).
This work showed that it is possible to auto-generate sentiment lexicons relying on bilingual dictionaries to minimize time and labor cost of manual sentiment lexicon preparation. The generated lexicon is expected to reach a sufficient sentiment lexicon size by porting it from resource rich language.
Some contributions of this work are briefly summarized below:

- The developed approach reduces cost and time of labeling terms in sentiment lexicon.
- The developed approach is generic enough that it can be adapted to generate sentiment lexicon to other less-resourced local languages.
- The entries in these generated sentiment lexicons contain part-of-speech information that can be applied or used in other linguistic tasks of interest.
- The approach can also be adapted to other tasks of natural language processing including information extraction, multilingual semantic lexicons, question and answering, just to name a few.

- The combination of all generated sentiment lexicons with the manual sentiment lexicon achieves best sentiment classification performance on Amharic texts.

Yet, the generated sentiment lexicons may lack accuracy for sentiment analysis of a particular language context where the approach potentially does not sufficiently capture the cultural and language specific connotations in a particular language. This is because the sentiment lexicon entries are obtained from another language. To address these issues, corpus-based approach is considered to capture and incorporate semantic information on the specific meaning of a term in a given context to provide higher precision in sentiment score assignment for each particular instance.

6.3 Corpus-based Lexicon Generation

6.3.1. Overview

Corpus-based approach is like dictionary-based approach because it uses seed list terms. Unlike a dictionary-based approach, corpus-based approach uses large corpora from which opinion words of similar co-occurrence pattern to the seed list terms are searched for based on specific strategies used in the algorithm. For instance, assume that if the new word frequently co-occurs with one entry of the Amharic seed words, they are assumed to have the closer semantic similarity. If a token's distance measure from a seed word is greater or less than a threshold, the token is added to the sentiment lexicon.

To minimize limitations of dictionary-based approach of capturing cultural connotation, a corpus-based approach is recommended to handle language and culture specific words in the sentiment lexicons [11]. However, it is unlikely to handle all the words in the language, as the corpora is not large enough to cover all tokens of the language. Given corpora in particular domains, it is possible to build sentiment lexicons. Because of this reason, the sentiment lexicon built using this approach is usually used for lexicon based sentiment analysis in the same domain from which it is built.

In this work, corpus-based approach is proposed to build Amharic sentiment lexicons automatically relying on the aforementioned Amharic corpora of category I and II described in section4.1.1.

Data are collected from different local news media organizations and also from Facebook comments.

6.3.2. Proposed Corpus-based Lexicon Generation

Corpus-based strategies include count-based and predictive-based approaches. In this subsection, count-based approach is proposed to generate Amharic sentiment lexicon from the corpora. The proposed approach is briefly specified:

Positive Point-wise Mutual Information (PPMI) Matrix : PPMI is a probabilistic measure of the semantic orientation between two items or words in a text. Detailed discussion of PPMI will be presented shortly. For construction of PPMI matrix, 572,921 documents tokenized into 46,907,990 tokens with vocabulary size of 1,671,158 are used. For building the matrix, the corpus stated in category I of section 4.1.1 is used. With this vocabulary, term-term co-occurrence semantic information is counted from these corpora collections. The frequency of term-term matrix is transformed to PPMI matrix, which further holds word-word occurrence probability distributions. The matrix with its vocabulary is written into a disk for further input into lexicon generation algorithms. This representation can capture frequently occurring bigrams to handle multi-word sentimental expressions. More details of this approach presented shortly.

In Figure 6.3, Corpus-based strategies include count-based and predictive-based approaches are depicted.

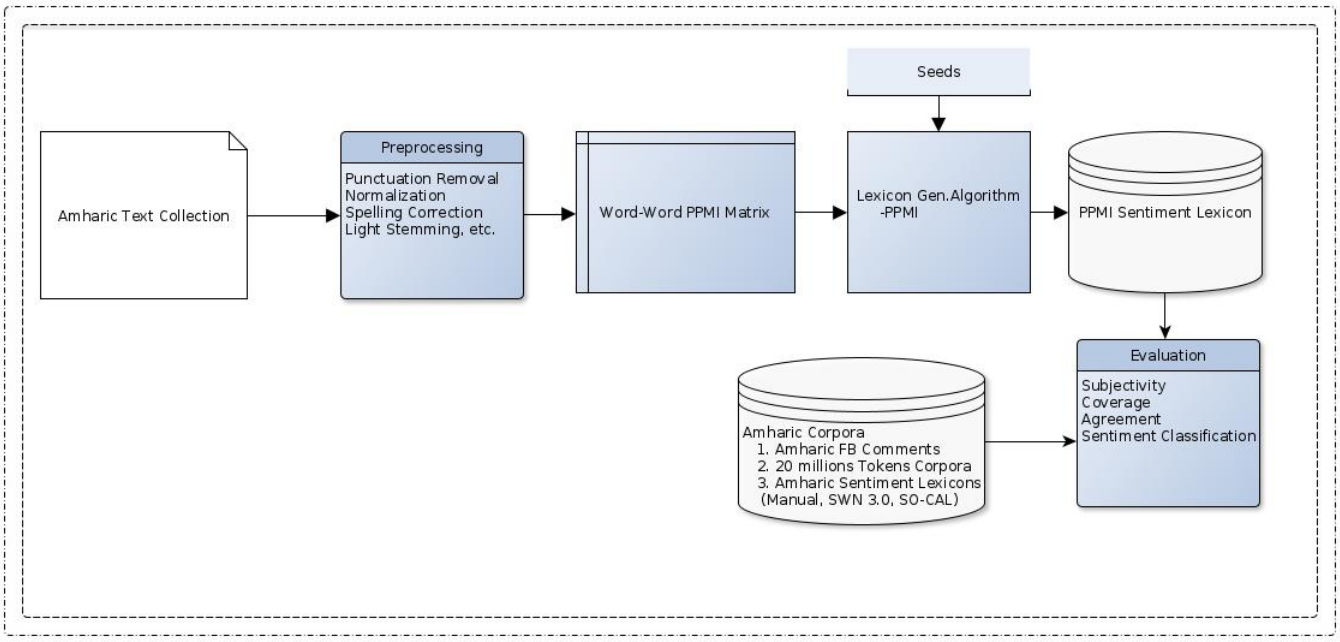


Figure 6.3.: Corpus-based Amharic Sentiment Lexicon Building Process

The algorithm and the seeds in Figure 6.3 are briefly described as follows.To generate Amharic sentiment lexicon, four major steps are followed:

1. Prepare a set of seed lists which are strongly negative and positively polarized adjectives, nouns and verbs (Note: as Amharic language contains few adverbs [108], adverbs are not taken as seed words). At least seven most polarized seed words are selected for each of aforementioned part-of-speech classes [155]. Selection of seed words is the most important step that affects the performance of bootstrapping algorithm [147]. Most authors choose the most frequently occurring words in the corpora as seed list. This is assumed to ensure the greatest amount of

contextual information to learn from. However, it is difficult to make sure about the quality of the contexts. Seed words are selected by adapting the procedures in Turney et al. [142] in which terms with high sentiment values (i.e. from both positive and negative) are considered. For example, /good/ and /bad/ are candidate seeds. After the candidate terms are computed relying on [142], the approach extends the original seed words in the lexicon. The seed expansion algorithm for building lexicon is presented in 6.2.

2. Build semantic space word-context matrix [114, 143] using the number of occurrences (frequency) of target word with its context words of window size±2. This word-word matrix can also be derived from word-document matrix using Latent Semantic Analysis (LSA). Word-context matrix is selected as it is more dense compared to other representations (i.e. word-document matrix) [114, 143] .

Initially, let F be a word-context raw frequency matrix with n_r rows and n_c columns formed from Amharic text corpora. Next, weighting functions apply to select word semantic similarity discriminant features. There are varieties of weighting functions to get meaningful semantic similarity between the word and its context. The most popular one is Point-wise Mutual Information (PMI) [142]. In this research, positive PMI is used by assigning 0 if it is less than 0 [27]. Let X be a new PMI based matrix got by applying Positive Point-wise Mutual Information (PPMI) to matrix F. Matrix X has the same number of rows and columns as matrix F. The value of an element f_{ij} is the number of times that word w_i occurs in the context c_j in matrix F. The corresponding element x_{ij} in the new matrix X is defined as follows.

$$x_{ij} = \begin{cases} PMI(w_i, c_j) \ if \ PMI(w_i, c_j) \ > \ 0 \\ 0 \ if \ PMI(w_i, c_j) \ \leq \ 0 \end{cases} \tag{6.2}$$

Where, $PMI(w_i, c_j)$ is the point-wise mutual information that measures the estimated co-occurrence of word w_i and its context c_j and is given as:

$$PMI(w_i, c_j) = \log \frac{P(w_i, c_j)}{P(w_i) \cdot P(c_j)} \tag{6.3}$$

Where, $P(w_i, c_j)$ is the estimated probability that the word w_i occurs in the context of c_j , $P(w_i)$ is the estimated probability of w_i and $P(c_j)$ is the estimated probability of c_i are defined in terms of frequency f_{ij}. In this step, problems associated to matrix computation and their corresponding possible remedies are presented as follows [143]:

- To make PMI based weighted matrix more efficient, one of smoothing techniques is applied.

3. Compute the cosine distance similarity between target term and centroid of seed lists (e.g. centroid for positive adjective seeds, $\overrightarrow{\mu^{+}_{adj}}$). To find the cosine distance of a new word from

seed list, first the centroids of seed lists of respective POS classes are computed; for example, centroids for positive seeds S+ and negative seeds S-, for adjective class are given by:

$$\overrightarrow{\mu^{+}_{adj}}(S^{+}) = \frac{\Sigma_{w \in S^{+}} \vec{w}}{|S^{+}|} \;\; \& \; \overrightarrow{\mu^{-}_{adj}}(S^{-}) = \frac{\Sigma_{w \in S^{-}} \vec{w}}{|S^{-}|} \tag{6.4}$$

Similarly, centroids of the other seed classes are found. The cosine distances of target word from positive and negative adjective seeds of centroids, $\overrightarrow{\mu^{+}_{adj}}$ and $\overrightarrow{\mu^{-}_{adj}}$ are given by:

$$cosine(\vec{w_i}, \overrightarrow{\mu^{+}_{adj}}) = \frac{\vec{w_i}.\overrightarrow{\mu^{+}_{adj}}}{||\vec{w_i}||||\overrightarrow{\mu^{+}_{adj}}||} \;\; \& \; cosine(\vec{w_i}, \overrightarrow{\mu^{-}_{adj}}) = \frac{\vec{w_i}.\overrightarrow{\mu^{-}_{adj}}}{||\vec{w_i}||||\overrightarrow{\mu^{-}_{adj}}||} \tag{6.5}$$

As word-context matrix x is vector space model, the cosine of the angle between two words vectors is the same as the inner product of the normalized unit word vectors. After obtaining cosine distances between word w_i and seed with centroid $\vec{w_i}, \vec{\mu^{+}_{adj}}$, the similarity measure can be found using either:

$$Similarity(\vec{w_i}, \overrightarrow{\mu^{+}_{adj}}) = cosine(\vec{w_i}, \overrightarrow{\mu^{+}_{adj}}) \tag{6.6}$$

Similarly, $Similarity(\vec{w_i}, \overrightarrow{\mu^{-}_{adj}})$ can also be computed. The sentiment score for each newly added candidate word obtains a sentiment value between zero (0) and a positive one (1). If the candidate word carries the highest sentiment, the angle between the vector of this word and the vector of the mean vector of the seed words is zero. The similarity score is computed using equation (6.6) (i.e. cosine of zero is one). In this way, this sentiment score for each candidate word is mapped and scaled to a real number between -1 and +1. Sentiment score of a target word is added to Amharic sentiment lexicon. Positive PMI with cosine measure is selected as it is performed consistently better than the other combination features with similarity distance metrics: Hellinger, Kullback-Leibler, City Block, Bhattacharya and Euclidean [27].

4. Repeat from step 3 for the next target term in the matrix to expand sentiment lexicon. Stop after a specified number of iterations threshold (i.e. obtained experimental trials) is reached.

The detailed algorithm for generating Amharic sentiment lexicon from PPMI is presented in listing 6.2

Algorithm 6.2: Amharic Sentiment Lexicon Generation Using PPMI

Input: $PPMI$: Load Word-Context PPMI matrix
Output: $AMLexiconByPPMI$: Generated Amharic Sent. lexicon

```
1  SeedNoun+, SeedNoun−, SeedAdj+, SeedAdj−,SeedVerb+, SeedVerb− ←Seed file;
2  PPMI ←Load PPMI Matrix file ;
3  VocabPPMI ←Load Vocabulary file;
4  AmharicSentimentLexByPPMI ←Null ;
5  foreach SeedLexicon ∈[SeedNoun+, SeedNoun−,SeedAdj+, SeedAdj−,SeedVerb+,
     SeedVerb−] do
6      foreach Seed ∈SeedLexicon: do
7          if Seed not ∈VocabPPMI then
8              Remove Seed from SeedLexicon ;
9      ThresholdsPPMI ←[100,200,300,400,500] ;
10     foreach Threshold ∈ThresholdsPPMI: do
11         foreach i←0 to Threshold: do
12             MeanVector←computeMean(SeedLexicon) by equation 6.4
13             TopiClosestTerms←compute_similarity(MeanVector,PPMI) by equation
                6.5
14             if term ∈TopiClosestTerms and term ∈SeedLexicon: then
15                 Remove the term from TopiClosestTerms list as it is duplicate;
16             Update SeedLexicon by inserting TopiClosestTerms list
17         AMLexiconByPPMI ←AMLexiconbyPPMI + SeedLexicon;
```

Algorithm description: The algorithm in listing 6.2 reads the seed lists and generates the merge of expanded seed lists using PPMI. Line 1 loads the seed lists and assigns them to their corresponding category. Similarly, from line 2 to 3 loads the lexical resources: pre-computed PPMI matrix and its vocabulary list and in line 4, the output $AmharicSentimentLexByPPMI$ is initialized to Null. From line 5 to 17, it iterates for each seed list polarity and categories. Line 6 to 8 checks that each seed term is found in the corpus vocabulary. Line 9-10 initializes the threshold by a selected trial number, i.e. 100,200,300,400,500 for PPMI. From line 11 to 16, it expands seeds either for PPMI based thresholds. From lines 12 to 16 it iterates from i=0 to threshold in order to perform the specified operations. Line 12 computes the mean of the seed lexicon based on equation 6.4 specified in the previous section. Line 13 computes the similarity distance between the mean vector and the PPMI word-word co-occurrence matrix and returns the top i closest terms to the mean vector, based on equation 6.6. Line 14-15 check the top i closest terms are not in the seed lexicon (i.e. removing duplicates). Line 16 updates these filtered top i terms into the seed lexicon. Finally, line 17 appends the expanded seed lexicon into the Amharic sentiment lexicon by PPMI.

However, it probably goes through unnecessary multiple iterations. As a result, semantic drift can occur while expanding seed words. This probably incorrectly diverts the semantic orientation of the newly expanded words, which may be far away from the sentiment category

of the original seed list.
In the literature, there are no clear guidelines to decide when to stop iteration of bootstrapping algorithms. Issues related to seed list selection decision, loop stopping condition and semantic drift are the major downsides of most bootstrapping algorithms [147]. As a result, one needs to take care while employing bootstrapping algorithms.
Using corpus-based approach, Amharic sentiment lexicon is built which allows finding domain dependent opinions which is not possible by sentiment lexicon generated using dictionary-based approach. A corpus-based approach is developed by bootstrapping relying on word-context semantic space representation of large Amharic corpora. As it uses the mono-lingual corpora for sentiment word generation, it is assumed that it captures sentiment words carrying cultural connotation specific to Amharic. The quality of this corpus-based lexicon is evaluated using similar techniques to those discussed for dictionary-based approaches. However, this approach probably does not produce sentiment lexicon with large coverage, as the corpora size does not include all polarity words in Amharic language.

6.3.3. Results and Discussions

Datasets: Amharic corpora with 572,921 documents and 2,871 Facebook comments as described in category I and II of section 4.1.1 are used. These corpora are used to build word-context representations. Initially, the data set is cleaned. The text is tokenized with Gensim bigram tokenizer and then stemmed with light stemmer adapted from [9] as specified in preprocessing steps of chapter 5. PPMI matrix is built. These pre-computed matrix representations are written into a file for further tasks of corpus-based Amharic sentiment lexicon generation (i.e. using PPMI).
Amharic Sentiment Seed Words: Amharic opinion words are manually prepared with high sentimental strength (i.e. either positively or negatively) with either of three parts-of -speech categories: adjectives, nouns and verbs. Adverbs are ignored as adverbial words in Amharic language almost do not exist [155]. Each seed category is expanded from pre-computed PPMI. These seed words are presented in Table 6.5.

Table 6.5.: Selected Seeds for Corpus-based Amharic Sentiment Lexicon Expansion

Part-of-speech	Negative	Positive
Adjectives	99	79
Nouns	145	120
Verbs	36	40

Light stemmer is built to reduce surface words into corresponding stems with minimal loss of semantic information. The number of rules in the sentiment is reduced as compared to the root stemmers in [9, 48]. For example, sample of positive and negative stemmed words and their possible corresponding surface words are presented in Table 6.6

Table 6.6.: Sample list of Terms in the Generated Amharic Sentiment Lexicon using PPMI with their corresponding Surface Words

Stem	POS	Sentiment	Sample Surface Words
ሀ-ስ-ት/'fake'/	noun	-0.82	['ሀሰተኛ/liar/,ሀሰት/fake/, የሀሰት/fake/, ከሀሰተኛ/from fake maker/,ሀሰተኛና/liar and/,የሀሰተኛ/for fake maker/,ለሀሰተኛ/to liar/,ሀሴት/pleasure/, ታህሳት/, ሀሰትነት/fakeness/,ሀሰትን/fake/ , ከሀሰተኛው/from the liar/,ሀሰተኛው/liar/,ሀሰቱን/liar/, ከሀሰት/from lie/,ሀሰትና/fake and/,ከሀሰቱ/from the fake/,ሀሴትና/pleasure and/,ሀሴትን/the pleasure/,ሀሳት/fake/,ለሀሰትና/to fake and/, ሁሰት/,የሀሰትና/for fake and/,ለሀሰት/to fake/,ሀሰቶች/a lot of fake/,etc...]
ሀ-ቅ /'fact'/	Noun	+0.81	[ሀቆች/facts/ ሀቅ/fact/,ሀቀኛ/honest/, ሀቅን/the truth/, ከሀቅ/from fact/, ሀቁ/the truth/,ሀቁን/the truth/, የሀቀኛ/the one who is honest/, ሀቁን/the truth/, ሀቀኛና/the truth and/, ሀቅና/truth and/, ከሀቁ/from the truth/, ሀቀኛውን/that who is honest/, ሀቆችን/facts/, ከሀቀኛ/from facts/, ሀቅነት/truthness/, ሀቅም/truth/, የሀቅ/for truth/, ሀቀና/fact and/,ለሀቅና/for truth and/, ለሀቅ/for truth/,የሀቅነት/for truthiness/,ሀቅ/fact/, ሀቃዊ/honesty/, ስለሀቅ/for truth/, ሀቀኛው/the one who is honest/,etc...]

Facebook Comment: This data set is used to evaluate subjectivity detection, lexicon coverage, and lexicon-based sentiment classification of the generated Amharic sentiment lexicon. The data set is manually annotated by GCOA professional and it is labeled either positive or negative.

6.3.4. Evaluation

Amharic sentiment lexicons are generated by expanding the seeds relying on the PPMI matrix of the corpora by implementing algorithm 6.2 at threshold values of 100,200,300,400 and 500. See sentiment lexicons of varied sizes' evaluation details in terms of agreement with manual sentiment lexicon, coverage on "20m Amharic token corpus" and subjectivity detection on Facebook comments by the generated PPMI-based sentiment lexicon with the above threshold values in Table 6.7. Further discussion of evaluation is provided for the generated PPMI-based sentiment lexicon size of 8132 at threshold of 500.

Table 6.7.: PPMI-based Sentiment Lexicons' Performance Evaluations

Threshold	Lex_Size	AgreeWith_Manual	CoverageOn_20m	Subject_detection
100	1068	83.429	25.453	88.83
200	2903	73.672	39.479	95.05
300	4660	66.667	51.460	98.36
400	5740	67.864	46.820	97.06
500	8132	64.207	51.460	98.36

In corpus-based approach, the PPMI-based sentiment lexicon is written to disk with entries containing stemmed word, part of speech, polarity strength and polarity sign.
The generated PPMI-based sentiment lexicon is evaluated in two ways: external to sentiment lexicon and internal to sentiment lexicon. External to sentiment lexicon is done to test the correctness of the sentiment information gained in each of the sentiment lexicons. Sentiment lexicons are used to compute the aggregated sentiment score of each of the Amharic Facebook comments. If the computed score is the same polarity as sentiment labels of the Facebook comment in corpora, the sentiment lexicon detects sentiment of the Facebook comment correctly. Internal evaluation is to test the extent to which generated sentiment lexicon overlaps (or agrees) with manual sentiment lexicon.
Subjectivity Detection: In this part, the accuracy of the subjectivity detection rate of generated sentiment lexicon is evaluated on 2871 Facebook comments. The procedures for aggregating sentiment score are done by summing up the positive and negative sentiment values of opinion words found in the comment if those opinion words are also found in the sentiment lexicon. If the sum of positive score is greater or less than the sum of negative score, the comment is said to be a subjective comment, otherwise the comment is neutral or mixed. Based on this technique, the subjectivity detection rate is presented in Table 6.8.

Table 6.8.: Comparison of the PPMI-based Sentiment Lexicon' Subjectivity Detection rate (in%) on Facebook Comments

Sentiment Lexicons	Detection withNoStem	Detection withStem
Amharic Manual(Baseline)	31.99	93.56
Amharic SO-CAL	31.28	96.65
Amharic SWN	75.83	99.33
Amharic PPMI	84.50	98.36

Discussion: The subjectivity detection rate of the PPMI -based sentiment lexicon and other dictionary based sentiment lexicons are depicted in Table 6.8. The detection rate of PPMI-based sentiment lexicon performs better than the manual sentiment lexicon (the baseline). whereas the sentiment lexicon from SWN outperforms the PPMI-based sentiment lexicon with 1% accuracy.

Generated Lexicon's Coverage: Coverage of the generated PPMI-based sentiment lexicon is evaluated externally by computing the proportion of terms in the generated sentiment lexicon found in the corpora. The coverage of PPMI-based sentiment lexicon on Facebook comments and a general Amharic corpus collection is presented in Table 6.9.

Table 6.9.: Comparison of PPMI-based Sentiment Lexicon' Coverage (positive/negative Count and in Percent)

Lexicons	2871 Amharic Comments		20 Millions Amharic Tokens	
	coverage(+,-) count	in%	coverage(+,-) count	in %
Manual	[4399, 2995]	31.95	[2713167, 2161501]	25.01
SO-CAL	[5738, 3953]	41.87	[4169817, 3391213]	38.88
SWN	[9447, 4803]	61.57	[6645592, 4006072]	54.77
PPMI-Lex	[849, 11424]	60.93	[964235, 7813415]	51.46

Discussion: Table 6.9 shows that the coverage of PPMI based sentiment lexicon has better coverage compared to the manual lexicon and SO-CAL in both Facebook and 20 millions Amharic tokens corpora. However, PPMI-based sentiment lexicon has a poor balance of positive

and negative terms. Unlike SWN, PPMI-based sentiment lexicon is generated from corpora. Because of this reason, its coverage for a general domain is limited.

Lexicon Agreement: In this part, the agreement (or overlap) among sentiment lexicon's terms showing to what extent the generated PPMI-based sentiment lexicons agreed (or overlapped) with the other lexicons (e.g. manual sentiment lexicon) is also evaluated. This type of evaluation (internal) validates by telling the percentage of entries in the sentiment lexicons which are in common. A greater percentage means better agreement or overlap of sentiment lexicons. The agreement of PPMI-based sentiment lexicon (in percentage) is presented in Table 6.10.

Table 6.10.: The Agreement Between PPMI-based Sentiment Lexicon and Other Lexicons

-	Lexicons combinations	Agreement (in %)
1	PPMI and SO-CAL	67.34
2	PPMI and SWN	97.93
3	PPMI and Manual	64.21

Discussion: Table 6.10 presents the extent to which the PPMI-based sentiment lexicon agrees with other sentiment lexicons. PPMI-based sentiment lexicon has got the highest agreement rate (overlap) with SWN sentiment lexicon.

Lexicon based Amharic Sentiment Classification: In this part, Table 6.11 presents the lexicon-based sentiment classification performance of generated PPMI-based sentiment lexicon on 2871 Facebook comments. The classification accuracy of generated PPMI-based sentiment lexicon and other sentiment lexicons are compared.

Table 6.11.: The Accuracy of PPMI-based Sentiment Lexicon for Sentiment Classification

Lexicons	**Accuracy(%)**	
	NoStem+NoNeg.	Stem+Neg.
Manual(baseline)	16.7	52.7
PPMI-Lex	56.13	60.22
Manual+PPMI	78.96	85.29

Discussion Besides the other evaluations of the generated PPMI-based sentiment lexicon, the usefulness of this sentiment lexicon is evaluated. As depicted in Table 6.11, the accuracy of

both PPMI-based sentiment lexicon for lexicon-based sentiment classification is better than the manual sentiment lexicon, i.e. the baseline sentiment lexicon.
Section 6.2 presented the discussion of dictionary-based sentiment lexicon for lexicon-based sentiment classification and showed that the sentiment lexicon has improved performance of sentiment classification by applying light stemming and negation-handling. Besides, the combination of the two sentiment lexicons (PPMI + Manual) achieved an increase of accuracy of 43% over the performances of the manually built lexicon.

6.3.5. Lexicon with Threshold Sizes

This subsection investigates the relationship between performance and threshold. That is, as threshold increases, its effect on sentiment lexicons' performance in terms of its agreement, its coverage and its subjectivity detection has been evaluated.

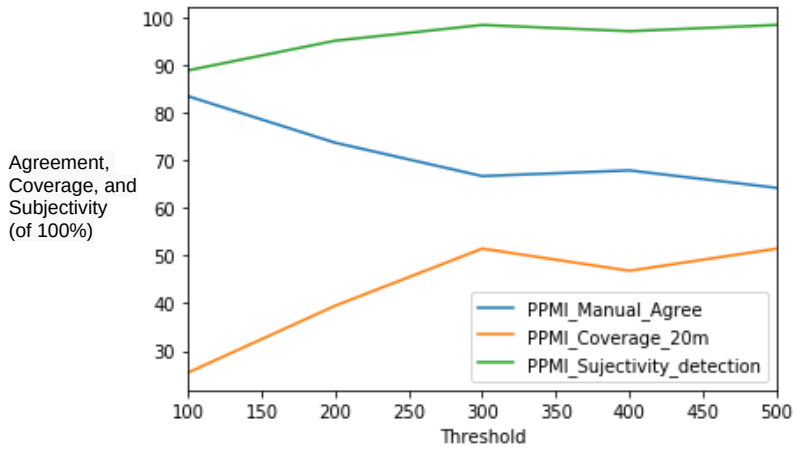


Figure 6.4.: PPMI based Lexicon Evaluation with Threshold sizes

As it is depicted in Figure 6.4, coverage (25% to 51%) and subjectivity (88% to 98%) of PPMI-based sentiment lexicon increases sharply from threshold 100 to 300 whereas it remains constant above a threshold of 300. However, agreement of PPMI-based sentiment lexicon with manual sentiment lexicon does the opposite with these same threshold values. As threshold increases from 100 to 300, the agreement of PPMI-based sentiment lexicon with manual sentiment lexicon decreases from 83% to 66%. Above the threshold of 300, its agreement is constant.

6.3.6. Summary

Generating sentiment lexicons is not an objective process. The generated PPMI-based sentiment lexicon depends on the domain in which it is used. This work has shown that it is possible to create sentiment lexicon for less-resourced languages from corpus collections. This

possibly captures the language-specific features and connotations related to the culture where the language is spoken and written.
The key contributions of this section are: (i) the development of corpus-based approach using PPMI matrix that generates Amharic sentiment lexicon with minimal manual inspection and can capture sentiment specific to Amharic language, (ii) the generated sentiment lexicon can enhance the performance of sentiment classifications in Amharic language, (iii) the proposed approach is a generic approach which can be adapted to other less-resourced languages to reduce cost of human annotation and the time to annotate sentiment lexicon and (iv) the collected, prepared and generated sentiment lexicon can be accessible for further enrichment of tasks of Amharic sentiment classification. In the future work, as the PPMI approach is based on a frequency count-based approach which can not have a capability to predict the co-occurring contexts. A word embedding model is suggested to have predictive properties, so that given a word, neighboring context words can be predicted (i.e. retrieved).

6.4 Human Evaluation of Lexicons

The generated sentiment lexicons are evaluated by two professionals (whose background were linguistics and psychology). First, a sample of size 100 sentiment words are randomly chosen from each of the three generated lexicons: Amharic SO-CAL, Amharic SWN and Amharic PPMI. The two professionals have been asked whether the automatically assigned polarity for each of the sentiment words is accepted or not. Using Google form, their responses for each of the three samples are recorded. Kappa of the two raters is computed using equation 4.1. The Kappa values for each of the generated sentiment lexicons are depicted in Table 6.12.

Table 6.12.: Two human evaluation of the generated sentiment lexicons using Cohen Kappa[92]

-	Lexicons	Agreement (in Kappa)
1	Amharic SO-CAL	0.851
2	Amharic SWN	0.839
3	Amharic PPMI	0.798

Discussion: Table 6.12 depicts the Kappa value for the inter-agreement between two human judgments of the sample of 100 sentiment words randomly drawn from each of the three Amharic sentiment lexicons (i.e. Amharic SO-CAL, Amharic SWN and Amharic PPMI) which are automatically generated in the earlier sections. To test the reliability of the lexicons, Cohen Kappa value is computed. The kappa value of Amharic SO-CAL (0.85) is the highest agreement between the two human evaluators. This shows that the agreement between the two annotators

is more consistent than the agreement on the other sentiment lexicons. Next to Amharic SO-CAL, Amharic SWN with Kappa of 0.83 is more reliable than the Amharic PPMI lexicon (i.e. Kappa of 0.79). The dictionary based lexicons are more reliable and accurate than the corpus based lexicons. Some of the on-spot errors have been identified from the sentiment words which are not accepted by both of the human evaluators. There are a number of sentiment words which are incorrectly generated in the process of corpus based lexicon generation than dictionary based lexicon generation. For example ትልቁ meaning "the big" is generated as negative sentiment word by the process of corpus based method. Usually the word "big" will have a positive sentiment orientation. A possible explanation is that the word might have opposite sentiment depending on its context it is used (i.e. because of the issues of context sensitivity or such words might be used as part of irony or sarcasm expressions in the corpora).

7. Machine Learning for Amharic Sentiment Classification

This chapter comprises three major sections: part I machine Learning for Amharic sentiment classification, part II Ensemble Learning for Amharic sentiment classification and part III BERT fine-tuning for Amharic sentiment classification. Part I tests conventional machine learning for Amharic sentiment classification, part II combines the best machine learning for Amharic sentiment classification (this is ensemble learning) and part III builds an Amharic BERT model which is fine-tuned and tested for Amharic sentiment classification.

7.1 Part I. Machine Learning

Part I is organized into seven subsections. The first subsection discusses the feature and model selection and corresponding discussion of experimental results is presented in the second subsection. The third subsection investigates the hyper-parameter tuning and discussion of associated experimental results are presented in the fourth subsection. The fifth subsection reports the hyper-parameter spaces for which feature sets combined with different machine learning algorithms improve performance for sentiment classification. The performance of the best selected algorithm with the best feature set is investigated with different training sizes in the sixth subsection. The last subsection presents the conclusions.

7.1.1. Feature and Model Selection Evaluation

In this study, the Facebook comments are partitioned into training and testing sets. This text data should be transformed to feature vectors for input to the most popular machine learning text classification algorithms [73] Logistic Regression (LR), Naive Bayes (NB), Support Vector Machine (SVM) and Random Forest (RF). For each model, mean accuracy is computed using 5 fold CV. The result is presented and compared to see the extent of performance improvement on each of the features for sentiment classification. The best features with best algorithms of high predictive performance are selected for further sentiment classification tasks. The above machine learning algorithms are used to distinguish the best text feature sets for sentiment classification of Facebook comments. Features for Amharic sentiment classification are identified

and proposed. Experiments for identifying features with best performance are set up. Features include BOW character/word ngram feature sets, TF-IDF character/word ngram feature set, average word2Vec, TF-IDFWeighted word2Vec, LDA, LSI and NMF.
The defaults and search spaces for different hyper-parameters of the above algorithms are depicted in Table 7.1.

Table 7.1.: Popular ML Hyper-parameters' Spaces

Classifier	Hyper-parameter	Type	Default Values	Search Spaces
CountVectorizer	analyzer	discr	word	None
CountVectorizer	max_df	cont	1	None
CountVectorizer	max_features	discr	None	None
CountVectorizer	ngram_range	disc	(1,1)	None
TF-IDFVectorizer	Same as above	Same as above	Same as above	None
TF-IDFVectorizer	use_idf	discr	TRUE	None
LDA	n_components	discr	10	[1,250]
LDA	shrinkage	cont	None	[0.0,1.0]
LDA	solver	discrete	None	[eigen,lsgr, svd]
LDA	tol	cont	None	[0.00001, 0.1]
TruncatedSVD	n_components	None	2	None
TruncatedSVD	tol	None	0	None
TruncatedSVD	n_iter	None	5	None
TruncatedSVD	random_state	None	None	None
NMF3	max_iter	None	200	None
NMF4	n_components	None	None	None
NMF5	tol	None	0.0001	None
NMF6	random_state	None	None	None
LR	C	cont	1	None
LR	class_weight	discr	None	None
LR	alpha	cont	None	[0.0000001, 0.1]
LR	average	discr	None	[false, true]
LR	penalty	disc	l2	[elasticnet, l1, l2]
LR	power_t	cont	None	[0.00001,1]
LR	tol	cont	0.0001	[0.00001,0.1]
Multinominal NB	alpha	con	1	[0.01, 100]
Multinominal NB	fit prior	cat	TRUE	[false, true]
SVM	C	con	1	[0.01, 10000]
SVM	coef0	con	0	[-1, 1]
SVM	class_weight	disc	None	None
SVM	degree	discr	3	[2, 5]
SVM	gamma	con	scale	[1, 10000]
SVM	kernel	disct	rbf	[poly, rbf, sigmoid]
SVM	shrinking	disct	TRUE	[false, true]
SVM	tol	con	0.001	[0.00001, 0.1]
Decision Tree	criterion	discr	gini	[entropy, gini]
Decision Tree	class_weight	discr	None	None
Decision Tree	max depth	discr	None	[1, 10]
Decision Tree	min samples leaf	discr	1	[1, 20]
Decision Tree	min samples split	discr	2	[2, 20]
Random Forest	bootstrap	discr	TRUE	[false, true]
Random Forest	class_weight	discr	None	None
Random Forest	criterion	disc	gini	[entropy, gini]
Random Forest	max features	con	auto	[0.0, 1.0]
Random Forest	min samples split	disc	2	[2, 20]
Random Forest	min samples leaf	discr	1	[1, 20]
Random Forest	n estimators	discr	100	[2, 100]

The default values are used for selecting the feature set which performs best for Amharic sentiment classification while feeding it to a particular machine learning algorithm.

7.1.2. Results and Discussion of Feature and Model Selection

For each of these text features identified, the most popular machine learning algorithms (SVM, NB, RF) with their default hyper-parameter setting are trained with a training set using Cross-Validation (CV) of 5 folds using PMO Facebook comments datasets described under category

II in chapter 4. The mean accuracy is generated for each feature with each of the proposed machine learning classifier combinations. The feature evaluation result is presented in Table 7.2

Table 7.2.: Models and Features Selection Performance

Model+Feature	Accuracy
SVM+char_TFIDF	0.8262
SVM+Word_TFIDF	0.7928
Mult_NB+Word_BoW	0.7912
RF+Char_TFIDF	0.7873
RF+Char_BoW	0.7864
SVM+Char_BoW	0.7819
RF+Word_TFIDF	0.7725
RF+Word_BoW	0.7685
RF+LSI	0.7640
SVM+Word_BoW	0.7557
RF+NMF	0.7465
Mult_NB+Word_TFIDF	0.7441
Bern_NB+Word_BoW	0.7129
Bern_NB+Word_TFIDF	0.7129
SVM+LDA	0.7014
SVM+LSI	0.6908
SVM+W2V	0.6906
SVM+W2V_TFIDF	0.6906
RF+W2V	0.6906
RF+W2V_TFIDFf	0.6906
Etree+W2V	0.6906
Etree+W2V_TFIDF	0.6906
Mult_NB+W2V	0.6906
Mult_NB+W2V_TFIDF	0.6906
SVM+NMF	0.6905
RF+LDA	0.6661

Discussion: The performance of each feature with classifiers are sorted and presented in Table 7.2. To put it in a nutshell, for CV of 5 folds on training sets, TF-IDF character level and TF-IDF word level with SVM are the best performing features (82.62 and 79.28) compared to features (i.e. PCA, LDA, NMF, LSI, w2Vec, etc.). The second best machine learning model with the BOW character level and TF-IDF character level text feature set is Random Forest (accuracy of 78.73 and 78.64), respectively. Similarly, Multinomial NB with BOW word level features also performs better (accuracy of 79.12) than RF. Random forest with LDA is the least important feature set (i.e. accuracy of 66.61%). For improved model performance, the top three machine learning algorithms (NB, SVM and RF) with TF-IDF feature sets are potential candidates for the process of optimization using hyper-parameter tuning in the next section.

7.1.3. Hyper-parameter Tuning of Features and Models

A model learns how to separate boundary or probability distribution of a training set, and optimization techniques are a method for searching the best parameters of the chosen model. Hyper-parameter tuning is a technique to seek the combination of hyper-parameter values; for this, their score computation on validation data is required. It is noted that tuning is not done with testing data. Rather, testing data is only used once for evaluating the ultimate model before the model is deployed in a real-time application. For tuning the model, Cross-validation (CV) is used. Even though, the use of CV for hyper-parameter tuning is a computationally time-consuming step, CV is a safe method to avoid over-fitting. The two popular simple hyper-parameter search techniques are Grid search and Randomized search. Grid search is a method that uses exhaustive guess search (brute force) that relies on past evaluation results. One major problem is that if a list of hyper-parameter values in the target domain is not available. Another problem is that it is computationally expensive as it tries to search for all hyper-parameter search spaces combinations for each cross-validation step. For a computational scarcity scenario, it is suggested for small numbers of hyper-parameters as it considers every combination of hyper-parameters.

With training data, multiple models are built by changing the values of different hyper-parameters. Among those models, the best model is selected using a hyper-parameter tuning procedure. In this research, grid search strategies are proposed to extract the model parameter setting at which the model performs better. This grid search algorithm can easily be accessed from Scikit learn python library.

Cross-validation with 5 folds is used with grid search and the average performance of those models in each fold are computed. The most popular machine learning in sentiment analysis is tested and selected the best features and best machine learning algorithms. The proposed algorithms include Naive Bayes, support vector machine, logistic regression and random forest. Relying on the training data, different machine learning models are developed from which the top best performing models are selected by using validation data. The selected models can further be optimized for predicting unseen texts.

For improved selected model performance, gridSearch CV is applied for hyper-parameter tuning of those best performing algorithms with their default parameter settings obtained in Scikit learn library. For finding hyper-parameters of TF-IDF vectorizer and selected machine learning models, the search spaces for each of these machine learning algorithms are obtained from the literature [115]. This is one of the reasons we choose gridSearch over randomized (or "uninformed") search. See hyper-parameters of those classifiers in Table 7.1.

7.1.4. Results and Discussion of Feature Hyper-parameter Tuning

As per the feature and model selection result stated in Table 7.2, TF-IDF feature set performs best across the machine learning algorithms compared to other feature sets. In this section, before machine learning is optimized, first hyper-parameters of TF-IDFVectorizer is optimized using Scikit learn library. The best fitted hyper-parameters of TF-IDVectorizer is presented in Table 7.3.

Table 7.3.: Best Feature Vectors with their Hyper- parameters

Vectorizer	Parameter	Type	Default	Search Spaces	best_params
TF-IDFVectorizer	analyzer	discr	word	[word, char]	char
TF-IDFVectorizer	max_features	discr	None	[500,2500,5000]	5000
TF-IDFVectorizer	ngram_range	disc	(1,1)	[(1,2), (1,3),(1,5),(1,7)]	[(1,7)]
TF-IDFVectorizer	use_idf	discr	TRUE	[True, False]	True

Discussion: As indicated in Table 7.3, the hyper-parameters of TF-IDF (i.e. TF-IDFVectorizer) ngram features are searched for the best hyper-parameter values of ngrams, analyzer and max-features using gridsearchCV for sentiment classification relying on SVM. TF-IDF vectorizer feature achieves performance gains at the settings analyzer = character level, ngram_range = (1,7), and max_feature = 5000. So, for the rest of the proposed machine learning approaches, TF-IDF character level (1,7) grams feature set is used as input to be learned by the classifier for sentiment classification of Amharic Facebook comments. A similar effect of the number of characters in the ngrams was obtained for Arabic sentiment classification in [42].

7.1.5. Results and Discussion of Model Hyper-parameter Tuning

With best hyper-parameter of TF-IDF character vectorizer, hyper-parameters of selected machine learning algorithms(sequential model, NB, SVM, RF) are using grid search with CV of 5 folds on PMO Facebook comments datasets described under category II of chapter 4. The best scores with the obtained parameters of corresponding algorithms are presented in Table 7.4. As can be seen in Table 7.4, the best scores with the best parameters are not significantly greater than the scores with default parameters in each of the classifiers. See their scores with default parameters in Fig. 7.1.

Table 7.4.: Selected ML Best-parameters' with 5CV

Classifier	Hyper-parameter	Type	Default Values	Search Spaces	best_params	best_scores
Multinominal NB	alpha	con	1	[0.01, 100]	0.01	0.795401
SVM	C	con	1	[0.01, 10000]	1.5	0.808331
SVM	class_weight	disc	None	None	None	0.808331
SVM	gamma	con	scale	[1, 10000]	auto	0.808331
SVM	kernel	disct	rbf	[poly, rbf, sigmoid]	rbf	0.808331
Random Forest	class_weight	discr	None	[None,'balanced']	balanced	0.7741
Random Forest	criterion	disc	gini	[entropy, gini]	gini	0.7741
Random Forest	max features	con	auto	(None, 'auto', 'sqrt', 'log2')	None	0.7741
Random Forest	min samples split	disc	2	[1,5,10]	10	0.7741
Random Forest	min samples leaf	discr	1	[1,5,10]	5	0.7741
Random Forest	n_estimators	discr	100	[100, 250, 500]	500	0.7741
KC	dropout	con	0	[0.3, 0.2, 0.1, 0]	0.3	0.787879
KC	optimizer	discr	sgd	['Adam','RMSprop', 'Adamax', 'sgd']	Adagrad	0.787879
KC	epochs	discr	10	[10, 40, 80]	40	0.787879
KC	init	discr	uniform	['uniform', 'zeros', 'normal']	uniform	0.787879
KC	batch_size	discr	0	[2, 16, 32,64]	64	0.787879

Discussions: As depicted in Table 7.4, the best performance of the machine learning algorithms with the best hyper-parameter settings is obtained for sentiment classification. SVM with best hyper-parameters (C = 1.5, gamma = auto, kernel = rbf) outperforms the other machine learning algorithms with their corresponding best hyper-parameters for sentiment classification. So, SVM is the selected machine learning algorithm that works best for Amharic sentiment classification using Facebook comments. With the obtained hyper-parameter values, it is re-trained those models to see their performance on test sets.

7.1.6. Effect of Varying Quantity of Training Data

Investigation of the effects of training size is needed in order to identify the minimum training set size required for training machine learning algorithms for sentiment classification in less-resourced languages like Amharic with acceptable performance. This is important to benchmark the sentiment labeled data sets needed to build a sentiment classifier. With folds of five (5) CV on PMO Facebook comments under category II, the effect of training data set sizes of (10, 40,160,640,3200,6300) is observed on performance of the models with the aforementioned features. The performance of all models increases with training size. SVM is used with text feature of character-level ngram of size 7 with TF-IDF weights. The effects of different text features on sentiment classification are compared using SVM. The performance (using average accuracy) of the above proposed machine learning classifiers are evaluated and the relationship between their performance and training set size is plotted in Figure 7.1.

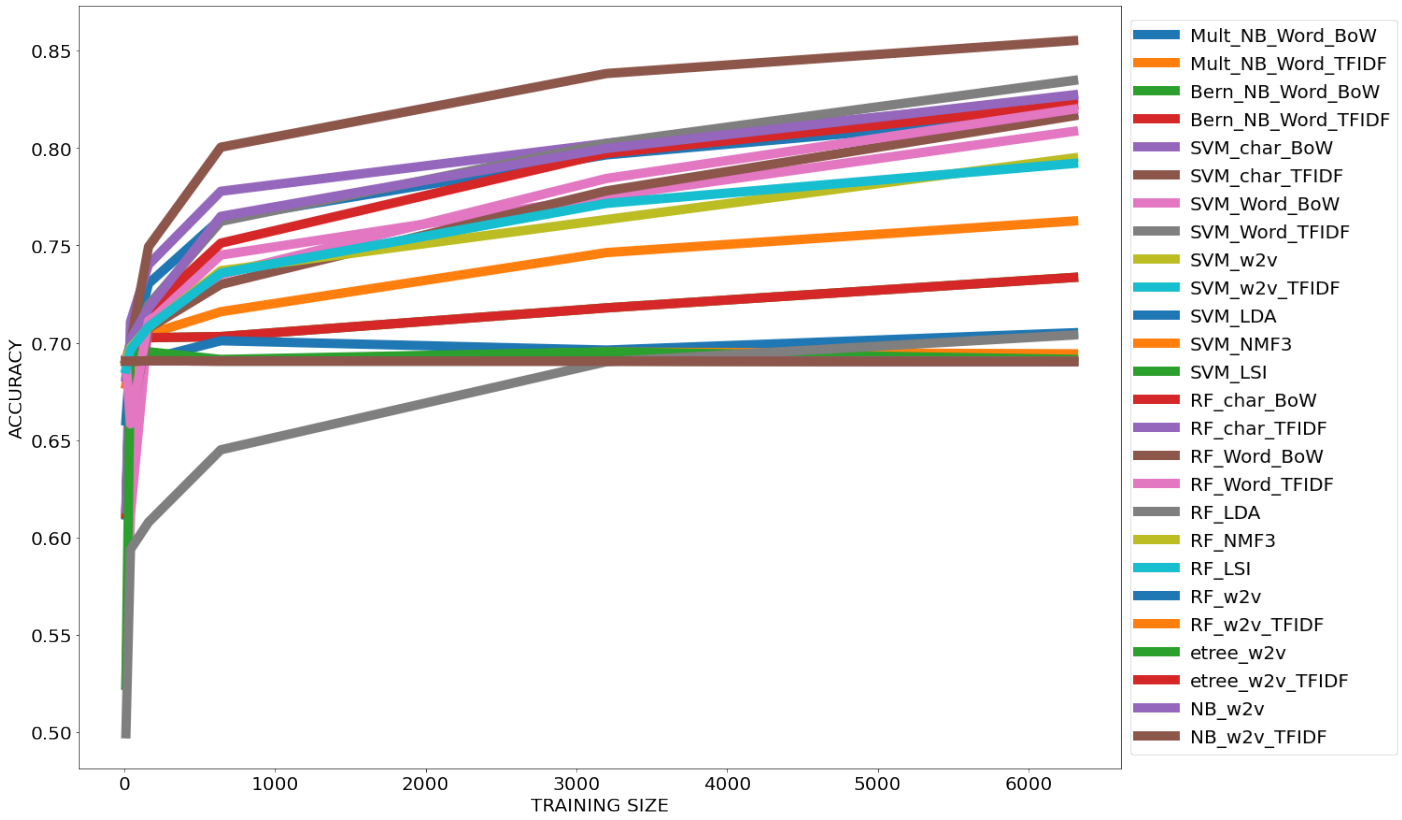


Figure 7.1.: Effects of Training Size on Performance of Machine Learning Classifiers

Discussion: As depicted in Figure 7.1, the performance of the machine learning algorithms for sentiment classification gradually increases as the training size increases from 1000 to 6000. With training size less than 1000, there is a sharp increase in performance of each of the classifiers. SVM with TF-IDF character level (1,7) grams feature sets outperformed all other classifiers with the same feature sets using CV of 5 folds. The RF with LDA feature set performs the poorest of all classifiers and input feature set combinations. In general, it is concluded that SVM performs sentiment classification very well (with average accuracy of over 80%) and progressively increases its performance as the training data size increases, starting from 1000 samples.

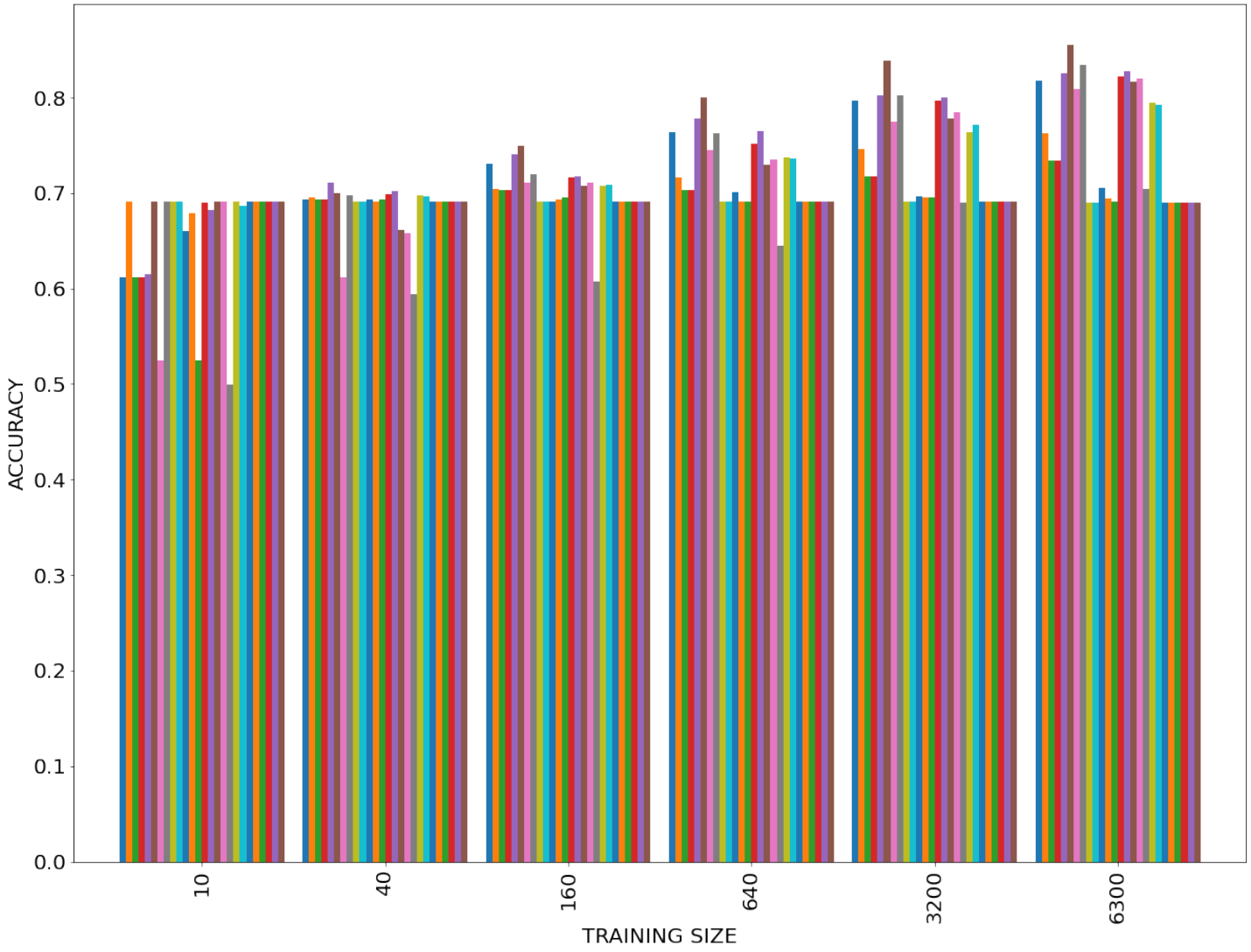


Figure 7.2.: Feature Selection on Facebook Comments using Machine Learning Models

Discussion: The bar plot in Figure 7.2 is an other way of visualizing the performance of machine learning classifiers with different combinations of input text features with respect to varying training sets, displaying the same result as in Figure 7.1.

7.1.7. Summary

This section presents a summary of part I.

(i) Feature set Selection: The best text feature sets for sentiment classification in social media are identified, extracted and selected: CountVectorizer character/word ngram features, TF-IDF Vectorizer character/word ngram features, averaged word embedding feature vector, TF-IDFWeighted word embeddings feature vector, LDA, LSI and NMF. All these feature sets are extracted from data sets. Each of the feature set is used to train the most common machine

learning algorithms, i.e. SVM, NB, and RF. With default values of these algorithms, the best performing feature set is TF-IDFVectorizer character feature set using SVM algorithm.

(ii) Model Selection and Hyper-parameter Tuning: Among the most popular machine learning algorithms, i.e. SVM, NB, and RF, SVM exhibits the best sentiment classification performance compared with other classifiers. SVM's hyper-parameter is further optimized by using grid search CV using training sets with the TF-IDFVectorizer character ngram feature sets. Similarly, the hyper-parameters of other machine learning algorithms are also optimized. The final SVM with its optimized hyper-parameter values consistently achieves performance gains in sentiment classification as compared to the other classifiers (i.e. RF, NB).

(iii) Performance on Varying Training Size: The performance of SVM with its tuned hyper-parameters is tested to determine what extent it works with insufficient labeled training datasets. The performance of SVM on sentiment classification increases progressively as the training size increases from 1000 samples to 6000 samples. This result revealed that optimized SVM can classify sentiment on Facebook comments very well under insufficient labeled samples.

7.2 Part II. Ensemble Learning for Amharic Sentiment Classification

Part II is organized into four subsections. The first subsection provides strategies of ensemble learning methods. The second subsection presents the proposed stacking ensemble approach. Results and discussions are presented in the third subsection. The last subsection presents summary.

7.2.1. Overview

Part II presents the performance of combinations of machine learning algorithms (i.e. base learners) on sentiment classification. The base learners NB, LR, SVM, RF with their tuned hyper-parameters based on the report in chapter 7.1 are combined. A method of combining machine learning classifiers is called ensemble learning (or ensemble method).

Ensemble Learning: Ensemble learning is a technique for combining various base-learners from which a new classifier is created, so that the new classifier improves performance compared to any of its constituent base learners. In ensemble learning, different base learners of the same or different types are combined using different strategies. The method of combining classifiers (i.e. either homogeneous or heterogeneous type) in either sequential (i.e. boosting) or in parallel (i.e. bagging) configurations can be used to achieve improved classification/regression performance of the target task than the performance of individual models. The key objective of ensemble method is to reduce variance and bias. Ensemble classifiers result in a model that achieves performance gains and also a model that can generalize well.

Ensemble learning is proposed to address bias-variance trade-offs in the performance of classifiers. Variance is the error caused by limitations of learning data, whereas one of the limitations of the base classifiers is bias.
The four major ensemble approaches: bagging, boosting, stacking and blending.

Bagging

Bagging, also called bootstrap aggregation, is a method for creating L base learners by corresponding M sample data generated randomly from the training dataset (by replacement). Suppose for N iterations, M random samples from training data are generated. The basic steps of the bagging method is presented in listing 7.1.

Algorithm 7.1: Algorithm: Steps in bagging method.

Input: Loading training data i.e. D=$\{x_i, y_i\}_{i=1}^{M}$
Output: Averaged prediction
1 **foreach** $i \in$*NumberOfIteration* **do**
2 | M random samples from training data D ;
3 | L base learners h_i of are created ;
4 | Train all created models h_i with i^{th} sample in parallel as models are independent ;
5 Combine the final predictions of all models by averaging predictions using eqn. 7.1

Description: The algorithm loads the training sets. For N iteration (line 1), generate M random sample from D (line 2) and create L base learners (line 3). Train all L base learners with corresponding samples in parallel (line 4). Line 5 returns the average prediction of these base learners. For L base learners, the final prediction of their ensemble is given by averaging all predictions from all L models using equation 7.1.

$$c_i = \frac{\sum_{j=1}^{L}(h_j(x) = i)}{L} \tag{7.1}$$

Where c_i is the average predicted class which is obtained by averaging the predictions of L number of models h for the i^{th} data sample, x. Some of the popular bagging algorithms are Random Forest and Bagging meta-estimators. Unlike boosting, it is noted that bagging does not consider weighting in training data.

Boosting

Boosting is an ensemble of base learners that are trained sequentially where each classifier is started with the equal weight, but after all models are trained once, a new weight is assigned for each model based on its performance. After model evaluations, larger weight is assigned for a mis-classified sample to provide greater focus in the next iteration and vice-versa. The final

model relies on a weighted averaging method.
Boosting classifiers include gradient boosting classifier, adaboost classifier, extreme gradient boosting classifier and light gradient boosting classifiers. Unlike bagging classifiers, boosting classifiers are sequential (i.e. the input of the next base learner is the output of its previous base-learners). Boosting tries to address bias, whereas variance is addressed by bagging [133]. In contrast, boosting is sensitive to over-fitting as it tries to fit the data to the model [118, 117].

Stacking

Apart from bagging and boosting, combining different base models is called stacking. Stacking, also called stacking generalization, is a hybrid of different (i.e. heterogeneous) types of base learners where a single training input is provided to those learners (level 0) and their predictions are the input to the next level base learner (meta-learner). The meta-learner (level 1) predicts the final decision. Stacking is like blending. The difference is that blending divides the data into training and validation sets, whereas stacking makes use of cross-validation for splitting dataset. Stacking and blending are hybrid methods, as these ensembles combine different base learners. Stacking (or blending) uses a meta-learner as an ensemble combiner. This type of learning is called meta-learning. For several reasons, ensembles improve performance. The steps of stacking methods are depicted in listing 7.2.

Algorithm 7.2: Algorithm: Steps in stacking method [64].

Input: Loading training data i.e. D=$\{x_i, y_i\}_{i=1}^{M}$
Output: Optimal model L
1 Partition the training set into two disjointed sets.;
2 Train several base learners L on the first set with CV=5. ;
3 Take average of the 5 models' accuracy. ;
4 Compute predictions for each base model on the second set ;
5 Use the predicted results to train the meta-learner

Description: The training set is loaded. The training sets are partitioned into two independent sets (line 1). Two or more base learners (SVM, NB, RF, LR) with CV of 5 folds are trained using the first data set (line 2). The average predictions of the five base models (line 3) are calculated. Predictions of each base learner to train a high level learner (meta-learner) (line 4) are prepared.

7.2.2. Proposed Stacking Algorithm

According to the hyper-parameter tuning result in chapter 7.1, the best performing machine learning algorithms are selected as base learners (SVM, NB and RF). During model selection in chapter 7.1, SVM outperforms the other classifiers for sentiment classification. SVM is taken as

the baseline for this experiment. The proposed stacking algorithm for sentiment classification is depicted in Figure 7.3.

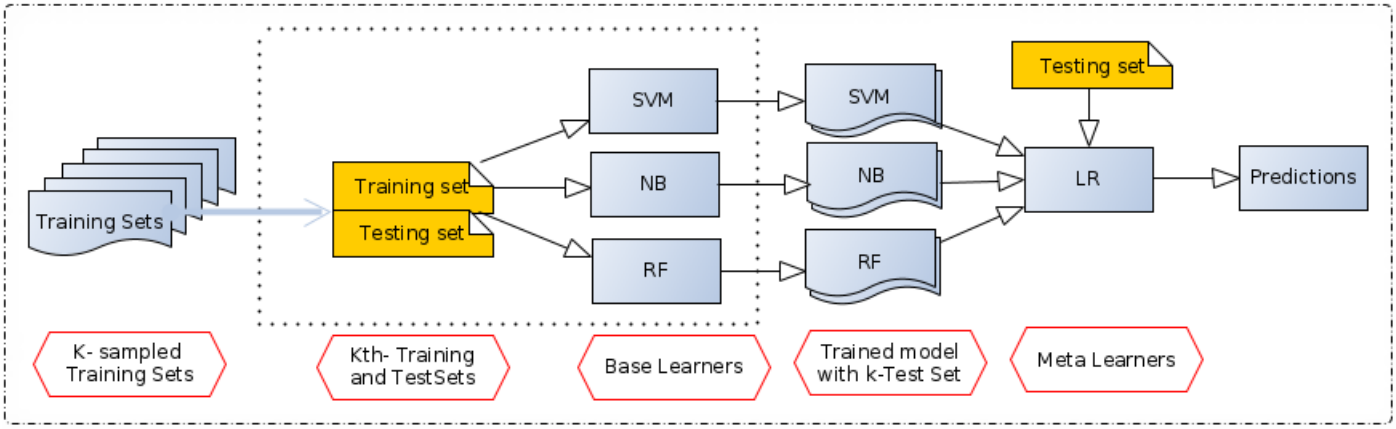


Figure 7.3.: Proposed Stacking Sentiment Classifier

If base models are weighted, the model with higher weight has a larger influence and the model with weight close to zero has least influence on the final performance of the ensemble model [64].

Imbalanced Learning : If a machine learning algorirthm is trained using imbalanced data sets, the model will tend to overfit and will have poor classification performance. This is because the model is overwhelmed by training samples with majority classes (i.e. they are over-represented, representing the highest proportion of the data sets). This leads to an increase in the rate of false negative class predictions on the samples with minority classes (i.e. under-represented samples where a class has a small proportion of samples). To reduce this problem, applying Synthetic Minority Oversampling TEchnique (SMOTE) has shown an improvement in recognizing minority classes in sentiment classification [4].

Stack classifiers with 5-fold cross-validation are proposed. In the proposed stack configuration, the best performing classifiers (i.e. SVM, RF, NB) from the previous section (i.e. Part I) are trained and their predictions are input into logistic regression for final prediction.

7.2.3. Results and Discussions

The ensemble approaches (bagging, boosting) are tested to determine to what extent they improve performance of Amharic sentiment classification. Each of these ensemble approach's result is no better than the baseline performance (i.e. the SVM base learner). In contrast, stacking with meta learner outperforms the baseline performance, as presented in Table 7.5.

Table 7.5.: Ensemble Classifier Stacking with CV of 5 folds for Amharic Sentiment Classification using Annotated User Comments Data under Category II

	5 Folds CV + SMOTE				
DataSets		SVM	NB	RF	STACKING
GCOA	Accuracy	0.86 (+/- 0.05)	0.81 (+/- 0.02)	0.84 (+/- 0.06)	0.87 (+/- 0.05)
EBC	Accuracy	0.87 (+/- 0.11)	0.79 (+/- 0.04)	0.84 (+/- 0.11)	0.88 (+/- 0.08)
PMO	Accuracy	0.89 (+/- 0.07)	0.82 (+/- 0.03)	0.87 (+/- 0.07)	0.90 (+/- 0.06)
ZEMEN	Accuracy	0.79 (+/- 0.06)	0.74 (+/- 0.04)	0.76 (+/- 0.07)	0.80 (+/- 0.06)

Discussion: With GCOA, EBC, PMO and ZEMEN comments data sets of TF-IDF character (1,7) grams feature set, the accuracy with 5CV + SMOTE is depicted in Table 7.5. In each of the four data sets, the proposed stack meta-learners outperformed the individual base learners (i.e. SVM, NB and RF). Specially on PMO data sets, base learners (SVM, NB and RF) combined with the meta-learner (i.e. LR) show performance gains with an average accuracy of 90(+/-0.06) over the base learners.

The proposed stack result in the least accuracy 80 (+/-0.06) on ZEMEN data set. One of the reasons for this is that ZEMEN data set differs from the other data sets in that it is collected from You Tube Comments. The length of the comments and the nature of the issues of the comments are another possible reason why the proposed stack is not significantly performing on this data set.

Combination of base learners using stacking followed by meta learner LR has achieved better performance over the individual learners across the four data sets. Stack meta learner with LR also performs better than SVM base learners (with accuracy of 83%) which is the baseline of this research.

7.2.4. Summary

This section presents the summary of Part II. The proposed stacking with meta learner using TF-IDF character (1, 7) grams feature sets (from feature selection in Part I) with CV of 5 folds is used to train the proposed stacking configurations of base learners (i.e. SVM, NB and RF taken from a model selection of Part I) and meta-learner (LR). The proposed stack model gains performance improvement over the base classifier (i.e. SVM in Part I) for sentiment classification of user generated text in social media.

The key contribution of this chapter, is the demonstration that stacking of base classifiers with meta learner has shown improving performance of sentiment classification as compared to stacking without meta learner (i.e. the base classifiers SVM, NB, and RF).

As stacking classifier is a combination of heterogeneous classifiers, it also performs better than a combination of homogeneous classifiers.

7.3 Part III. BERT Fine-Tuning for Amharic Sentiment Classification

Part III is organized into five subsections. The first subsection overviews BERT. The second subsection states the proposed BERT fine-tuning approach for sentiment classification in Amharic. The third subsection discusses the results of the performance of proposed approach. The fourth subsection examines the sample error cases made by the proposed approach. The last subsection presents the summary.

7.3.1. Overview

In this research, "transfer learning" is defined as a machine learning technique which makes use of pre-trained models on a large dataset and then fine-tune them for specific natural language tasks.

Transfer learning is an approach for optimizing performance on new tasks, leveraging existing knowledge in the source domain. Transfer learning is a technique for solving a new task (target) through transferring knowledge from another task (source). Transfer learning is applied only when the features learned in the first task (source domain) are general [25]. Transfer learning is also called domain adaptation. Transfer learning is a vital machine learning technique which are achieving many in transferring knowledge into different domains of computer vision and NLP [78].

In NLP stream, Transfer Learning (TL) is classified into two major categories [120]: (i) Transductive TL (transfers between from source to target domain with each of them having the same task where there is labeled data only in the source domain) and (ii) Inductive TL (transfer from source to target of different tasks and labeled data is available only in the target domain).

Since 2018, different NLP pre-trained models have emerged for transfer learning in downstream tasks of NLP with small labeled datasets. One of the core component of modern NLP is language model. Language model is a statistical tool which analyzes the patterns of human language to prediction of words or phrases.

Currently, modern pre-trained language models include Embeddings from Language Models (ELMO) [110], Universal Language Model Fine-tuning (ULMFiT) [61], Bidirectional Encoder Representations from Transformers (BERT) [37] and their variants. The above language models came up with special functionality. For example, the major feature of ULMFiT transfers learned knowledge by using fine-tuning, ELMO transfers by using feature extraction and BERT transfers by using attention extraction

In [41], authors present an overview of BERT and show how to create and fine-tune BERT for sentiment classification. The researcher is inspired by the works of [37] in which BERT pre-trained models are fine-tuned for downstream NLP tasks with insufficient labeled training data, faster computational speed, quicker development and better results.

In this research, the inductive TL is proposed to develop Amharic sentiment classification system by fine-tuning pre-trained language models (i.e. BERT). The performance of these pre-trained models on Amharic sentiment classification can be compared against a baseline of traditional machine learning algorithms [61]. Unlike ULMFiT, BERT has only two steps: pre-training using general domain corpora and fine-tuning of classifier [37].
BERT-based fine-tuning approach is presented in the subsequent sections for Amharic sentiment classification.

7.3.2. Proposed Approach

For transfer learning of BERT pre-trained models, fine-tuning the last layer is done for any of NLP tasks: named-entity classification, text classification, language inference, question and answering, sentiment classification and so BERT model for Amharic language is developed by adopting the same architectural settings.
In line with the framework of [37], the proposed BERT architecture has 3 main layers, as shown in Figure 7.4 : 1) Input layer 2) BERT Encoder and 3) Output layer

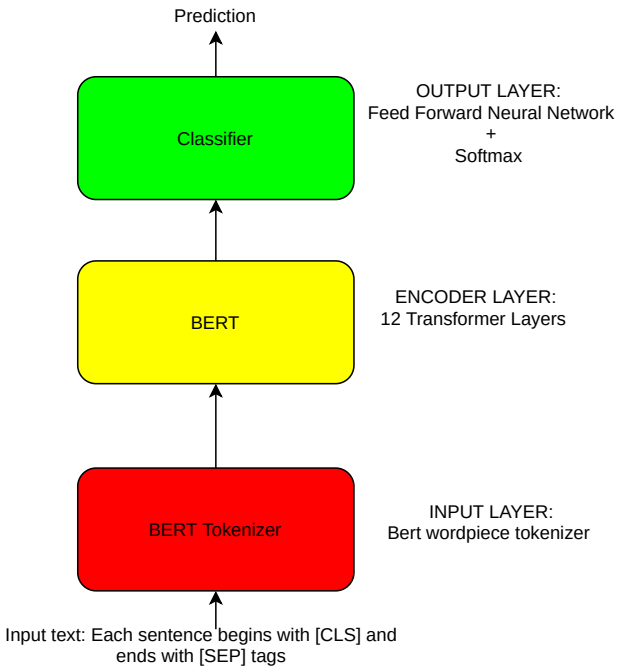


Figure 7.4.: Proposed BERT Sentiment Classifier

(1) **Input layer**: Input layer is the embedding layer which adds up position embeddings, segment embeddings and token embeddings and is followed by normalization. (2) **BERT encoder**: BERT encoder comprises 12 transformers where the very power of BERT is that

each transformer comprises attention mechanism to capture the context of a word in the input text.

For example, **"አዲስ ወደ አዲስ አበባ መጣ:: አዲስ አበባን ይወዳታል:: አዲስ ለአበባ አበባ ሰጣት:: አዲስ አዲስ መኪና ገዛ::"**/*"Addis came to Addis Ababa. He loves Ababa. Addis gave her flower. He bought a new car."*/ Words **"አዲስ"** and **"አበባ"** in the text are represented differently by the encoder using attention mechanism, as these two words have different meanings and contexts in these sentence.

Each transformer contains multi-headed attention, which has three input matrices (i.e. query, key and value) to represent an item in a sequence. Each transformer uses self-attention (where query, key and values are equal), but each of them is projected in different independent sub-spaces. Each transformer represents a word with all its contexts using multiple headed attention, which is a unique feature allowing the model to jointly capture information from different sub-spaces at different positions at a time.

The equation to compute the multi-headed attention of the i^{th} (query, value and key) is given by:

$$Attention(Q_i, V_i, K_i) = Softmax(\frac{Q_i K_i^T}{\sqrt{d}}) * V_i \tag{7.2}$$

Where Q_i, V_i and K_i are query, value and key matrices of the i^{th} attention head with d-dimension subspace projections of matrices of an input word in a sequence. (3)Output layer: Output layer is a classifier for target task. This is a fully connected layer associated with [CLS] token information. [CLS] stands for classification and is a special classification token in the last hidden state of BERT, which is used for classification task. The output layer, also called task-specific layer, is customized depending on the type of the task.

While moving from the input layer to the output layer, the model captures more specific information related to the task, whereas as one goes from the output layer to the input layer, it represents more general information independent of the task. The input and BERT encoder layers are trained with large generic corpora and then the model is fine-tuned for different downstream NLP tasks by customizing the output layer. This minimizes the need for large labeled data for developing classifiers in the target task, resulting in fewer computational resources and a shorter development time.

As Amharic is not included in the Huggingface pre-trained model library, BERT tokenizer is developed for Amharic language and pre-trained language model is developed from scratch for further fine-tuning to carry out Amharic text classification tasks (i.e. sentiment classification). The next subsection presents the steps to develop the BERT pre-trained model for Amharic language.

Amharic text BERT Tokenizer and Input Formatting

Tokenizers encode inputs to the models. BERT pre-trained models comprise their own tokenizers' library [1]. Amharic language is not included in any of BERT pre-trained models. And, Amharic BERT wordPieces tokenizer is not present in any of existing BERT tokenizer vocabulary. The first task of this research is to build BERT wordPiece tokenizer vocabulary for Amharic texts before building BERT language model for Amharic.

There are various approach to create BERT wordPiece tokenizer vocabulary for new languages (Amharic). The subword vocabulary needs to be learned and extracted.

For the sake of demonstrating how subword vocab is built, the **Byte-Pair subword embeddings (BPemb)** is used. **Byte-Pair subword embeddings (BPemb)** is a collection of pre-trained subword unit embeddings in 275 languages, based on Byte-Pair Encoding (BPE). Amharic is one of these languages. The subword vocabularies of Byte-Pair subword embeddings (BPemb) pre-trained models are available with different dimensions and sizes, which are publicly accessible in [57].

For example, if the input text is: **"ጀግኖቻችን በርቱ ምርጥ ያባገዳልጆች"** /"Be strong heros of the sons of Aba Gedda"/, then the generated wordPiece is: ['_', **'ጀ', 'ግ', 'ኖ', 'ቻችን'**, '**_በ', 'ር', 'ቱ**', '**_ም', 'ር', 'ጥ**', '**_ያ', 'ባ', 'ገ', 'ዳ', 'ል', 'ጀ', 'ች**']. This generated format does not match BERT vocab format. For example, underscore (_) is illegal to be part of BERT vocab entries. Because of this, the BPEmb wordPiece vocab format is transformed to BERT wordPiece vocab format. Below are the two rules which do the conversion.

Rule 1: If a wordPiece begins with under-score, the underscore is removed and then added into the BERT vocab.

Rule 2: If the wordPiece does not begin with under-score, ## is added to the beginning of that wordPiece and then added to the BERT vocab.

For the above example, the BPEmb wordPiece format is transformed to BERT wordPiece format as: [**# ,##ጀ ,##ግ ,##ኖ ,##ቻችን ,በ ,##ር ,##ቱ ,ም ,##ር ,##ጥ ,ያ ,##ባ ,##ገ ,##ዳ ,##ል ,##ጀ ,##ች**..

Using this method, Amharic text input to BERT model can be subword tokenized before encoding. Fifty thousand of Amharic BPemb tokens are extracted from the pre-trained tokenizer model [2], and these are converted into BERT format using the above rules. Finally, the converted BERT vocab is stored and used for building BERT language model for Amharic. As BPemb pre-trained tokenizer model is trained on large wiki text, this potentially reduces out-of-vocabulary subwords.

The other alternative is to use the **transformer python library**. A BERT wordPiece tokenizer object is created from the BERT Tokenizer class. The tokenizer object is created by specifying

[1] https://huggingface.co/transformers/

[2] https://bpemb.h-its.org/am/am.wiki.bpe.vs50000.vocab

vocab and minimum frequency parameters and, the object is trained using text corpus of the target language (i.e. Amharic). Finally, the trained tokenizer is stored for further BERT tokenization tasks. For simplicity, this research used this approach for building Amharic BERT tokenizer with vocabulary size of 30522.

Sentence Length and Attention Mask: To determine the maximum length, the frequency distribution of lengths of training samples is computed. Relying on this, the maximum length of input sequence will be 512. Truncation is performed from the beginning of sentences as Amharic verbs are usually found at the end of a sentence. Padding preserves information loss of Amharic text sequences by setting it from beginning of training samples.

BERT Model Building and Fine-Tuning

Pre-training: is a one time step (i.e. build once and use many times). BERT is pre-trained on either Masked Language Model (Masked LM) Task or Next Sentence Prediction Language model (NSP LM) using very large cross-domain corpus collections [135].

To capture the general semantic and syntactic knowledge of a language, the language pre-trained models are trained using large corpora in an unsupervised manner relying on either Masked LM or NSP LM. In Masked Language Modeling (Masked LM), the objective of the model is to guess the randomly masked token in the input text, whereas in Next Sentence Prediction (NSP), the language model is trained to predict a fraction of randomly masked tokens in one sequence of a pair of input sequences.

For example, if the input text is: **"ወይ ጥልቅ ተሀድሶ እያየን ያለው ጥልቅ ጉድጓድ እና ጥልቀት ያለው እስር ቤት ነው ።"**/"Oh, the deep reforms, that we saw, are deep wells and deep prisons."/, the masked LM will generate: **"ወይ ጥልቅ [MASK] እያየን ያለው ጥልቅ ጉድጓድ እና ጥልቀት ያለው እስር [MASK] ነው ።"**

On the other hand, if the input text is **"እስቲ ለፅድቅም ቢሆን ተናገሩ...መግስት ይሄዳል ይመጣል...ሕዝብ ሁሌም ይኖራል...ውሸት ይብቃ ትውልድ ይዳን::"**/"Speak up for righteousness ... Magistrate will come and go ... People will always exist ... Enough lies and a generation will be saved."/, the corresponding masked NSP LM can be **"እስቲ [MASK] ቢሆን ተናገሩ...መግስት ይሄዳል ይመጣል..[MASK]ሁሌም ይኖራል...ውሸት [MASK] ትውልድ ይዳን::"**.

In each of the cases, the MASKs are given as output to be learned while the other texts are fed as input to the model. The two phases of BERT (i.e. pre-training and fine-tuning) are shown in Figure 7.5.

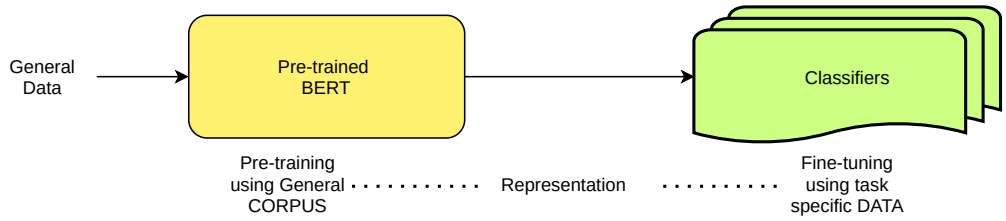


Figure 7.5.: Two Stages of BERT

Basically, there are two BERT architectures (the base vs. the large). Each of them has four fundamental parameters, which differentiate BERT architectures [113]. These include the number of hidden layers in the transformer encoder also called transformer blocks (12 vs. 24)), the number of attention heads also called self-attention (12 vs.16), the size of hidden layer in the feed-forward network, and maximum sequence length parameters, also called maximum input size (768 vs. 1024).

The BERT base architecture model is selected for training Amharic language model from scratch. However, the current pre-trained BERT model is widely available for dominant languages. Less-resourced language like Amharic has no pre-trained BERT in the BERT pre-trained model hugging-face library. A small-scale pre-trained BERT -base model is built for Amharic for further tasks to be used for fine-tuning to perform sentiment classification. The details of hyper-parameters are shown in Table 7.6.

Fine-Tuning: this phase minimizes the need for resources: large labeled datasets and high computational resources to perform NLP tasks (i.e. sentiment classification). It is investigated the extent to which fine-tuning the pre-trained model accurately performed sentiment analysis of Amharic in situations where there are limited resources (i.e. scarcity of annotated sentiment corpora). The pre-trained BERT model is fine-tuned for downstream NLP tasks: text classification, language understanding and generation, cross-lingual inference and so on.

For fine-tuning of BERT for text classification, final hidden state h of first token [CLS] is taken as representation of the complete input sequence.

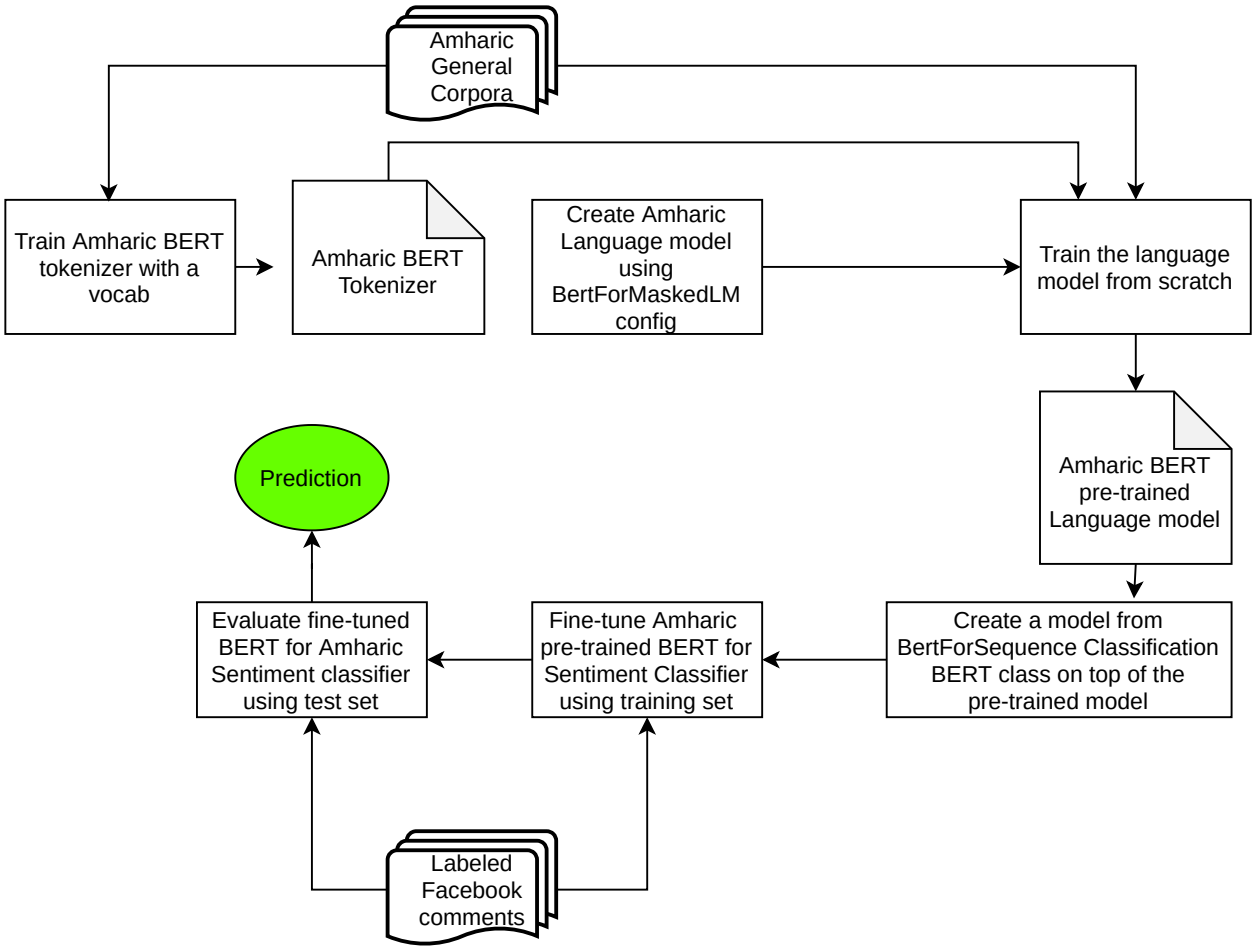


Figure 7.6.: Flow chart of Proposed BERT Sentiment Classifier

Algorithm 7.3: Algorithm of the Proposed BERT Sentiment Classifier

Input: Load Amharic general corpus collections
Input: Load labeled comments i.e. D=$\{x_i, y_i\}_{i=1}^{N}$
Output: Fine-tuned BERT for Amharic Sentiment Classifier
`/* Train Amharic BERT tokenizer with a vocab */`
1 Train Amharic Byte-level BPE (BBPE) Tokenizer on the Amharic general corpora by using the Tokenizers library (Hugging Face) ;
2 Store the generated Amharic BERT tokenizer ;
3 Load Amharic BERT tokenizer ;
`/* Create Amharic Language model using BertForMaskedLM config */`
4 Set Language model training parameters ;
5 Train the language model from scratch using Amharic general corpora (split into training and test sets);
6 Store the trained Amharic BERT pre-trained Language model ;
7 Load the trained Amharic BERT pre-trained Language model ;
`/* Fine-tuning for sentiment classification */`
8 Split labeled sentiment corpora into training and testing sets ;
9 Load the Amharic BERT pre-trained model and Amharic BERT tokenizer files;
10 Create a model from BertForSequence Classification BERT class on top of the pre-trained model ;
11 **foreach** $i \in TrainSize$ **do**
    `/* Amharic Sentiment Classifier Training Loop: */`
12 Unpack the data inputs and labels ;
13 Load data onto the GPU for acceleration ;
14 Clear out the gradients calculated in the previous pass. ;
15 Forward pass (feed input data through the network) ;
16 Backward pass (backpropagation) ;
17 Tell the network to update parameters with optimizer.step() ;
18 Track variables for monitoring progress ;
19 **foreach** $i \in TestSize$ **do**
    `/* Amharic Sentiment Classifier Evaluation loop: */`
20 Unpack the data inputs and labels;
21 Load data onto the GPU for acceleration;
22 Forward pass (feed input data through the network);
23 Compute loss on the validation data ;
24 Track variables for monitoring progress ;

Description: The listing in 7.3 loads Amharic general corpora and labeled social media comments (GCOA, PMO, EBC and ZEMEN). Line 1 trains Amharic BPE tokenizer using the general Amharic corpora. Line 2 stores the trained Amharic BERT tokenizer and then line 3 loads it.

Line 4 sets the language model training parameters by using BertForMaskedLM config. Line 5 trains the language model using the Amharic general corpora. Line 6 stores the trained BERT model for Amharic language and loads it (line 7). From line 8-18, fine-tuning Amharic BERT

for sentiment classification is performed. Line 8 splits the labeled comments into training set (of TrainSize) and testing set (of TestSize). Line 9 loads the Amharic BERT pre-trained language model and Amharic BERT tokenizer files.
Line 10 creates a model from BertForSequence Classification BERT class on top of the pre-trained model. For each training sample in training set, lines 11 to 18 are executed. Line 12 unpacks the input features and labels into tensor format. Line 13 loads it to GPU for acceleration. Line 14 clears the gradient computation history. Line 15 trains the network by feeding in the input data (i.e forward pass). Line 16 performs backward pass. Line 17 updates the network parameters. Line 18 tracks the progress of the network model by monitoring the variables.
For each testing sample in testing set (of TestSize), lines 19-24 are executed. Line 20 unpacks the data into input features and labels. Line 21 loads data into GPU for the sake of acceleration. Line 22 feeds input data into the network i.e. forward pass. Line 23 computes the loss on testing datasets. Line 24 tracks variables for monitoring progress.
To predict probability the class label c, softmax function is used,

$$p(c|h) = softmax(Wh), \tag{7.3}$$

where W is task-specific parameter matrix, h is the final hidden state taken from the first token [CLS] representing the entire sequence. All parameters of BERT are fine-tuned jointly with W to maximize the log-prob of a correct label of class c [135].
The proposed BERT transformer model architecture has 201 different named parameters. To mention a few in embedding layer, word_embeddings.weight, position_embeddings.weight, token_type_embeddings.weight and LayerNorm.bias which are implemented using python pytorch library.

7.3.3. Results and Discussions

Two experiments have been carried out. The detailed discussions of these are presented as follows.
Experiment I Building Small-scale BERT Pre-trained Model: In this case, the BERT Base architecture configuration template is adopted using "BertForMaskedLM" as suggested for developing pre-trained language models for new language.
A "small-scale" Amharic language model is built with 84 million parameters = (12 layers, 768 hidden size, 12 attention heads) which is the same number of layers and attention heads as BERT (small). The configuration of the proposed pre-trained architectural configurations is set up with the above settings and a vocabulary of size 30522. The parameter value settings for the proposed BERT-base architecture are summarized in Table 7.6.

Table 7.6.: The Parameter Settings of the Proposed BERT base Architecture System

Hyper-parameter	Value
Attention heads	12
vocabulary size	30522
Batch size	4
Epochs	5
Gradient accumulation steps	16
Hidden size	768
Hidden layers	12
Learning rate	0.00005
Maximum sequence length	512
Parameters	84M

One of the aims of this approach is to fine-tune one of downstream tasks of Amharic (i.e. sentiment classification under insufficient annotated sentiment corpora). For pre-training experiment, google colab TPU of 25GB RAM and 99GB storage is used, as allocated for google Gmail account. A general domain text collection of size 904MB described under category-I of chapter 4 is used. The final pre-trained model can be publicly available for doing other downstream tasks of Amharic. The data is partitioned into training and validation data with 80:20 ratio. Finally, the pre-trained model is stored for further use.

Experiment II Fine-tuning LM model: For this experiment, the labeled comments data sets of category II are used for fine-tuning Amharic sentiment classification. In this experiment, the BertForSequenceClassification model is used to perform Amharic sentiment classification. After fine-tuning of BERT for Amharic sentiment classification, the performance of the BERT classifier is evaluated using Matthews Correlation coefficient (MCC), Accuracy, Precision, Recall and F1-Score on the four data sets (PMO, ZEMEN, EBC and GCOA) as depicted in Table 7.7.

Table 7.7.: Performance of BERT fine-tuned Amharic Sentiment Classifier using Annotated User Comments Data under Category II

DataSets		Precision	Recall	F1-score	Accuracy	MCC
GCOA	Negative(0)	0.95	0.97	0.96	0.95	0.899
	Postive(1)	0.96	0.92	0.94		
EBC	Negative(0)	0.94	0.98	0.96	0.94	0.851
	Postive(1)	0.94	0.85	0.89		
PMO	Negative(0)	0.96	0.97	0.96	0.95	0.886
	Postive(1)	0.94	0.9	0.92		
ZEMEN	Negative(0)	0.93	0.93	0.93	0.94	0.885
	Postive(1)	0.95	0.95	0.95		

To evaluate the BERT pre-trained model for transfer learning, Matthews correlation coefficient (MCC) is used. MCC is used to evaluate the quality of models trained for binary and multi-class classification. This metric is used as a balanced-measure of models' performance trained on samples with different class sizes (imbalanced classes).

MCC values range from -1 (worst prediction) to +1 (perfect prediction). The MCC value of GCOA is the highest of all other data sets. This shows that the GCOA data consists of highly balanced data sets (i.e. No.POS = 1728 and No.NEG = 1143 as stated in Table 4.2). In general, MCC values for all the data sets are balanced as they is greater than 0.85.

With respect to the F1-score measure, the classifier has the best of all F1-scores (i.e. 96 and 94) for negative and positive classes, respectively. The accuracy of BERT classifier on GCOA was the same (i.e. 0.95) as PMO data set. This is because the two data sets contain similar (overlapping or related) issues expressing the government activities and national media sectors. The result is also interpreted using confusion matrix which is depicted in Table 7.8.

Table 7.8.: The Confusion matrix of BERT fine-tuned for Amharic Sentiment Classification using PMO Data Sets

Target Class	0	1
0	873	24
1	14	380

As presented in Table 7.8, only 24 of the negative test samples are wrongly classified as positive category 14 of positive samples are incorrectly classified as negative. This is a dramatic

improvement on performance of Amharic sentiment classification when it is compared to the performance of the baseline approaches shown in earlier chapters (i.e. SVM baseline).

7.3.4. Error Analysis

The samples which are predicted by the BERT model are traced. For error analysis, a few wrongly and correctly predicted samples are shown in Table 7.9 below.

Table 7.9.: Sample of Predicted Samples by the trained BERT model

Texts (in Am-En)	Class_Org.	Class_Pred.
"ህይወት ንብረት ከጠፋ በሁላ መግለጫ መስጠቱ ምንም አይሰራም"/ Mourning press conference has nothing after life and property is lost/	-	+
"ጥሩ ውሳኔ የተወሰነ"/ It is a good decision/	+	-
"ጠቅላይ ሚንስትር አብይ ለህዝቡ ክብር የሰጠ የሀገሪቱ የመጀመሪያ መሪ"/The current PM is the first leader who has given respect to his people./	+	-
"በዴሞክራሲያዊ አሰራር የብዙሀን ድምፆ እየተከበረ አይደለም"/Democratic leadership and the voice of the people is not respected./	+	-
"ጀግኖቻችን በርቱ ምርጥ ያባገዳ ልጆች"/ Keep on patroitic son of Aba Gedda/	+	+
"እባብን መቸም አምነህ በጉያህ አትይዘዉም አይታመንምና"/ how you trust the snake in your pocket/	-	-

Discussion: The sample in the first row contains words: **ምንም/nothing/, አይሰራም/not working/** and negative opinion word i.e. **ከጠፋ/lost/**. This sample is wrongly predicted as positive. From the lexical analysis, the two negation words cancel each other out and what remains is a negative opinion word, i.e. loss of lives. The cumulative sentiment is negative. The reason for this error might be the BERT model might lack sufficient labeled samples for fine-tuning the pre-trained model.

The second, third and fourth rows contain samples which are wrongly predicted as negative. From the terms in the third sample, the positive opinion words are **ክብር/respect/, መሪ/leader/**. Therefore, relying on these positive words, the cumulative sentiment should be positive. The cause of this error by BERT model might be again due lack of sufficient labeled samples for fine-tuning the model or the pre-trained BERT model for adequately representing the language, due to lack of coverage of the language corpora during pre-training. Therefore, to enhance its prediction performance, extending the pre-training data for building the BERT model is needed. On the other hand, the two samples in fifth and sixth rows are predicted correctly by BERT

pre-trained model. The sixth sample is an ironic sentence. It is not directly interpreted by lexical analysis. But, the BERT model captures such an ironic sample, which is difficult to detect using conventional approaches. Though the labeled data is small, the language might be represented and captured the different contexts in the pre-trained model of Amharic. The target labeled data sets for fine-tuning the pre-trained BERT model is fundamental to successfully handle the downstream tasks of NLP application.

7.3.5. Summary

This subsection presents a summary of part III. As Amharic is a less-resourced language, lack of sufficient labeled corpora makes it challenging to carry out sentiment classification using machine learning. The researcher is inspired by the work of [41] to create and fine-tune BERT pre-trained model for sentiment classification under insufficient labeled corpora, less computational resource, less development time and in a cost-effective way as BERT makes a dramatic advancement of performance to carry out downstream tasks of NLP.

To apply sentiment classification tasks for Amharic user generated comments, first BERT is developed from scratch for Amharic language using general corpora and second, BERT pre-trained model are fine-tuned using labeled comments. The pre-trained Amharic model is stored in a file for further use to other down stream tasks of Amharic, apart from sentiment classification. The proposed BERT outperforms the other approaches of Amharic sentiment classification (SVM with tuned hyper-parameters is used as a baseline).

The key contributions of the proposed approach are: (i) the development of the first BERT tokenizer and BERT model for Amharic language and (ii) the fine-tuning of the pre-trained BERT model for Amharic sentiment classification, (iii) the performance of fine-tuned BERT as a great sign that the language model can be used for other downstream tasks of the Amharic language, and (iv) a notable performance improvement on sentiment classification of Amharic language when there is a lack of labeled sentiment corpora as compared with performance of baseline models (SVM).

Because of limited computational resource like GPU, the pre-trained BERT for Amharic language model is quite small and needs to be extended achieve performance gains in different tasks of Amharic language.

8. Hybrid Model to Amharic Sentiment Classification

This chapter presents the hybrid model for Amharic sentiment classification. This approach tests three hybrid building strategies (voting, averaging and blending) for aiming to gain performance improvements of Amharic sentiment classification on social media texts. This hybrid model of Amharic sentiment classification combines heterogeneous sentiment classification methods, including lexicon feature, supervised, stacked ensemble and transfer learning.

8.1 Overview

The applications of hybrid and ensemble methodologies in the areas of soft computing and intelligent system are becoming more popular and visible. Ensemble approach combines multiple classifiers (of homogeneous type) whereas hybrid model combines multiple classifiers (of heterogeneous type) [66].

The purpose of combining different classifiers is. (i) to increase performance in terms of accuracy and generalization, especially for high dimensional, highly uncertain and complex classification problems, (ii) to improve quality of performance by combining base learners of complementary nature, (iii) to achieve more intelligent and more accurate solutions and (iv) to achieve improved optimization on various domains using empirical data [66, 112, 153]. In general, it has been statistically proved that classifiers make independent errors [112]. The combination of classifiers can potentially boost performance of the system.

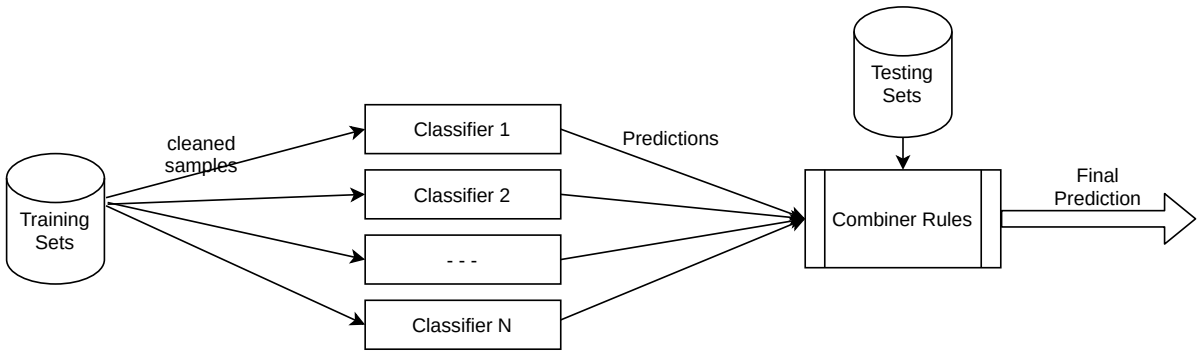
There is a consensus that the diversity of individual classifiers are categorized into different levels: (1) no more than one classifier is wrong for classifying one pattern, (2) the majority of the classifiers always classify correctly, (3) at least one classifier is correct for classifying each pattern, and (4) all classifiers incorrectly detect some patterns. The choice of fusing methods depends on the knowledge of the designers about the diversity [112].

To leverage the strength of individual classifiers, a high quality intelligent classification system is designed in several ways: **(i) High Intelligent System (HIS):** the combination of heterogeneous classifiers that covers all phases of the system to produce a blended and more effective system. **(ii) Information Fusion (IF):** combines different information sources to add new features or properties that enable effectively solving the problem. The new feature

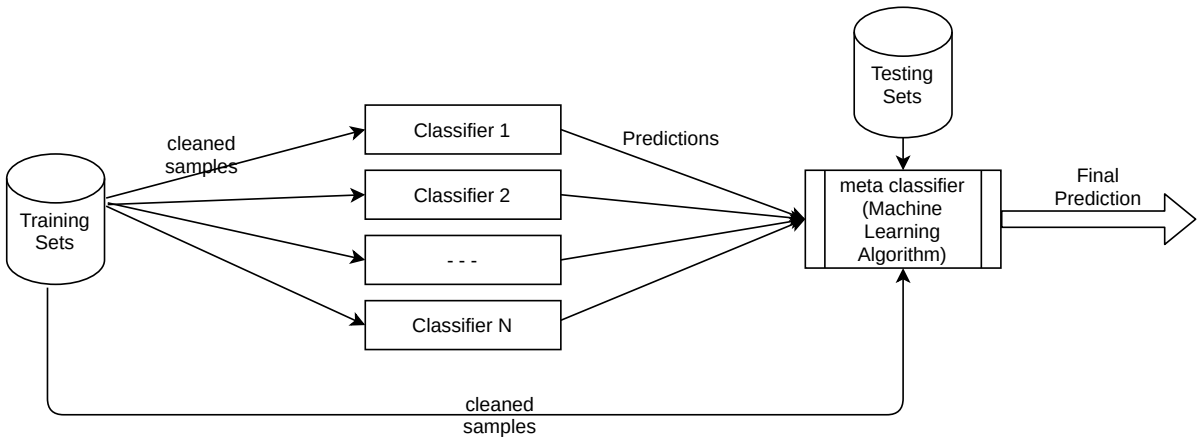
or information source is the output of another classifier or computational process. **(iii) Multi Classifier Systems (MCS):** combines heterogeneous or homogeneous classifiers to produce the ultimate solution, which is also a subcategory of the HIS [112].

8.2 Proposed Hybrid Architectures

The proposed hybrid model is categorized under MCS because it combines different sentiment classification approaches which are designed in two ways relying on:
(i) the fusion of class labels of the constituent base classifiers (as shown in Figure 8.1(a)).



(a) Hybrid Model using combining rules as fusing target class predictions relying on either voting or weighted averaging



(b) Hybrid Model using meta classifier for learning and predict target class

Figure 8.1.: Proposed Hybrid Framework Sentiment Classifier Architectures

The information generated by base classifiers is combined in either of the following approaches:
A. Voting approach: In this approach, the most frequently predicted target by those individual classifiers is selected. Voting is done by taking class c_i which has several predictors N. For sample x, class i is assigned if class i is predicted the greatest number of times. Mathematically,

$$c_i = mode\{h_1(x); h_2(x); ...; h_N(x)\} \tag{8.1}$$

where c_i is the predicted class, mode is statistical function that returns the most frequent prediction by classifiers $h_1, h_2, ...h_N$ for sample x. If classifiers are given different weights in their prediction, different weights $\epsilon(0, 1]$ are set. Mathematically speaking,

$$c_i = argmax_i\{w_j \cdot h_j(x)\}_{j=1}^{N} \tag{8.2}$$

where w_j is the weight of j^{th} classifier h for predicting class i. For example, for binary classification with class i$\epsilon\{0, 1\}$, let weight$\epsilon 0.6, 0.2, 0.2$ for three classifiers h1,h2 and h3. $c_i = argmax_i(0.2 \cdot i_0, 0.2 \cdot i_0, 0.6 \cdot i_1)$ where i_0 and i_1 are predictions of classifiers.

B. Weighted Averaging: In this method, the individual classifier is assigned a particular weight, then final predicted class is obtained as the sum of the product of the individual classifier's predicted value with its corresponding assigned weight. Mathematically, similar to maximum weighted voting, averaging weighted voting is given as,

$$c_i = \frac{\sum_{j=1}^{N} w_j \cdot h_j(x)}{N} \tag{8.3}$$

where w_j is the weight of j^{th} classifier h for predicting class i. For example, for binary classification with class i$\epsilon\{0, 1\}$, let weight$\epsilon 0.6, 0.2, 0.2$ for three classifiers h1,h2 and h3. Then the average weighting is given as follows. $c_i = (0.2 \cdot i_0 + 0.2 \cdot i_0 + 0.6 \cdot i_1)/3$ where i_0 and i_1 are predictions of classifiers.

ii. Fusion using Trainable learner: In this architecture, the combiner is an other classifier which can be trained using the information (data) generated by the individual classifiers (as shown in Figure 8.1(b)). Blending used as combiner where the predictions of the individual base classifier and the original training set are used to train meta learner and the meta learner is tested using the prediction of test set and original test by base learners. The above-mentioned three hybrid models are depicted in context diagram shown in Figure 8.2.

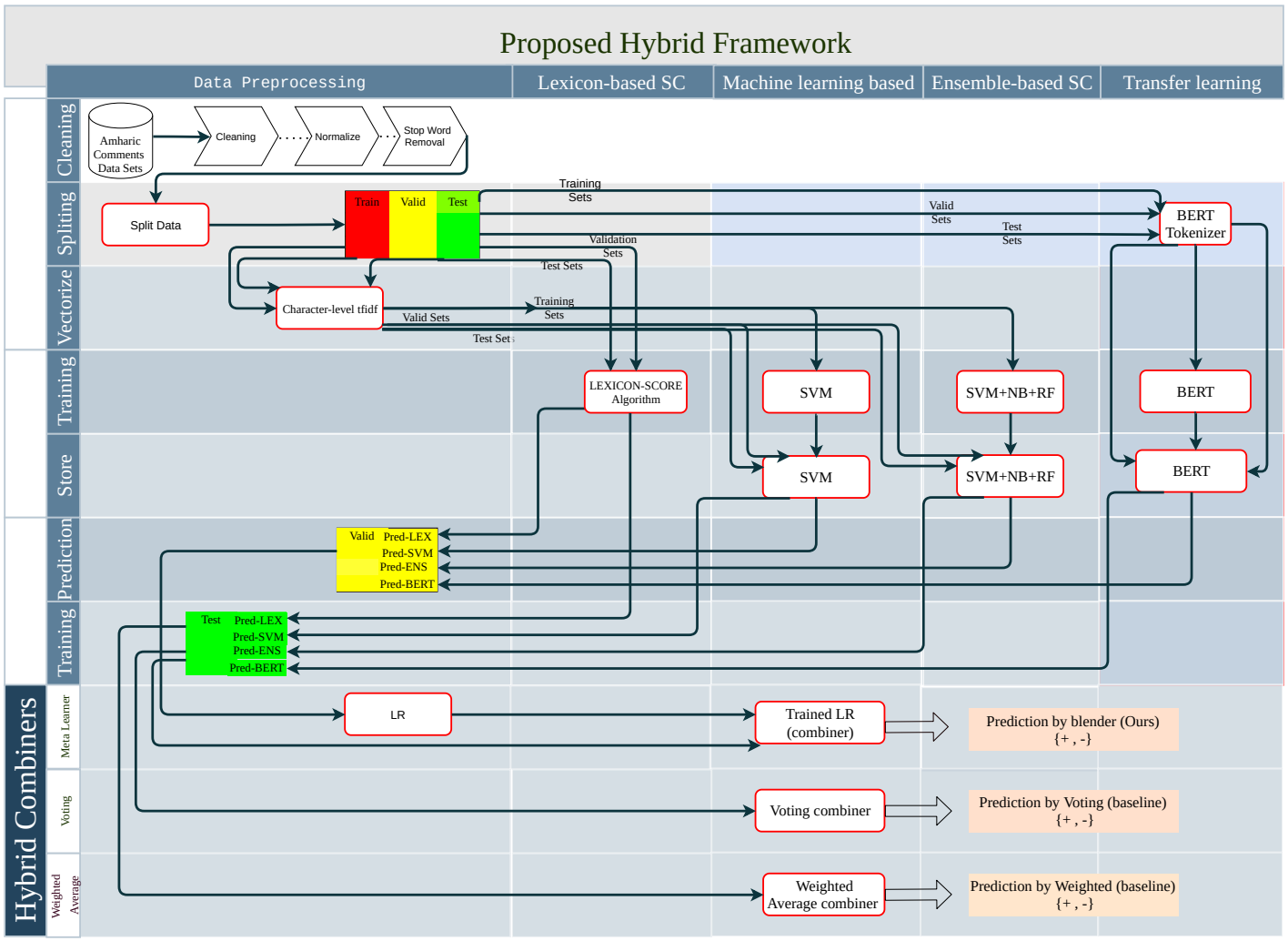


Figure 8.2.: Proposed Hybrid Framework Sentiment Classifier: The data sets consist of training set(in red), validation data(in yellow) and test set(in green), the merged validation set and its prediction(in yellow) , the merged test set and its prediction(in green)

The proposed hybrid model comprises several stages: preprocessing, data partitioning into training set, validation set and test set in a ratio of 80:10:10. The base classifiers are trained using training sets, then the predicted targets of validation set and original validation sets are concatenated and fed as input to the fusion or meta-learner layer. Lastly, the proposed model is tested using the test set. The flow chart in Figure 8.3 shows detailed procedure for the hybrid model.

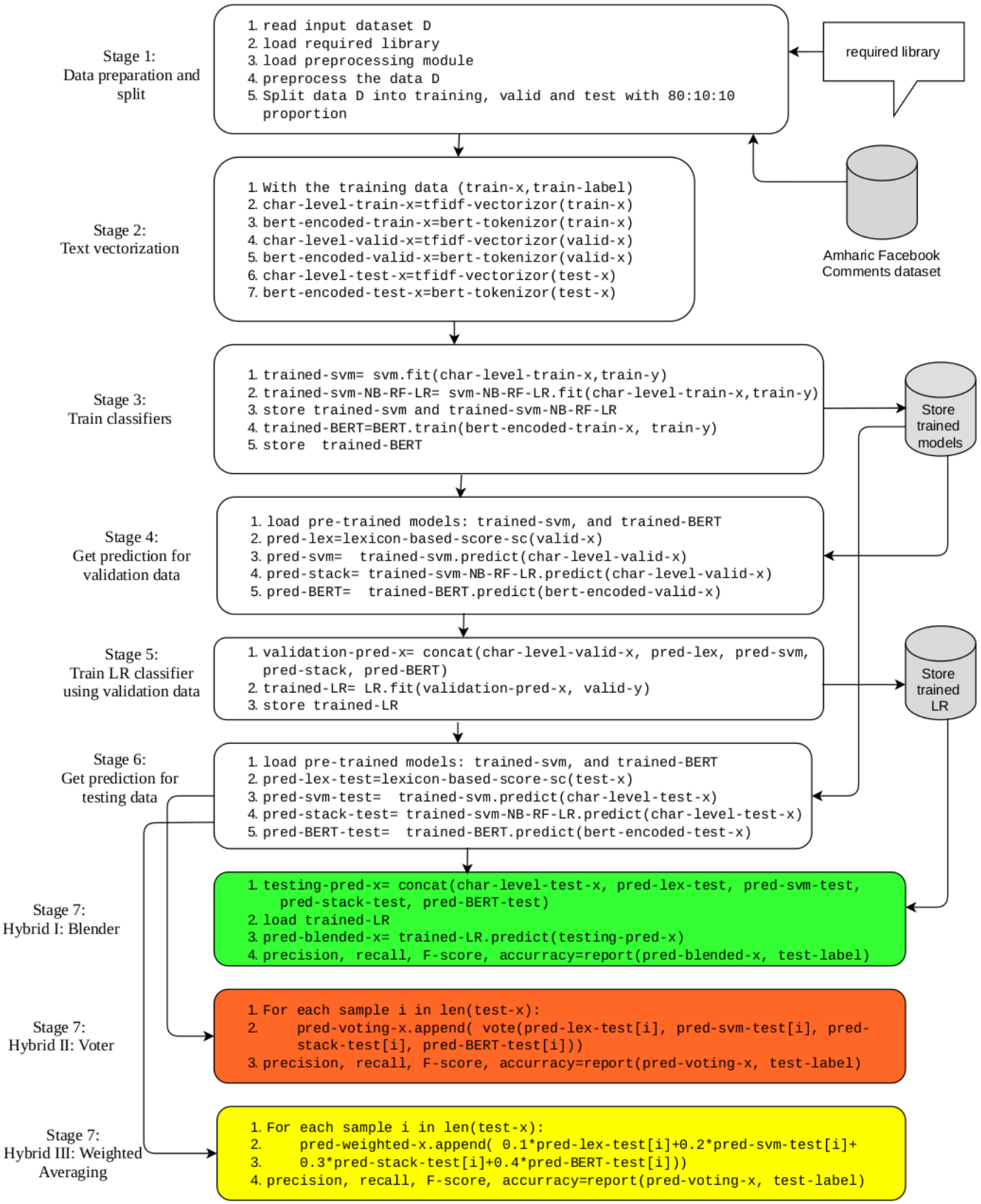


Figure 8.3.: Flow Chart for the Proposed Hybrid Sentiment Classifiers

Description: The algorithm in listing 8.3 represents the specification of the hybrid model. The rectangular boxes represents the seven stages for each of the three hybrid models (blender, voter and weighted averaging) shown in the listings. The first six stages are common to each of the three hybrid models. These stages are briefly described as follows.

Stage 1: Data preparation and splitting: In this stage, lines(1-5), the dataset is read, and the python library and preprocessing interface are loaded, the dataset is preprocessed and finally the preprocessed dataset is partitioned into training, validation and training set with ratio of 80:10:10.

Stage 2: Text vectorization stage: From lines (1-7), transform the training data, validation data, testing data of the prevous stage using TF-IDF character (1, 7) grams vectorizer and Amharic BERT tokenizer.

Stage 3: Training classifier stage: From lines (1-5), in this stage, SVM classifier, Ensemble stack classifier (SVM-NB-RF-LR) using training set with TF-IDF feature set and fine-tune BERT pre-trained model using training set with BERT wordPiece embeddings feature vectors. Once these classifiers are trained, they are stored in file for use of in further stages.

Stage 4; Get Prediction for validation data stage: In this stage from lines (1-5), loads SVM, ensemble stack and fine-tuned BERT pre-trained classifiers from files. Using validation sets of stage 2 (i.e. text vectorization), the sentiment lexicon features of the validation set are also obtained. These transformed validation predictions are obtained by using corresponding models.

Stage 5: Train LR classifier stage: From lines(1-3), the validation set predictions obtained from previous stages are concatenated together. Using these concatenated validation set, the LR classifier is trained and the trained LR classifier is stored in file to use it in the next stages of the hybrid model.

Stage 6: Get prediction for testing data. Lines (1-3) using the pre-trained model in stage 4, the predictions for test data are obtained. That is, the predictions of test set by SVM, ensemble stack and fine-tuned BERT are stored for further use. The stored prediction test set uses either of the three hybrid models in the last stage (i.e. stage 7).

Stage 7: Final prediction stages: This stage has three options (i.e. Hybrid -I, Hybrid -II and Hybrid -III) to get the final prediction for the stored test set predictions obtained in stage 6. The first option is hybrid-I (i.e. Blender) (lines 1-4). In this option, the stored test set predictions in stage 6 are concatenated. This concatenated set is given to the LR trained model (in stage 5) and corresponding prediction is obtained. This prediction is the final prediction for the test set using hybrid I. The second option is hybrid-II (voter). Using again test set predictions sets in stage 6, the voter produces the corresponding predictions sets. This is the final prediction set for this approach (i.e. hybrid II). The third option is hybrid III (average weighting). Using average weighting, the final predictions are obtained for the test sets predictions stored in stage 6.

8.3 Results and Discussion

Data Preparation:The proposed hybrid model is comprehensively evaluated using the four sentiment labeled data sets (category II) from 4.1.1. Each is divided into training, validation, and test sets with a ratio of 80:10:10. The top classifiers from the previous stages (i.e. tuned SVM, stacked, BERT) are trained and the models are stored in a file for further steps. The proposed hybrid approaches are tested.

Discussion: The three hybrid models are tested and the results of the approaches are measured with evaluation metrics accuracy, precision, recall and F-score which are presented in Table 8.1 and 8.2.

Table 8.1.: Average Accuracy and Average MCC Scores of Hybrid Models to Sentiment Classifier across the four Data Sets (category II)

Classifier	Voting Approach	Weighted Averaging	Blending Approach
Average Accuracy	0.75	0.92	0.94
Average MCC	0.49	0.83	0.86

Discussions: Table 8.1 depicts the performance of the three hybrid models in terms of accuracy and MCC score using all four category II datasets. The blending approach outperforms the other ensemble approaches with a mean accuracy of 94% and a mean MCC score of 0.86. This shows that blending combiner can learn the best meta features to distinguish sentiment classes of Amharic texts.

The other approaches (Voting and Weighted averaging) achieve the mean accuracies of (75% and 92%), respectively. The voting combiner achieves the least accuracy as it depends only on the frequencies of the predicted class. In addition, voting has also the least MCC value. This shows that voting has built weak correlation between classes and features, which results in low quality binary sentiment classifier.

Table 8.2.: The mean Score of Precision, Recall and F-score of Hybrid Models Sentiment Classifier using all four datasets.

Metrics	Voting Approach		Weighted Averaging		Blending Approach	
	Negative(0)	Positive(1)	Negative(0)	Positive(1)	Negative(0)	Positive(1)
Precision	0.73	0.84	0.93	0.90	0.93	0.94
Recall	0.95	0.47	0.93	0.90	0.96	0.90
F-score	0.82	0.58	0.93	0.90	0.95	0.91

Discussion: As depicted in Table 8.2,the precision (0.73) for voting is the least for negative classes. Similarly, voting has a poor score for positive classes in both recall (0.47) and F-score(0.58).
The reason is that voting only depends on the frequency of the prediction of the base classifiers. If the predictions of the individual classifiers have different classes, the voting combiner has a greater chance of wrong prediction. The blender combiner has achieved slightly better F-score in both negative (0.95) and positive (0.91), compared to the weighted averaging combiner.
The precision, recall and F-score of weighted averaging have achieved the same mean score of 0.93 for negative classes and 0.90 for positive classes. Based on these metrics, the blender approach performs the best as compared to the other approaches.

8.4 Summary

This part presents the proposed hybrid framework for Amharic sentiment classification. The hybrid model is a combinations of the different best classification methods selected from the previous chapters. This include SVM with hyper-parameter tuned, ensemble stack learner method and BERT fine-tuned classifier. The result has shown that the hybrid model outperforms the individual classifiers.
The experimental results showed that the hybrid model can be combined using various fusion strategies: majority voting, weighted voting and blending learner. The blending combiner has achieved better performance than the other hybrid combinations. Unlike the other combiners, the blender is a machine learning algorithm which can learn the meta-features of the base learners to distinguish sentiment classes.
Errors tracking also shows that some samples wrongly predicted by lexicon-based classifier are correctly classified by the SVM classifier. If SVM wrongly predicts a sample, the BERT fine-tuned classifier correctly predicts it. This is logically consistent because BERT has a better capability for capturing different contexts of the same word depending on its usage.

9. Model Evaluation and Prototype

In machine learning, hypothesis represents a candidate model which approximates a target function mapping input samples to outputs. In the literature [119, 26], the term "hypothesis" is interchangeably used with "model". Notation for a single hypothesis is represented by h and a set of possible hypothesis spaces is represented by H. The process of selecting a hypothesis that best approximates target function is challenging. Hypothesis explains available evidence (training data) and it is falsifiable where its performance is estimated and compared to its baseline to evaluate the extent it can be used in new situations (predicts the future observations).
This chapter presents the evaluation of the different proposed approaches in the experimental setups of earlier chapters. The Research Questions (RQ) (1-6) are answered by testing the corresponding hypothesis. This chapter is organized into six sections. The first section overviews the hypothesis testing tools. The second section selects the testing method for proving the formulated hypotheses tests. The third section discusses the results of hypotheses tests. The fourth section presents the prototype of Amharic sentiment classification approaches in social media and the fifth section reports the expert evaluation of the prototype. The last section presents the summary.

9.1 Overview

Amharic sentiment classification systems are built by applying different approaches such as lexicon-based sentiment classifier, machine-learning-based sentiment classifier, hybrid-Negation Handling, BERT fine-tuned pre-trained model for sentiment classifier and ensemble learner. The performance of these proposed classifiers are compared and evaluated by testing corresponding hypotheses. During hypothesis testing, there are two types of errors: (1) wrongly rejection of a true null hypothesis (type I error) and (2) wrongly accepting the null hypothesis (type II error).
For statistical comparison of two classifiers on a single data set and classifiers running only once, the McNemar's test is recommended hypotheses testing tool. The McNemar's test, also called "within-subjects chi-squared test, is a non-parametric statistical test for paired comparison of performance of two learning classifiers (i.e. comparing prediction of two models to each other using a 2X2 contingency table or a 2X2 confusion matrix). Alternatively, if two classifiers can run repeated number of times (e.g. CV of 10 folds), 5x2CV t-test is recommended [119].

Figure 9.1 below shows the flow chart depicting the choice of the type of hypothesis testing which is used for statistical comparison of two or more classifiers on a single or multiple data set(s).

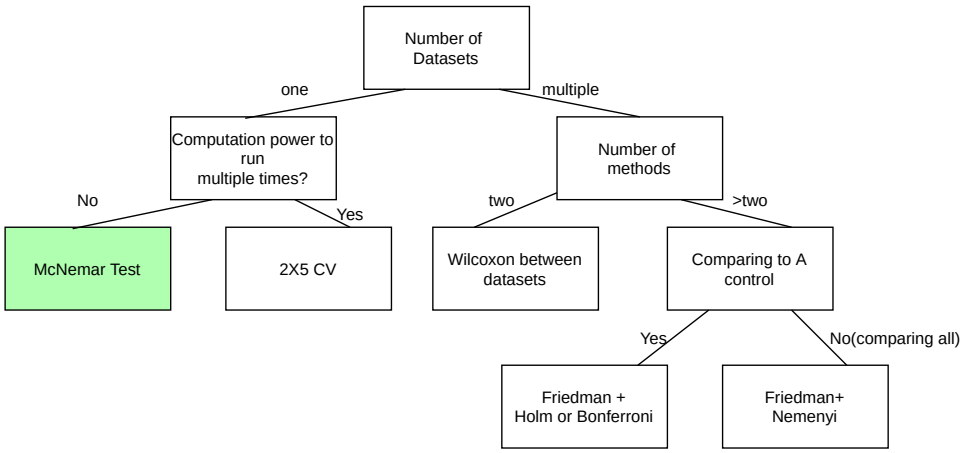


Figure 9.1.: Types of Hypothesis Testing [119]

Description: According to the flow chart depicted in Figure 9.1, the shaded (McNemar's test) is the proposed statistical hypothesis test suitable for the problem settings.

9.2 Proposed Hypothesis Tests

For this research, McNemar's test [13, 118] is proposed, for comparing models, which are trained using data sets sampled only once. McNemar's test is a statistical test that is suitable for comparing classifiers as it is proved to have minimal type-I errors [39]. The most widely used standard significance level (i.e. α) is 0.05. If McNemar's test yields probability p-value less than α, the null hypothesis is rejected (i.e. the alternative hypothesis is accepted: there is a significant difference between these two models) [118].
McNemar's hypothesis test is the only test with acceptable type-I error. McNemar's test is used for algorithms that can be executed only once [39]. That is, it is used to compare algorithms running only once and tested using the same test sets. For algorithms (e.g. deep learning) running for days, McNemar's test is the best choice, as deep learning algorithms are computationally expensive to use cross-validation.

9.3 Evaluation through Testing Hypothesis

First, the factors which are given as treatments are identified to see the extent to which they improve the performance of Amharic sentiment classification system. Treatments are usually called independent variables and their effect is measured as dependent variables. For this study, independent variables include generated Amharic sentiment lexicon features, training/testing

text data features and machine learning algorithms whereas the dependent variable is the performance of Amharic sentiment classification system.
Thus, each of the research questions formulated in chapter 1 is mapped into testable hypotheses.

Table 9.1.: MCNemar's Tests of the Formulated Hypotheses using Independently Sampled Test Set of 1954 Samples of GCOA under Category II

RQ[1]	Hypo.	Treatments	Proposed	Baseline	p-Value	Chi-Sq.	Acc.	Null-Hypo.
RQ1	H1	SOCAL-Lexicon	SVM	SVM-default+Manual	0.000	23.68	59	Rejected
		SWN-Lexicon	SVM	SVM-default+Manual	0.001	11.36	63	Rejected
RQ2	H2	PPMI-Lexicon	SVM	SVM-default+Manual	0.319	0.99	51	Accepted
RQ3	H3	Tuned-Feature	SVM	SVM-UnTuned-Feature	0.541	0.372	89.0	Accepted
RQ4	H4	Tuned-Learner	SVM	SVM-UnTuned-Feature	0.00012	14.67	89.0	Rejected
RQ5	H5	Tuned-BERT	BERT	SVM-UnTuned-Feature	0.0006	11.73	95.0	Rejected
RQ6	H6	Stack-Ensemble	SVMN-BRF	SVM-UnTuned-Feature	0.005	0.45	86.0	Rejected

Discussion: For research question RQ1, the independent variables include generated sentiment lexicons from English lexicon and the dependent variable is Amharic sentiment classification system. To test the effect of treatment on the independent variable on the dependent variable, that is, on extent to which it changes/improves the performance of Amharic sentiment classification.
Corresponding research hypotheses are formulated.

- Null Hypothesis H0: Amharic sentiment lexicon(s) generated from English sentiment lexicon (s) performs the same as Amharic manual sentiment lexicon.
- Alternative Hypothesis H1: Amharic sentiment lexicon(s) generated from English sentiment lexicon (s) performs better than the performance of Amharic manual sentiment lexicon.

For statistical comparison of sentiment classifiers relying on lexicon features, McNemar's test is used to prove the formulated hypothesis.
As depicted in Table 9.1, performance sentiment classifiers (SVM + SWN lexicon features and SVM + manual lexicon features) is tested using McNemar's test, which yields the p-value of 0.001 (i.e. is less than the confidence level p=0.05). Therefore, the null hypothesis is rejected. That is, the SWN lexicon feature (i.e. treatment) with SVM classifier performs significantly

better than the baseline (manual lexicon feature with SVM classifier). The treatments influenced the performance of Amharic sentiment classification system.
Similarly, the SO-CAL lexicon feature with SVM classifier and SVM + manual lexicon features was tested using McNemar's test in an independent and randomly selected test set. The generated P-value (= 0.001) is less than the confidence interval (P-value of 0.05). That is, the two classifiers are skewed (i.e. disagreed on predictions of test samples).
Therefore, SO-CAL lexicon feature (i.e. treatment) with SVM classifier is significantly improving performance of Amharic sentiment classification than the manual lexicon feature. Similarly, for research question RQ2, the independent variable (generated sentiment lexicons from English lexicon) and dependent variable is Amharic sentiment classification system. Hypothesis is:

- Null Hypothesis H0: Amharic sentiment lexicon(s) generated from Amharic corpora performs the same as Amharic manual sentiment lexicon.
- Alternative Hypothesis H2: Amharic sentiment lexicon(s) generated from Amharic corpora performs better than the performance of Amharic manual sentiment lexicon.

From Table 9.1, McNemar's test yields a p-value of 0.319 which is greater than the significant value. This shows that the treatment had no effect on sentiment classification system. PPMI sentiment lexicon feature does not significantly improve performance of SVM sentiment classifier compared to SVM with manual sentiment lexicon feature. Therefore, the null hypothesis is accepted.
To address Research Question RQ3, the independent variable is Amharic text features listed in RQ3 whereas the dependent variable is the Amharic sentiment classification system. It is then mapped into a research hypothesis.

- Null Hypothesis H0: SVM with engineered Amharic text feature(s) contributes the same as SVM without engineered text features for the performance of Amharic sentiment classification system.
- Alternative Hypothesis H3: SVM with engineered Amharic text feature(s) contributes better than SVM without engineered text features for the performance of Amharic sentiment classification system.

This hypothesis is tested using McNemar's test that the classifier (SVM) with fine-tuned features (i.e. character level ngram TF-IDF as treatment) is significantly better than the same classifier with features of default settings (TF-IDF ngram). With McNemar's test of these two classifiers settings in an independently sampled test set, a p-value of 0.000127 is obtained which is less than the significant level (i.e. p = 0.05) as shown in Table 9.1. This shows that the proposed classifier (SVM) with fine-tuned features had an effect on performance of Amharic sentiment classification system.

Similarly, research question RQ4 has independent variables (i.e.treatment is tuned machine learning listed in RQ4) and dependent variable (i.e. Amharic sentiment classification system). Hypothesis is formulated as follows:

- Null Hypothesis H0: Fined tuned machine learning performs the same as machine learning with default settings on Amharic sentiment classification system.
- Alternative Hypothesis H4: Fined tuned machine learning performs better than machine learning with default settings on Amharic sentiment classification system.

For research question RQ4, a fine-tuned SVM classifier (i.e. fine-tuned hyper-parameters of SVM is a treatment) and SVM with default parameters setting is tested using McNemar's test. From Table 9.1, the p-value of 0.5418 is obtained using sampled test sets. This p-value is greater than the threshold (i.e. $p = 0.05$). This means that the null hypothesis is accepted. Thus, fine-tuned SVM classifiers have not been shown to improve performance of Amharic sentiment classification.
To answer research question RQ5, the independent variables and dependent variables are identified as proposed pre-trained (BERT) model and Amharic sentiment classification system, respectively. A testable hypothesis is formulated as:

- Null Hypothesis H0: Fine-tuning pre-trained BERT model performs the same as machine learning with hyper-parameter default settings on Amharic sentiment classification system.
- Alternative Hypothesis H5: Fine-tuning pre-trained BERT model performs better than machine learning with hyper-parameter default settings on Amharic sentiment classification system.

For research question RQ5, the McNemar's test of BERT fine-tuned pre-trained model (i.e. treatment) classifier and SVM classifier (i.e. hyper-parameters of default settings) has achieved p-value of 0.0006129 as depicted in Table 9.1. This is less than the threshold significant p-value (0.05). So the p-value is sufficient to reject the null hypothesis. Thus, the BERT fine-tuned sentiment classifier has a significant effect on the performance of Amharic sentiment classification system.
Lastly, to answer RQ6, the independent variables and dependent variables are identified as proposed stacked ensemble learner (i.e. SVM + NB + RF as base learners and LR as meta learner) and Amharic sentiment classification system, respectively. A testable hypothesis is formulated as follows:

- Null Hypothesis H0: Proposed stacked ensemble learner performs the same as SVM with hyper-parameters of default settings on Amharic sentiment classification system.

- Alternative Hypothesis H6: Proposed stacked ensemble learner performs better than SVM with hyper-parameters of default settings on Amharic sentiment classification system.

For this research question, McNemar's test of stacked ensemble learner and SVM with hyper-parameters of default settings using sampled test sets gives the p-value of 0.00528 which is less than the threshold p-value. Thus, the p-value is sufficient support for rejection of the null hypothesis. This is, stacked ensemble approach is significantly performing Amharic sentiment classification.

9.4 Prototype

In this section, the proposed solutions to perform Amharic sentiment classification and associated detection tools (i.e. language identification, topic detection and subjectivity detection) are demonstrated by developing a prototype. The prototype for Amharic sentiment classification can be shown and tested in two ways, real-time and offline modes. For real-time demonstration of the prototype of Amharic SC, comments from Facebook pages are extracted and stored using Facebook Graph API and other scraping tools. However, Facebook has set data access restrictions and the review process of data access request becomes complicated.
The most popular Facebook scraping tools like Netvizz[2] is blocked from accessing Facebook comments. Accessing Facebook data for real-time applications through its API is changed and restricted due to change of data access and user data privacy policy. So, the prototype is displayed in offline mode. Facebook raw comments are manually collected from official Facebook page (e.g. the Facebook page of PM of Ethiopia). The comments of the viewers in response to the events/issues posted on their official Facebook pages are stored in a file. Then, by feeding these raw comment samples, the performance of the prototype is tested.
Thus, manually collected Facebook raw comments are used to demonstrate the proposed system can work on real-time data to predict the sentiments of viewers' comments on a certain topic/issues. The prototype of the proposed system has four modules (i) preliminary detection module, (ii) preprocessing module, (iii) sentiment classification module and (iv) file storage module. The key procedures are presented in what follows:
First, the user loads the raw Facebook comments using file menu from file storage module. The user selects from among language detection, subjectivity detection and topic detection. If the language of the comment is Amharic, the system detects subjectivity of the comment and topic category of the comments are detected. If the comment is subjective, the user selects one of the proposed SC approaches carried out in earlier chapters, and the selected classifier predicts the sentiment of that comment. Finally, the predicted language, subjectivity, the topic and sentiment is displayed for each comments. If the language is not Amharic, only language of the comment is displayed. The graphical user interface is shown in Figure 9.2.

[2]https://tristanhotham.com/2019/08/21/goodbye-netvizz/

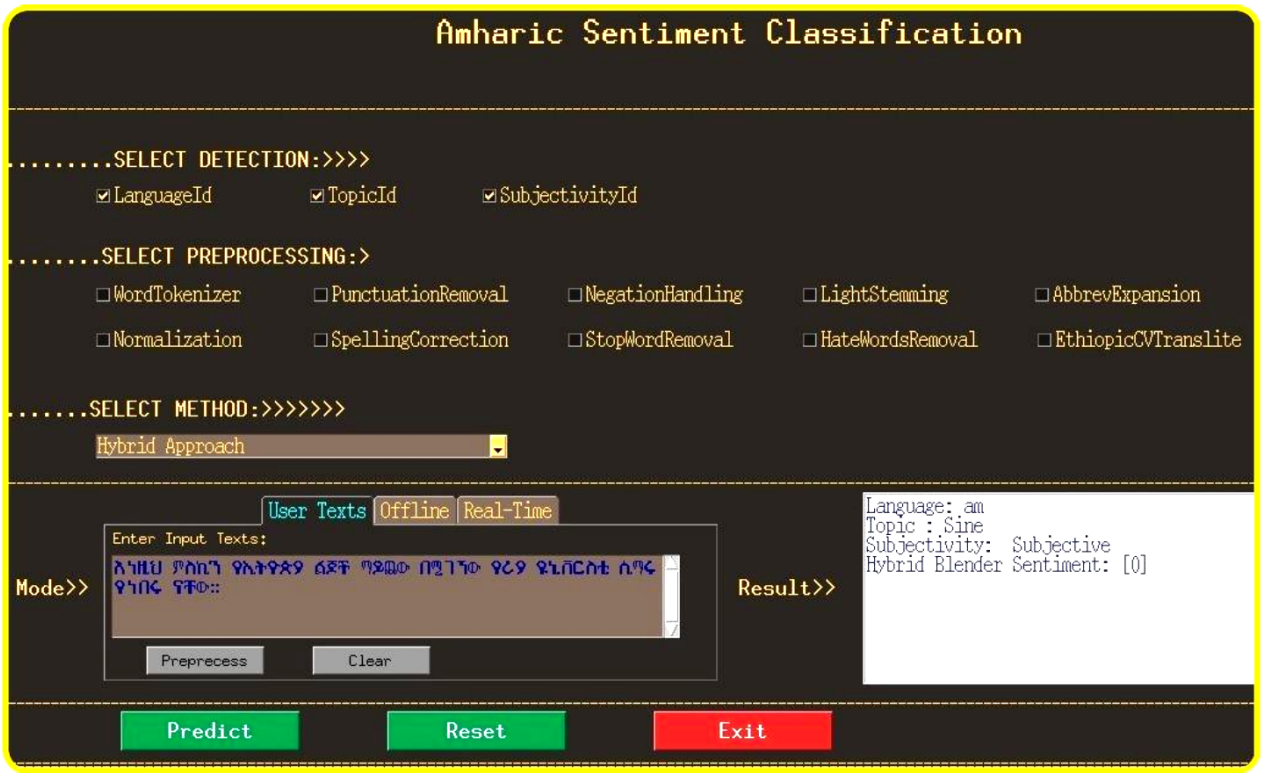


Figure 9.2.: Amharic Sentiment Classification Prototype

When a specific classifier is selected the trained model is loaded from file storage module. The best pre-trained models from the approaches in earlier chapters are recorded and deployed for further use.

9.5 Expert Evaluation

From the users (or stakeholder) point of view, this section presents the expert evaluation of the performance of the proposed Amharic sentiment classification and detector tools (i.e. language detector, topic detector and subjectivity detector) using the prototype. Does the proposed solutions yield artifacts which are useful and minimize cost and time with acceptable accuracy for carrying out Amharic SC from the perspective of the needs of the stakeholders?
The proposed approaches have been developed to support the professionals who are doing the work manually. As the GCOA[3] office is transformed due to change of governance structure of the country, the professionals are moved to other local communication media offices (i.e. EBC, WALTA and FANA). In the meantime, to the best of the researcher's knowledge, there are no professionals employed in social media section to analyze sentiment of Facebook comments of the public in response to the posting of news/events in their official pages.

[3]https://www.thereporterethiopia.com/article/gcao-transfers-employees-public-media

To qualitatively evaluate the performance of the proposed system, two professionals who were working in social media and archival sections were contacted. The professionals were familiar with how Amharic sentiment classification is analyzed for decision making. In addition, other human evaluators from psychology and communication and journalism were contacted.
Procedures of expert evaluation: The steps for expert evaluation of the prototype performance are as follows.
(i) first, the four professionals were oriented regarding the research background and its potential applications so that they would have a common understanding of the proposed system, (ii) second, the prototype was shown with visual interaction using test cases presented to those four professionals, (iii) written closed-ended questions with questionnaire forms were given to them to collect their responses about the performance of the prototype and its significance and finally, discussion of their response was reported. The summary of the evaluators' response is depicted in Table 9.2.

Table 9.2.: The Response of the Four Professionals about the Performance of the Prototype

No	Question (Accept or Reject ?)	No of responses with 'Accept'
1.	The prototype is quick, easy to use and interact with it.	4
2.	The system provides correct language identification result.	4
3.	The system is producing right topic identification result.	4
4.	The system is yielding correct subjectivity detection result.	4
5.	The system is predicting sentiment correctly and consistently.	4
6.	The proposed system is useful and cost effective assisting professionals to serve the community.	3

Discussion: The responses of the four professionals to questionnaires about the prototype are presented in Table 9.2. According to the demonstrated prototype, all the responses of the professionals for questions 1 to 5 show that all the evaluators are satisfied with the prototype performance with respect to identification of language, detection of topic, subjectivity detection and sentiment prediction of raw Facebook comments. On the other hand, the response of one of the evaluators to the last question in the questionnaire form indicates lack of satisfaction with performance of the proposed system in assisting the professionals.
Based on the general qualitative evaluation of professionals' judgment of the prototype, the overall performance of sentiment classification of the proposed approaches is shown to minimize the manual activities of extracting useful sentiment information of Facebook comments for decision making at the right time by the concerned government officials. Though Facebook restricts access to public data and sets up a new data access policy, the prototype needs to be improved and extended to work in real-time settings to summarize the sentiment of Amharic in other social media like Twitter.

9.6 Summary

This chapter presents the evaluation of the different proposed approaches for Amharic sentiment classification on Facebook comments. Hypotheses for each of these proposed approaches are formulated and set baseline. Using MCNemar's test, SVM with Amharic sentiment lexicons such as SO-CAL, SWN, TF-IDF character (1,7) ngram feature sets are statistically significant. In contrast, SVM with PPMI sentiment lexicon is proved insignificant. Ensemble model with TF-IDF character (1,7) ngram feature sets and fine-tuned BERT pre-trained model is significant for the performance of Amharic sentiment classification.
Based on these hypotheses, the performances of the proposed approaches can further be revised to improve the performance of sentiment classification.

10. Contributions

This chapter presents the key contributions of this research. The contributions are organized into major and minor contributions. The major contributions are related to lexicon generation approaches, feature and machine learning selection, hyper-parameter tuning and hybridization of heterogeneous approaches for improving performance of sentiment classification. Minor contributions comprise sentiment labeled data sets, subjectivity, pre-trained models, and other related documents. In addition, the approaches used to generate resources can also be adapted to other less-resourced languages.

10.1 Major Contributions

10.1.1. Sentiment lexicon Generation

Three Amharic sentiment lexicons are generated. These include Amharic SO-CAL (5,681) and Amharic SWN (13,677) using dictionary-based and Amharic PPMI Sentiment Lexicon (8,131) using corpus-based generation. Dictionary based sentiment lexicon generation resulted in the main contributions: (i) the algorithm uses average weighting of synonyms of English tokens to assign the sentiment score for the corresponding Amharic term in the bilingual Amharic-English dictionary, and (ii) the approach in this work can also handle negation words that appear in the Amharic text. On the other hand, the corpus based generated lexicon has not achieving significant effect on sentiment classification. However, the development of corpus-based algorithm using PPMI matrix that generates Amharic sentiment lexicon with minimal manual inspection can be taken as one of the contributions of this research.

Both of the developed lexicon generation approaches (i.e. dictionary based and corpus based) reduce cost and time of labeling Amharic sentiment terms to be added in sentiment lexicons. With minimal cost of human inspection, the developed approaches are generic enough that they can be adapted to generate sentiment lexicon for other less-resourced local languages. The entries in the generated sentiment lexicons contain part-of-speech information that can be applied or utilized in other linguistic tasks of interest.

Yet, there is still room for improvement in the approach that generates the sentiment lexicons.

10.1.2. Selecting Best Text Feature Sets, Best Model and Searching for Best Hyper-parameter

This research makes contributions related to issues such as (i) Feature Set Selection: The best text feature sets are identified, extracted and selected for sentiment classification in social media. The text feature sets investigated are CountVectorizer character/word ngram features, TF-IDF Vectorizer character/word ngram features, averaged word embedding feature vector, TF-IDF Weighted word embeddings feature vector, LDA, LSI and NMF. All these feature sets are extracted from data sets. Each of the feature set is used to train the most common machine learning algorithms, i.e. SVM, NB, and RF. With default values of these algorithms, the best performing feature set is TF-IDF Vectorizer character ngram feature set using SVM algorithm. (ii) Model Selection and Hyper-parameter tuning: Among the most popular machine learning algorithms i.e. SVM, NB, and RF, SVM has achieved the best sentiment classification performance compared with other classifiers. SVM's hyper-parameters are also optimized by grid search CV using training sets with the TF-IDF Vectorizer character ngram feature sets. Similarly, the hyper-parameters of other machine learning algorithms are also optimized. The final SVM with its optimized hyper-parameter values consistently achieves performance gains in sentiment classification as compared to the other classifiers (i.e. RF, NB).

10.1.3. BERT fine-tune for Amharic Sentiment Classification

Transfer learning of Amharic language models (BERT) fine-tuned to downstream tasks (i.e. Amharic sentiment classifications) is evaluated. The key contributions of the proposed BERT fine-tuning approach for Amharic sentiment classification include: (i) the development of the first BERT tokenizer and BERT model for Amharic language and (ii) the fine-tuning of the pre-trained BERT model for Amharic sentiment classification, (iii) a strong indication that the language model can be used for other downstream tasks of the Amharic language, and (iv) a notable performance improvement on sentiment classification of Amharic language as compared with performance of baseline models (SVM). Because of limited computational resource like GPU, the pre-trained BERT for Amharic language model is small and needs to be expanded to a scale at which it could achieve performance gains in different tasks of Amharic language.

10.1.4. Ensemble and Hybrid Approach for Amharic Sentiment Classification

The research combined heterogeneous machine learning algorithms for Amharic sentiment classification through meta-learner. The key contribution of meta-learners (i.e. ensemble learner) for Amharic sentiment classification is the demonstration that stacking of base classifiers with

meta learner improves performance of sentiment classification as compared to stacking without meta learner (SVM, NB, and RF).
New hybrid approach combines heterogeneous sentiment classifiers (lexicon-based, tuned-SVM, ensemble learner and fine-tuned-BERT) and it is demonstrated that it works the best when compared to the performance of the individual classifier. Evaluation of the proposed hybrid model has revealed that the combination of base-classifiers outperforms the individual classifiers. The hybrid model makes use of various fusion strategies: majority voting, weighted voting and blending learner. Blending learner has achieved better performance than the other hybrid combinations. Errors are also tracked, showing that some samples wrongly predicted by lexicon-based classifier are correctly classified by the SVM classifier. If SVM wrongly predicts a sample, the BERT fine-tuned classifier may correctly predict it. This is logically consistent because BERT has a better capability for capturing different contexts of the same word depending on its usage. As per the knowledge of the researcher, this approach has never been tested before for Amharic sentiment classification. Besides, the approach is recommended to be adapted for other less-resourced languages where there is no large-scale sentiment lexicon and insufficient sentiment labeled corpora.

10.2 Minor Contributions

10.2.1. Pre-trained Models and Related Codes

The best pre-trained models such as language detector, topic detector, PPMI model, word embeddings, supervised pre-trained models, ensemble pre-trained model, Amharic BERT WordPiece Tokenizer, Amharic BERT pre-trained Language model, BERT fine-tuned for Amharic sentiment classification pre-trained model, and associated source code files are one of the contributions of this research.

10.2.2. Collected and Labeled Data Sets

Different general corpora and sentiment data sets in Amharic language are collected and prepared. The labeled collection can be used as a benchmark for the future researcher in the field. The data sets can be further enhanced and upgraded for other related tasks of sentiment classification for interested future researchers in this area.

10.2.3. Subjectivity Detection for Amharic Sentiment Classification

The social media generated texts usually contain user opinions which carry subjectivity expressing user stances in favor or against an issue (or topic) or sometimes free from sentiment

orientations (i.e. neutral /factual/objective/fact). In this research, subjectivity detection technique is developed to filter objective texts from subjective texts, given the social media texts. Therefore, the generation of subjectivity detection method enhances efficiency of further tasks of sentiment classification. The subjectivity detection method uses the generated sentiment lexicons to filter texts with no sentiment. The sentiment classification system receives only texts with subjective information so that it learns to detect either in favor (positive sentiment) or against (negative sentiment). As subjectivity detection has not been dealt with before in Amharic language, more research besides the subjectivity technique which is developed here.
So far, it has been seen that the proposed approach still lacks accuracy for sentiment analysis of a specific language, i.e. Amharic. The approach potentially does not sufficiently capture the language-specific features that help to identify the sentiment class of Amharic comment in social media as Facebook. Further work should be performed to reduce the amount of errors in sentiment analysis of user generated comments.

10.2.4. Relevance of Work to other Low-resource Languages

By using these small sentiment labeled datasets (of Category II), the results show to what extent conventional machine learning, Stack Ensembler, BERT, and a combination of these perform well across the four data sets for sentiment classification. For example, using PMO data sets, the performance of the different approaches such as SVM, Stack Ensembler, BERT, and hybrid model (blender) have achieved an accuracy of 89%, 90%, 95%, and 95%, respectively. SVM and Stack Ensembler have used TF-IDF character (1,7) grams, whereas BERT has used BERT wordpiece embeddings and the hybrid model has used a combination of the above features. In general, the findings of the research will give more insights and can help future researchers to extend this research or adapt the approaches to address similar challenges of sentiment classification in other less-resourced languages.

11. Conclusion and Future Work

11.1 Conclusion

This research addressed the formulated problems, the lack of sentiment lexical resources and insufficient annotated Amharic sentiment corpora.

For the first problem, three Amharic sentiment lexicons are generated. The first two sentiment lexicons (Amharic SWN and Amharic SO-CAL) are generated using a dictionary-based approach from English SWN 3.0 and English SO-CAL. For propagating sentiment labels from English to Amharic terms, an Amharic-English bilingual dictionary is compiled. However, the generated lexicons from English language have a problem of handling language-specific sentiment connotations which are not used in the source language. To handle this, an Amharic sentiment lexicon from an enormous collection of Amharic corpora is generated using a corpus-based approach. In this approach, a PPMI matrix is built from the corpora. PPMI focuses on capturing the count of co-occurrence terms, and the matrix holds this co-occurence information of terms. With this approach, Amharic seed words are expanded to the PPMI-based Amharic sentiment lexicon. Each of these sentiment lexicons are comprehensively evaluated using metrics such as coverage, subjectivity detection rate, agreement, and sentiment classification performance. Based on these evaluations, each of the generated sentiment lexicon outperforms the manual sentiment lexicons [49].

Still, these sentiment lexicons require manual inspection or an approach that can improve the quality and the size of the sentiment lexicons. The combinations of these sentiment lexicons have shown performance gains as compared to the individual sentiment lexicon's performance. As the proposed approaches are not language-dependent, the approach can be adopted to other resource-limited languages.

For the second problem (lack of sufficient labeled sentiment corpus), different approaches that can work with insufficiently labeled data sets are investigated. The various text features (such as TF-IDF, BOW,word embeddings), and various machine learning approaches (supervised, ensemble methods, transfer learning approach) are tested. Among the generated text features, character level TF-IDF performed best for sentiment classification in small labeled data sets. The popular machine learning algorithms (NB, SVM, LR, RF, and so on) are also tested to select the one that performs best on Amharic sentiment classification with available labeled data sets. SVM has shown the best performance as compared to the other classifiers. SVM classifier

with its default settings of hyper-parameters is used as a benchmark for investigating other proposed approaches to sentiment classification. SVM hyper-parameters are tuned. However, the performance is not significantly better than the performance of SVM with default settings. In similar approach, an ensemble of SVM, NB, and RF as base learners and LR as meta learners are built. This ensemble method performed significantly better than the baseline.
In another approach, a BERT model for Amharic language is fine-tuned from scratch for transfer learning to carry out Amharic sentiment classification. Fine-tuned BERT has significantly improved the performance of Amharic sentiment classification as compared to the other approaches.
Finally, a new hybrid approach is proposed by combining the best sentiment classifiers (SVM-tuned, ensemble-stack, BERT-fine-tuned) using different hybridization methods (i.e. voting, weighted averaging, and meta-classifier). The performance of the proposed hybrid approach with a meta-classifier (i.e. LR) performs the best of all other approaches.
Besides, the MCNemar's Test report in Table 9.1, the answers for the formulated research questions (RQ1 to RQ6) are answered in Table 6.4, Table 6.11, Table 7.2, Table 7.4, Table 7.7, and Table 7.5 & 8.2, respectively.

11.2 Future Work

This research lays foundations for researchers interested in this avenue. However, it still has several limitations. This research does not consider emoji features for sentiment classification. If the user generated content contains emojis, it is important to automatically and precisely label social media contents. Unlabeled Facebook comments can be labeled based on the emoji feature. So, future research should address the potential benefits of effects of emoji features more carefully for sentiment classification of user generated Amharic texts where there is insufficient labeled datasets for training machine learning algorithm. There are a few researchers [44] who tried to use emoji for auto-labeling for social media comments, i.e. Facebook comments and YouTube comments. However, this area requires more work to leverage this feature for supplementing other resources to increase the number of labeled datasets rather than wasting resources such as time, budget, and human power.
Social media users write comments in different languages on the same topic or issues. Languages such as Amharic, Afaan Oromo,English, Tigrinya and Somali are frequently used. Users who write in Amharic use either Amharic alphabet or Latin alphabet to write their opinions. This research developed detection of this language to filter Amharic comments written in Amharic alphabet. However, there is an enormous resource to be leveraged by including the comments written in other languages for carrying out multilingual sentiment classification [35].
The current social media comments and reviews are not mostly in favor of (or against) a certain topic or issues by a particular party or group. Stance is defined as the expression of a

speaker's standpoint or judgment towards an issue. The emerging paradigm of opinion mining to analyze the stance of a person towards a target topic is called stance detection. To get such information, the government of Ethiopia employed professionals to review and summarize social user generated contents semantically oriented either positively or negatively towards an issue on news posted, for example, the opinion of Facebook users comments towards or against a particular national issues or events. Apart from sentiment classification, stance detection is recommended as early on stance detection prediction is more important than after the event or issues [38]. If the stance is predicted before a particular incident happens, government or concerned officials can take action before the occurrence of the loss of lives and properties. Before certain ethnic conflicts happen, stance prediction can provide information, enabling a decision to take the right measure at the right time. Besides, once the stance of the public (favor or against) a certain issue or topic is detected, this information helps to validate and reject fake information which might trigger religious, nationalist, and/or ethnic conflict [32]. Similarly, after the stances or sentiment of the majority is detected, hate speech can be detected and get filtered. This permits the removal of different conflict triggering contents which may be fake or hate speech user content. Hate speech filtering is one of the most essential research avenue for future researchers. Due to lack of GPU facilities, a large scale BERT pre-trained model for Amharic has not been built. Such a model is so essential to carry out other downstream NLP tasks by transfer learning using small labeled dataset in the target tasks. This is important in future work to be done by interested researchers as many languages are resource-limited. This research will leverage the advanced NLP technologies to address various problems and can result in various NLP applications like chat bot, translator, recommender, auto-help desk, just to name a few. Such NLP applications in local languages will minimize religious-nationalist-ethnic division by reducing communication barriers in the community.

As the datasets are extended and the quality of the sentiment lexicons is enhanced, government change and sentiment words might accordingly change. As a result, time series sentiment lexicon and labeled sentiment corpora are required. It is important to understand the sentiment by time and by a particular government during a time-span. As governments change structurally and politically the direction of policy also changes to a certain extent. Therefore, sentiment words also change accordingly and further research is therefore called for.

It is also recommended that researchers extend the datasets to incorporate more sentiment granularity. Apart from Facebook comments, other domains like entertainment and product reviews in e-commerce platforms are potential research avenues of sentiment classification for the future researchers.

A. Amharic Facebook Comments' Sentiment Annotation Guideline

This section provides guidelines how to annotate Facebook comments for the purpose of this research i.e. to detect sentiment of Amharic Facebook comments. The annotated data sets is used for only research purpose. The raters are gently requested to label each Amharic comments as either positive and negative depending on the judgments expressed in the comments towards an issue or events. So, raters are expected to annotate comments as either positive (i.e. favor the issues written in the headline) or negative (i.e. against the headline). Definition of the key terms such as positive , negative or neutral (none) are provided with example cases.

Positive sentiment comments are users' opinion which favors/supports the issue. These comments usually contains positive opinion words e.g. **ጥሩ**/good/, **አድናቂ**/wonderful/ **ደስተኛ**/happy/, **አወድሃለሁ**/like you/, **ድንቅ**/fabulous/, etc.

Negative sentiment comments are users' opinion which are against the issue. These comments usually contains negative opinion words e.g.**እፍፍፍፍፍ**/f*ck/, **ደደብ**/dull/, **ደነዝ**/numbed/, **አዝናለሁ**/sorry/, **አልወድም**/hate/, **ቲሽ**/bastard/, etc.

The comments are collected from official Facebook page of PMO. Some Facebook comments express favor and support on an issue, some viewers express against the issue, and some other viewers might be neutral (i.e. comment does not express opinion). Example of each category is as follows in Table A.1.

Table A.1.: Annotation Example of Facebook Comments of PMO Facebook Posts

Facebook Comments (Am)	English Translated	Explanation of Annotation
1. **ጥሩ መስራት በታሪክ ያዘክራል።**	/A good work that is remembered in history/	This is positive comment as it favors by using positive words like good.
2. **የጀግና ሚስት ጀግና ናት !**	/A hero's wife is a hero mother./	This comment is also a positive comment as it is expressed by using sentiment word like hero.
3. **በርሀብ የሚስቃዬትንም ጉብኝ**	/let you visit those are in hunger/	This is negative comment as it is problem of hunger that needs government support to those in need.
4. **ግም ለግም አብረህ አዝግም**	/People who are doing bad, learns doing nothing new/	This comment is negative as passes negative message to those government officials who are weak , no good job expected from such leaders.
5. **ሺ ዓምት ንገስ**	/let your empire extends to thousands years/	This is a positive comment as it the current PM is doing satisfactorily. The viewer gives praise to PM by wishing his power to be extended a thousand years.

Facebook Comments with neutral comments are those comments which contain neither positive nor positive sentiments. In other words, the view of the comments does not contain sentiment information to support the issues nor against it. The subjectivity of the comment is automatically detected. These subjective comments are also manually checked to remove the remaining neutral (or objective comments) prior to distributing the comments to the annotators. Two annotators are employed to label the Facebook comments and the intera-annotator agreement is calculated using Cohen Kappa in chapter 4. The guideline is prepared in consultation of linguists and professionals.

B. Amharic Alphabet

238 Characters

	ä/e [ɛ/ə]	u [u]	i [i]	a [a]	e [e]	ə [ɨ/ʉ]	o [o/ɔ]
h [h]	ሀ ha	ሁ hu	ሂ hi	ሃ ha	ሄ he	ህ hə	ሆ ho
l [l]	ለ lä	ሉ lu	ሊ li	ላ la	ሌ le	ል lə	ሎ lo
ḥ [h]	ሐ ḥa	ሑ ḥu	ሒ ḥi	ሓ ḥa	ሔ ḥe	ሕ ḥə	ሖ ḥo
m [m]	መ mä	ሙ mu	ሚ mi	ማ ma	ሜ me	ም mə	ሞ mo
ś [s]	ሠ śä	ሡ śu	ሢ śi	ሣ śa	ሤ śe	ሥ śə	ሦ śo
r [r]	ረ rä	ሩ ru	ሪ ri	ራ ra	ሬ re	ር rə	ሮ ro
s [s]	ሰ sä	ሱ su	ሲ si	ሳ sa	ሴ se	ስ sə	ሶ so
š [ʃ]	ሸ šä	ሹ šu	ሺ ši	ሻ ša	ሼ še	ሽ šə	ሾ šo
q [k’]	ቀ qä	ቁ qu	ቂ qi	ቃ qa	ቄ qe	ቅ qə	ቆ qo
b [b]	በ bä	ቡ bu	ቢ bi	ባ ba	ቤ be	ብ bə	ቦ bo
v [v]	ቨ vä	ቩ vu	ቪ vi	ቫ va	ቬ ve	ቭ və	ቮ vo
t [t]	ተ tä	ቱ tu	ቲ ti	ታ ta	ቴ te	ት tə	ቶ to
č [ʧ]	ቸ čä	ቹču	ቺ či	ቻ ča	ቼ če	ች čə	ቾ čo
ẖ [h]	ኀ ẖa	ኁ ẖu	ኂ ẖi	ኃ ẖa	ኄ ẖe	ኅ ẖə	ኆ ẖo
n [n]	ነ nä	ኑ nu	ኒ ni	ና na	ኔ ne	ን nə	ኖ no
ny/ñ [ɲ]	ኘ nyä	ኙ nyu	ኚ nyi	ኛ nya	ኜ nye	ኝ nyə	ኞ nyo
’/ʾ [ʔ]	አ ’a	ኡ ’u	ኢ ’i	ኣ ’a	ኤ ’e	እ ’ə	ኦ ’o
	ä/e [ɛ/ə]	u [u]	i [i]	a [a]	e [e]	ə [ɨ/ʉ]	o [o/ɔ]

	ä/e [ɛ/ə]	u [u]	i [i]	a [a]	e [e]	ə [ɨ/ʉ]	o [o]
k [k]	ከ kä	ኩ ku	ኪ ki	ካ ka	ኬ ke	ክ kə	ኮ ko
ķ [h]	ኸ ķä	ኹ ķu	ኺ ķi	ኻ ķa	ኼ ķe	ኽ ķə	ኾ ķo
w [w]	ወ wä	ዉ wu	ዊ wi	ዋ wa	ዌ we	ው wə	ዎ wo
ʻ/ʿ [ʔ]	ዐ ʻa	ዑ ʻu	ዒ ʻi	ዓ ʻa	ዔ ʻe	ዕ ʻə	ዖ ʻo
z [z]	ዘ zä	ዙ zu	ዚ zi	ዛ za	ዜ ze	ዝ zə	ዞ zo
ž [ʒ]	ዠ žä	ዡ žu	ዢ ži	ዣža	ዤ že	ዥ žə	ዦ žo
y [j]	የ jä	ዩ ju	ዪ ji	ያ ja	ዬ je	ይ jə	ዮ jo
d [d]	ደ dä	ዱ du	ዲ di	ዳ da	ዴ de	ድ də	ዶ do
ǧ [ʤ]	ጀ ǧä	ጁ ǧu	ጂ ǧi	ጃ ǧa	ጄ ǧe	ጅ ǧə	ጆ ǧo
g [g]	ገ gä	ጉ gu	ጊ gi	ጋ ga	ጌ ge	ግ gə	ጎ go
ṭ [t’]	ጠ ṭä	ጡ ṭu	ጢ ṭi	ጣ ṭa	ጤ ṭe	ጥ ṭə	ጦ ṭo
č [ʧ’]	ጨ čä	ጩ ču	ጪ či	ጫ ča	ጬ če	ጭ čə	ጮ čo
ṗ [p’]	ጰ pä	ጱ pu	ጲ pi	ጳ pa	ጴ pe	ጵ pə	ጶ po
ṣ [ts]	ጸ ṣä	ጹ ṣu	ጺ ṣi	ጻ ṣa	ጼ ṣe	ጽ ṣə	ጾ ṣo
ś [ʦ]	ፀ ṡä	ፁ ṡu	ፂ ṡi	ፃ ṡa	ፄ ṡe	ፅ ṡə	ፆ ṡo
f [f]	ፈ fä	ፉ fu	ፊ fi	ፋ fa	ፌ fe	ፍ fə	ፎ fo
p [p]	ፐ pä	ፑ pu	ፒ pi	ፓ pa	ፔ pe	ፕ pə	ፖ po
	ä/e [ɛ/ə]	u [u]	i [i]	a [a]	e [e]	ə [ɨ/ʉ]	o [o]

አማርኛ ፊደል

Amharic script

9 Punctuation Sy

※	section mark	÷	colon
:	word separator	፦	preface colon
።	full stop (perio	፧	question mark
፣	comma	⁝	paragraph separat
፤	semicolon		

20 of Amharic Numerals

፩	፪	፫	፬	፭	፮	፯	፰	፱	፲
አንዶ	ሁለት	ሰለት	አራት	አምስት	ስዶዶስት	ሰባት	ስምንት	ዘጠኝ	አስር
and	hulätt	sost	aratt	amməst	səddəst	säbatt	səmmənt	zätäññ	asser
1	2	3	4	5	6	7	8	9	10
፳	፴	፵	፶	፷	፸	፹	፺	፻	፼
ሠ	ሰላሳ	አርባ	ሀምሳ	ስልሳ	ሰባ	ሰማንያ	ዘጠና	መቶ	ሺ
haya	sälasa	arba	hamsa	səlsa	säba	sämaña	zätena	mäto	ši
20	30	40	50	60	70	80	90	100	1000

58 Characters

ሏ [lʷa]	ሗ [hʷa]	ሟ [mʷa]	ሧ [sʷa]	ሯ [rʷa]	ሷ [sʷa]	ሿ [ʃʷa]	ቈ [k’ʷə]	ቍ [k’ʷi]
ቋ [k’ʷa]	ቌ [k’ʷe]	ቍ [k’ʷɨ/ʉ]	ቧ [bʷa]	ቯ [vʷa]	ቷ [tʷa]	ቿ [ʧʷa]	ኈ [hʷə]	ኈ [hʷi]
ኋ [hʷa]	ኌ [hʷe]	ኍ [hʷɨ/ʉ]	ኗ [nʷa]	ኟ [ɲʷa]	ኧ [ʔa]	ኰ [k’ʷə]	ኰ [kʷi]	ኳ [kʷa]
ኲ [kʷe]	ኵ [kʷɨ/ʉ]	ኼ [hʷə]	ዂ [hʷi]	ዃ [hʷa]	ዄ [hʷe]	ዅ [hʷɨ/ʉ]	ዟ [zʷa]	ዧ [ʒʷa]
ዷ [tsʼʷa]	ጇ [ʤʷa]	ጐ [gʷə]	ጕ [gʷi]	ጓ [gʷa]	ጔ [gʷe]	ጕ [gʷɨ/ʉ]	ጧ [tʼʷa]	ጯ [ʧʼʷa]
ጷ [pʼʷa]	ጿ [tsʷa]	ፏ [fʷa]	ᎍ [pʷa]					

C. Approval Page

APPROVAL SHEET

This is to certify that the dissertation prepared by Girma Neshir Alemneh, entitled **"Hybrid Model for Amharic Sentiment Classification"** submitted in fulfillment of the requirement of Doctor of Philosophy in Information Technology (Information Retrieval) complies with the regulation of the university and meets the accepted standards with respect to originality, content, and quality.

Signed by the dissertation examination committee

Name	**Signature**	**Date**
_____	_____	_____
(**Program Chairman**)		
_____	_____	_____
(**External Examiner**)		
_____	_____	_____
(**Internal Examiner**)		

D. List of Publications

1. Girma Neshir Alemneh, Andreas Rauber, Solomon Atnafu, **"Dictionary Based Amharic Sentiment Lexicon Generation"**, Information and Communication Technology for Development for Africa, 2019, p311-326, Springer International Publishing, https://www.springer-professional.de/en/dictionary-based-amharic-sentiment-lexicon-generation/17031764

2. Girma Neshir Alemneh, Andreas Rauber, Solomon Atnafu, **"Negation handling for Amharic sentiment classification"**, 2020, Workshop Co-located with Association for the Advancement of Artificial Intelligence (AAAI) 2020, http://kdd.cs.ksu.edu/Workshops/AAAI-2021/1.pdf

3. Girma Neshir Alemneh, Andreas Rauber, Solomon Atnafu, **"Corpus based Amharic sentiment lexicon generation"**, CEUR Proceedings, Forum Artificial Intelligence Research (FAIR), 2019, http://ceur-ws.org/Vol-2540/FAIR2019_paper_6.pdf, and a workshop co-located in Association for the Advancement of Artificial Intelligence (AAAI), http://kdd.cs.ksu.edu/Workshops/AAAI-2021/2.pdf,

4. Girma Neshir Alemneh, Andreas Rauber, Solomon Atnafu, **"BERT Fine-Tuning for Amharic Sentiment Classification"**, 2020, Workshop Co-located with Eighth Swedish Language Technology Conference (SLTC).

5. Girma Neshir Alemneh, Andreas Rauber, Solomon Atnafu, **"Topic Modeling for Amharic Texts in Social Media"**, Accepted for Publication in Book Proceeding of European Alliance for Innovation (EAI) Core Series, 9th EAI International Conference on Advancements of Science and Technology (ICAST), 2021.

Bibliography

[1] Ahmed Abbasi, Hsinchun Chen, and Arab Salem. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *Association of Computing Machinery (ACM) Transactions on Information Systems (TOIS)*, 26(3):12, 2008.

[2] Amine Abdaoui, Jérôme Azé, Sandra Bringay, and Pascal Poncelet. Feel: A french expanded emotion lexicon. *Language Resources and Evaluation*, 51(3):833–855, 2017.

[3] Basant Agarwal and Namita Mittal. Text classification using machine learning methods-a survey. In *Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), December 28-30, 2012*, pages 701–709. Springer, 2014.

[4] Julien Ah-Pine and Edmundo-Pavel Soriano-Morales. A study of synthetic oversampling for twitter imbalanced sentiment analysis. In *Workshop on Interactions between Data Mining and Natural Language Processing (DMNLP)*, 2016.

[5] Munir Ahmad, Shabib Aftab, Muhammad Salman Bashir, Noureen Hameed, Iftikhar Ali, and Zahid Nawaz. Svm optimization for sentiment analysis. *International Journal of Advanced Computer Science and Applications*, 9(4):393–398, 2018.

[6] Yassine Al Amrani, Mohamed Lazaar, and Kamal Eddine El Kadiri. Random forest and support vector machine-based hybrid approach to sentiment analysis. *Procedia Computer Science*, 127:511–520, 2018.

[7] Raed Kamil Naser-Maslinda Mohd Nadzir Alaa Thamer Mahmood, Siti Sakira Kamaruddin. A combination of lexicon and machine learning approaches for sentiment analysis on facebook. *Journal of System and Management Sciences*, 10 (2020) No.3:140–150, 2020.

[8] Haifa K. Aldayel and Aqil M. Azmi. Arabic tweets sentiment analysis–a hybrid scheme. *Journal of Information Science*, 42(6):782–797, 2016.

[9] Nega Alemayehu and Peter Willett. Stemming of amharic words for information retrieval. *Literary and Linguistic computing*, 17(1):1–17, 2002.

[10] Girma Neshir Alemneh, Andreas Rauber, and Solomon Atnafu. Negation handling for amharic sentiment classification. In *Proceedings of the The Fourth Widening Natural Language Processing Workshop*, pages 4–6, 2020.

[11] D. Alessia, Fernando Ferri, Patrizia Grifoni, and Tiziana Guzzo. Approaches, tools and applications for sentiment analysis implementation. *International Journal of Computer Applications*, 125(3), 2015.

[12] Rana Alnashwan, Adrian P O'Riordan, Humphrey Sorensen, and Cathal Hoare. Improving sentiment analysis through ensemble learning of meta-level features. In *Centro de Estudios Urbano Regionales (CEUR) Workshop Proceedings*, volume 1748. Sun SITE Central Europe (CEUR)/RWTH Aachen University, 2016.

[13] Ethem Alpaydm. Combined 5x2 cv f-test for comparing supervised classification learning algorithms. *Dalle Molle Institute for Perceptual Artificial Intelligence*, 1998.

[14] Paul Anderson et al. *What is Web 2.0?: Ideas, Technologies and Implications for Education*, volume 1. Joint Information Systems Committee (JISC) Bristol, 2007.

[15] Wissam Antoun, Fady Baly, and Hazem Hajj. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*, 2020.

[16] Atelach Alemu Argaw and Lars Asker. An amharic stemmer: Reducing words to their citation forms. In *Proceedings of the 2007 workshop on computational approaches to semitic languages: Common issues and resources*, pages 104–110. Association for Computational Linguistics, 2007.

[17] Eugenio Artinez-Cámara, M. Teresa Martín-Valdivia, M. Dolores Molina-González, and José M. Perea-Ortega. Integrating spanish lexical resources by meta-classifiers for polarity classification. *Journal of Information Science*, 40(4):538–554, 2014.

[18] Ebru Aydoğan and M Ali Akcayol. A comprehensive survey for sentiment analysis tasks using machine learning techniques. In *International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, pages 1–7. Institute of Electrical and Electronics Engineers (IEEE), 2016.

[19] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Language Resources and Evaluation Conference (LREC)*, volume 10, pages 2200–2204, 2010.

[20] Alexandra Balahur and Marco Turchi. Comparative experiments for multilingual sentiment analysis using machine translation. In *The 1st International Workshop on Sentiment Discovery from Affective Data (SDAD)*, page 75, 2012.

[21] Marco Bonzanini. *Mastering Social Media Mining with Python*. Packt Publishing Ltd, 2016.

[22] Samuel Brody and Noemie Elhadad. An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 804–812. Association for Computational Linguistics, 2010.

[23] Jason Brownlee. Feature selection for machine learning in python. *Machine learning mastery*, 20, 2016.

[24] Jason Brownlee. *Master Machine Learning Algorithms: Discover How They Work and Implement Them From Scratch*. Machine Learning Mastery, 2016.

[25] Jason Brownlee. A gentle introduction to transfer learning for deep learning. *Machine Learning Mastery*, 2017.

[26] Jason Brownlee. *Statistical Methods for Machine Learning: Discover How to Transform Data into Knowledge with Python*. Machine Learning Mastery, 2018.

[27] John A. Bullinaria and Joseph P. Levy. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526, 2007.

[28] Erik Cambria. Affective computing and sentiment analysis. *The Institute of Electrical and Electronics Engineers (IEEE) Intelligent Systems*, 31(2):102–107, 2016.

[29] Wei-Lun Chao. Machine learning tutorial. *National Taiwan University*, 2011.

[30] Pimwadee Chaovalit and Lina Zhou. Movie review mining: A comparison between supervised and unsupervised classification approaches. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pages 112c–112c. The Institute of Electrical and Electronics Engineers (IEEE), 2005.

[31] Iti Chaturvedi, Erik Cambria, Roy E Welsch, and Francisco Herrera. Distinguishing between facts and opinions for sentiment analysis: Survey and challenges. *Information Fusion*, 44:65–77, 2018.

[32] Ali K Chaudhry, Darren Baker, and Philipp Thun-Hohenstein. Stance detection for the fake news challenge: Identifying textual relationships with deep neural nets. *CS224n: Natural Language Processing with Deep Learning*, 2017.

[33] Yanqing Chen and Steven Skiena. Building sentiment lexicons for all major languages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 383–389, 2014.

[34] Amitava Das and Sivaji Bandyopadhyay. Sentiwordnet for indian languages. In *Proceedings of the Eighth Workshop on Asian Language Resouces*, pages 56–63, 2010.

[35] Kerstin Denecke. Using sentiwordnet for multilingual sentiment analysis. In *The Institute of Electrical and Electronics Engineers (IEEE) 24th International Conference on Data Engineering Workshop*, pages 507–512. IEEE, 2008.

[36] Chilote Dessalew. Public sentiment analysis for amharic news. Master's thesis, Bahir Dar University, 2019.

[37] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[38] Kuntal Dey, Ritvik Shrivastava, and Saroj Kaushik. Twitter stance detection — a subjectivity and sentiment polarity inspired two-phase approach. In *The Institute of Electrical and Electronics Engineers (IEEE) International Conference on Data Mining workshops (ICDMW)*, pages 365–372. IEEE, 2017.

[39] Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. 1896.

[40] Xiaowen Ding, Bing Liu, and Philip S Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of International conference on Web Search and Data Mining*, pages 231–240. Association of Computing Machinery (ACM), 2008.

[41] Cairns Doug and Meng Xiangxiang. Nlp with bert: Sentiment analysis using sas deep learning and dlpy. *SAS Institute Inc.*, 2020.

[42] Rehab Duwairi and Mahmoud El-Orfali. A study of the effects of preprocessing strategies on sentiment analysis for arabic text. *Journal of Information Science*, 40(4):501–513, 2014.

[43] Rehab M. Duwairi and Islam Qarqaz. Arabic sentiment analysis using supervised classification. In *International Conference on Future Internet of Things and Cloud*, pages 579–583. The Institute of Electrical and Electronics Engineers (IEEE), 2014.

[44] Turegn Fikre. Effect of preprocessing on long short term memory-based sentiment analysis for amharic language. Master's thesis, Addis Ababa University, 2020.

[45] Genet Mezemir Fikremariam. Automatic stemming for amharic text- an experiment using successor variety approach. Unpublished Masters Thesis and Department of Information Science and Addis Ababa University and Addis Ababa, 2009.

[46] Yitna Firdyiwek and Daniel Yaqob. The system for ethiopic representation in ascii. 1997.

[47] Björn Gambäck. Tagging and verifying an amharic news corpus. *Language Technology for Normalisation of Less-Resourced Languages*, page 79, 2012.

[48] Michael Gasser. Hornmorpho: A system for morphological processing of amharic, oromo, and tigrinya. In *Conference on Human Language Technology for Development, Alexandria, Egypt*, 2011.

[49] S. Gebremeskel. Sentiment mining model for opinionated amharic texts. Unpublished Masters Thesis and Department of Computer Science and Addis Ababa University and Addis Ababa, 2010.

[50] Siji George C G and B.Sumathi. Grid search tuning of hyperparameters in random forest classifier for customer feedback sentiment prediction. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 11(9), 2020.

[51] Mebrahtu Tadesse G.Medhin. Trilingual sentiment analysis on social media. Master's thesis, Addis Ababa University, 2018.

[52] HaBiT. Harvesting big text data for under-resourced languages. Addis Ababa University, Masarykova Univerzita, Norges Teknisk-naturvitenskapelige Universitet, The University of Oslo and Hawassa University, http://habit-project.eu/wiki/HabitSystemFinal, 2016.

[53] William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2016, page 595. The National Institutes of Health (NIH) Public Access, 2016.

[54] Tri Handhika, A. Fahrurozi, Ilmiyati Sari, Dewi P. Lestari, Revaldo Ilfestra Metzi Zen, et al. Hybrid method for sentiment analysis using homogeneous ensemble classifier. In *2nd International Conference of Computer and Informatics Engineering (IC2IE)*, pages 232–236. Institute of Electrical and Electronics Engineers (IEEE), 2019.

[55] Ammar Hassan, Ahmed Abbasi, and Daniel Zeng. Twitter sentiment analysis: A bootstrap ensemble framework. In *International Conference on Social Computing*, pages 357–364. Institute of Electrical and Electronics Engineers (IEEE), 2013.

[56] Yulan He, Alani Harith, and Deyu Zhou. Exploring english lexicon knowledge for chinese sentiment analysis. In *Canadian Information Processing Society (CIPS)-SIGHAN Joint Conference on Chinese Language Processing*, 2010.

[57] Benjamin Heinzerling and Michael Strube. Bpemb: Tokenization-free pre-trained subword embeddings in 275 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[58] Alan Hevner and Samir Chatterjee. Design science research in information systems. In *Design Research in Information Systems*, pages 9–22. Springer, 2010.

[59] Ricky Ho. Big data machine learning: Patterns for predictive analytics. *DZone Refcardz*, 158(4), 2012.

[60] Markus Hofmann and Andrew Chisholm. *Text Mining and Visualization: Case Studies Using Open-Source Tools*, volume 40. CRC Press, 2016.

[61] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, 2018.

[62] Kuo-Wei Hsu. A theoretical analysis of why hybrid ensembles work. *Computational Intelligence and Neuroscience*, 2017.

[63] Mark Janse. Language death and language maintenance: Problems and prospects. In *Language Death and Language Maintenance: Theoretical, Practical And Descriptive Approaches*, pages 9–17. John Benjamins, 2003.

[64] David Julian. *Designing Machine Learning Systems with Python*. Packt Publishing Ltd, 2016.

[65] Soroush Karimi and Fatemeh Sadat Shahrabadi. Sentiment analysis using bert (pretraining language representations) and deep learning on persian texts. *Iran University of Science and Technology*, 2019.

[66] Przemyslaw Kazienko, Edwin Lughofer, and Bogdan Trawinski. Editorial on the special issue "hybrid and ensemble techniques in soft computing: Recent advances and emerging trends", 2015.

[67] Jacqueline Kazmaier and Jan H van Vuuren. A generic framework for sentiment analysis: Leveraging opinion-bearing data to inform decision making. *Decision Support Systems*, 135:113304, 2020.

[68] Worku Kelemework. Automatic amharic text news classification: A neural networks approach. *Ethiopian Journal of Science and Technology*, 6(2):127–137, 2013.

[69] Alistair Kennedy and Diana Inkpen. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22(2):110–125, 2006.

[70] Madiha Khalid, Imran Ashraf, Arif Mehmood, Saleem Ullah, Maqsood Ahmad, and Gyu Sang Choi. Gbsvm: Sentiment classification from unstructured reviews using ensemble classifier. *Applied Sciences*, 10(8):2788, 2020.

[71] Benjamin Philip King. *Practical Natural Language Processing for Low-Resource Languages*. Phd thesis, Department of Computer Science, University of Michigan,USA, 2015.

[72] Paul R. Kleinginna and Anne M. Kleinginna. A categorized list of emotion definitions, with suggestions for a consensual definition. *Motivation and emotion*, 5(4):345–379, 1981.

[73] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura E. Barnes, and Donald E. Brown. Text classification algorithms: A survey. *Information*, 10(4), 2019.

[74] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[75] Fazal Masud Kundi, Aurangzeb Khan, Shakeel Ahmad, and Muhammad Zubair Asghar. Lexicon-based sentiment analysis in the social web. *Journal of Basic and Applied Scientific Research*, 4(6):238–48, 2014.

[76] Khang Nhut Lam, Feras Al Tarouti, and Jugal Kalita. Creating lexical resources for endangered languages. In *Proceedings of Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 54–62, 2014.

[77] J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174, 1977.

[78] Yu-An Chung Hung-Yi Lee and James Glass. Supervised and unsupervised transfer learning for question answering. page 1585–1594, 2018.

[79] Wolf Leslau. *Introductory grammar of Amharic*, volume 21. Otto Harrassowitz Verlag, 2000.

[80] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances In Neural Information Processing Systems*, pages 2177–2185, 2014.

[81] Bing Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, 2012.

[82] Bing Liu et al. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing*, 2(2010):627–666, 2010.

[83] Ruijun Liu, Yuqian Shi, Changjiang Ji, and Ming Jia. A survey of sentiment analysis-based on transfer learning. *The Institute of Electrical and Electronics Engineers (IEEE) Access*, 7:85401–85412, 2019.

[84] Oscar Romero Llombart. Using machine learning techniques for sentiment analysis. *Computer Engineering, School of Engineering(EE),Universtat Automata De Barcelona (UAB)*, 2016.

[85] Siaw Ling Lo, Erik Cambria, Raymond Chiong, and David Cornforth. A multilingual semi-supervised approach in deriving singlish sentic patterns for polarity detection. *Knowledge-Based Systems*, 105:236–247, 2016.

[86] Siaw Ling Lo, Erik Cambria, Raymond Chiong, and David Cornforth. Multilingual sentiment analysis: From formal to informal and scarce resource languages. *Artificial Intelligence Review*, 48(4):499–527, 2017.

[87] Guoqin Ma. Tweets classification with bert in the field of disaster management. *StudentReport@, Stanford Education*, 2019.

[88] Naeem A Mahoto, Sehar Gul Khan, and Sadaquat Ali Ruk. A lexicon-based method to determine subjectivity of unstructured data. In *2018 5th International Multi-Topic ICT Conference (IMTIC)*, pages 1–7. Institute of Electrical and Electronics Engineers(IEEE), 2018.

[89] Fawaz HH Mahyoub, Muazzam A Siddiqui, and Mohamed Y Dahab. Building an arabic sentiment lexicon using semi-supervised learning. *Journal of King Saud University-Computer and Information Sciences*, 26(4):417–424, 2014.

[90] James H. Martin and Daniel Jurafsky. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Pearson/Prentice Hall Upper Saddle River, 2009.

[91] MaríA-Teresa MartíN-Valdivia, Eugenio MartíNez-CáMara, Jose-M Perea-Ortega, and L Alfonso UreñA-LóPez. Sentiment polarity detection in spanish reviews combining supervised and unsupervised approaches. *Expert Systems with Applications*, 40(10):3934–3942, 2013.

[92] Mary L. McHugh. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282, 2012.

[93] N. Medagoda, S. Shanmuganathan, and J. Whalley. Sentiment lexicon construction using sentiwordnet 3.0. *The 11th International Conference on Natural Computation (ICNC)*, page 802–807, 2015.

[94] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113, 2014.

[95] Melese Mihret and Muluneh Atinaf. Sentiment analysis model for opinionated awngi text. In *The Institute of Electrical and Electronics Engineers (IEEE) African Conference (AFRICON)*, pages 1–6. Institute of Electrical and Electronics Engineers(IEEE), 2019.

[96] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[97] Myriam D Munezero, Calkin Suero Montero, Erkki Sutinen, and John Pajunen. Are they different? affect, feeling, emotion, sentiment, and opinion detection in text. *The Institute of Electrical and Electronics Engineers (IEEE) Transactions on Affective Computing*, 5(2):101–111, 2014.

[98] Cataldo Musto, Giovanni Semeraro, and Marco Polignano. A comparison of lexicon-based approaches for sentiment analysis of microblog posts. In *The Dynamic Analysis and Replanning Tool (DART) and Artificial Inteligence (AI)*, pages 59–68. Citeseer, 2014.

[99] Cataldo Musto, Giovanni Semeraro, and Marco Polignano. A comparison of lexicon-based approaches for sentiment analysis of microblog posts. *Information Filtering and Retrieval*, 59, 2014.

[100] Adel Qasem Al-Shabi Nazlia Omar, Mohammed Albared and Tareq Al-Moslmi. Ensemble of classification algorithms for subjectivity and sentiment analysis of arabic customers' reviews. 5(14), 2013.

[101] Girma Neshir, Andeas Rauber, and Solomon Atnafu. Amharic light stemmer. 2019.

[102] Girma Neshir, Andreas Rauber, and Solomon Atnafu. Topic modeling for amharic user generated texts. *Information*, 12(10):401, 2021.

[103] Finn Årup Nielsen. A new anew: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*, 2011.

[104] Bianka Nusko, Nina Tahmasebi, and Olof Mogren. Building a sentiment lexicon for swedish. In *Digital Humanities, From Digitization to Knowledge : Resources and Methods for Semantic Processing of Digital Works/Texts, Proceedings of the Workshop, Krakow, Poland*, number 126, pages 32–37. Linköping University Electronic Press, 2016.

[105] Alexis Palmer. Computational linguistics for low-resource languages. Presentation at Saarland University, http://www.coli.uni-saarland.de/courses/CL4LRL, 2011.

[106] Girish Palshikar, Manoj Apte, Deepak Pandita, and Vikram Singh. Learning to identify subjective sentences. In *Proceedings of the 13th International Conference on Natural Language Processing*, pages 239–248, 2016.

[107] Sinno Jialin Pan and Qiang Yang. A survey on transfer tearning. *The Institute of Electrical and Electronics Engineers (IEEE) Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2009.

[108] Lucia Passaro, Laura Pollacci, and Alessandro Lenci. Item: A vector space model to bootstrap an italian emotive lexicon. In *Second Italian Conference on Computational Linguistics CLiC-IT*, pages 215–220. Academia University Press, 2015.

[109] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[110] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of National Association for the Advancement of Caucasian Latinos (NAACL)*, 2018.

[111] Scott SL Piao, Paul Rayson, Dawn Archer, Francesca Bianchi, Carmen Dayrell, Mahmoud El-Haj, Ricardo-María Jiménez, Dawn Knight, Michal Křen, Laura Löfberg, et al. Lexical coverage evaluation of large-scale multilingual semantic lexicons for twelve languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*, pages 2614–2619, 2016.

[112] Moacir P Ponti Jr. Combining classifiers: From the creation of ensembles to the decision fusion. In *24th Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens (SIBGRAPI) Conference on Graphics, Patterns, and Images Tutorials*, pages 1–10. Institute of Electrical and Electronics Engineers(IEEE), 2011.

[113] Marco Pota, Mirko Ventura, Rosario Catelli, and Massimo Esposito. An effective bert-based pipeline for twitter sentiment analysis: A case study in italian. *Sensors*, 21(1):133, 2021.

[114] Chris Potts. Distributional approaches to word meanings. https://web.stanford.edu/class/linguist236/materials, 2013. Ling 236/Psych 236c: Representations of meaning, Spring 2013.

[115] Yuqiang Chen Wenyuan Dai Yu-Feng Li Wei-Wei Tu Qiang Yang Yang Yu Quanming Yao, Mengshuo Wang. Taking the human out of learning applications: A survey on automated machine learning. *arXiv*, 2019.

[116] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

[117] Sebastian Raschka. Python machine learning unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics. 2015.

[118] Sebastian Raschka. Mlxtend: Providing machine learning and data science utilities and extensions to python scientific computing stack. *Journal of Open Source Software*, 3(24), 2018.

[119] Sebastian Raschka. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*, 2018.

[120] Sebastian Ruder. *Neural Transfer Learning for Natural Language Processing*. Phd thesis, 2019.

[121] Hassan Saif, Yulan He, and Harith Alani. Semantic sentiment analysis of twitter. In *Proceedings of the 11th International Conference on The Semantic Web-Volume Part I*, pages 508–524. Springer-Verlag, 2012.

[122] Dipanjan Sarkar. Text analytics with python. 2016.

[123] Dipanjan Sarkar, Raghav Bali, and Tushar Sharma. Practical machine learning with python. *A Problem-Solvers Guide To Building Real-World Intelligent Systems. Berkely: Apress*, 2018.

[124] Kamal Sarkar. Heterogeneous classifier ensemble for sentiment analysis of bengali and hindi tweets. *Sādhanā*, 45(1):1–17, 2020.

[125] Ranjan Satapathy, Iti Chaturvedi, Erik Cambria, Shirley S Ho, and Jin Cheon Na. Subjectivity detection in nuclear energy tweets. *Computación y Sistemas*, 21(4):657–664, 2017.

[126] Klaus R Scherer. What are emotions? and how can they be measured? *Social Science Information*, 44(4):695–729, 2005.

[127] Hinrich Schütze, Christopher D. Manning, and Prabhakar Raghavan. *Introduction to Information Retrieval*, volume 39. Cambridge University Press Cambridge, 2008.

[128] Rai B. Shamantha, Sweekriti M. Shetty, and Prakhyath Rai. Sentiment analysis using machine learning classifiers: Evaluation of performance. In *The Institute of Electrical and Electronics Engineers (IEEE) 4th International Conference on Computer and Communication Systems (ICCCS)*, pages 21–25. IEEE, 2019.

[129] Nurul Fathiyah Shamsudin, Halizah Basiron, Zurina Saaya, Ahmad Fadzli Nizam Abdul Rahman, Mohd Hafiz Zakaria, and Nurulhalim Hassim. Sentiment classification of unstructured data using lexical based techniques. *Jurnal Teknologi*, 77(18), 2015.

[130] Ramalingam Shanmugam. Practical text analytics: Maximizing the value of text data: by murugan anandarajan, chelsey hill and thomas nolan, switzerland, ag, springer press, springer nature, isbn: 978 -3-319-956663-3, 2019.

[131] Musa Shikur. A hybrid sentiment classification for amharic book reviews. Master's thesis, St. Mary's University, 2019.

[132] C Sindhu, Binoy Sasmal, Rahul Gupta, and J Prathipa. Subjectivity detection for sentiment analysis on twitter data. In *Artificial Intelligence Techniques for Advanced Computing Applications*, pages 467–476. Springer, 2021.

[133] Jaspreet Singh, Gurvinder Singh, and Rajinder Singh. Optimization of sentiment analysis using machine learning classifiers. *Human-centric Computing and information Sciences*, 7(1):32, 2017.

[134] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, 2013.

[135] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer, 2019.

[136] Maite Taboada. Sentiment analysis: An overview from linguistics. *Annual Reviews*, 2016.

[137] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307, 2011.

[138] Yodit Teshome. Sentence level opinion mining for amharic language. Master's thesis, Debre Birhan University, 2019.

[139] Tulu Tilahun. Linguistic localization of opinion mining from amharic blogs. *International Journal of Information Technology & Computer Sciences Perspectives*, 3(1):890, 2014.

[140] Abinash Tripathy, Ankit Agrawal, and Santanu Kumar Rath. Classification of sentimental reviews using machine learning techniques. *Procedia Computer Science*, 57:821–829, 2015.

[141] Miniybel Tsegaw. Sarcasm detection for amharic text. Master's thesis, Bahir Dar University, 2020.

[142] Peter D. Turney and Michael L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *Association of Computing Machinery (ACM) Transactions on Information Systems (TOIS)*, 21(4):315–346, 2003.

[143] Peter D Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188, 2010.

[144] Clara Vania, Moh. Ibrahim, and Mirna Adriani. Sentiment lexicon generation for an under-resourced language. *International Journal of Computers and Applications*, 5(1):59–72, 2014.

[145] Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. The viability of web-derived polarity lexicons. In *Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 777–785. Association for Computational Linguistics, 2010.

[146] R. Hevner Von Alan, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *Management Information System (MIS) quarterly*, 28(1):75–105, 2004.

[147] Daniel Waegel. A survey of bootstrapping techniques in natural language processing. *University of Delaware. Newark*, 2003.

[148] Yun Wan and Qigang Gao. An ensemble sentiment classification system of twitter data for airline services analysis. In *The Institute of Electrical and Electronics Engineers (IEEE) International Conference on Data Mining workshop (ICDMW)*, pages 1318–1325. IEEE, 2015.

[149] Gang Wang, Jianshan Sun, Jian Ma, Kaiquan Xu, and Jibao Gu. Sentiment classification: The contribution of ensemble learning. *Decision support systems*, 57:77–93, 2014.

[150] Negessa Wayessa and Sadik Abas. Multi-class sentiment analysis from afaan oromo text-based on supervised machine learning approaches. *New Media and Mass Communication*, 2020.

[151] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, pages 347–354, 2005.

[152] Wondwossen Mulugeta Wondwossen Philemon. A machine learning approach to multi-scale sentiment analysis of amharic online posts. *HiLCoE Journal of Computer Science and Technology*, 2(2).

[153] Michał Woźniak, Manuel Graña, and Emilio Corchado. A survey of multiple classifier systems as hybrid systems. *Information Fusion*, 16:3–17, 2014.

[154] Yunqing Xia, Xiaoyu Li, Erik Cambria, and Amir Hussain. A localization toolkit for sentic net. In *The Institute of Electrical and Electronics Engineers(IEEE) International Conference on Data Mining Workshop*, pages 403–408. IEEE, 2014.

[155] Baye Yimam. *(-)yäamarIña säwasäw*. Educational Materials Production and Distribution Enterprise (EMPDE), 2000E.C /2007 G.C./.

[156] Lei Zhang and Bing Liu. Aspect and entity extraction for opinion mining. In *Data Mining and Knowledge Discovery for Big Data*, pages 1–40. Springer, 2014.