Sentence level Sentiment Analysis of Afaan Oromoo text-based from social media using Deep Learning

By

Lelisa Bushu Teso

A Thesis Submitted to the department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September 28, 2022

Adama, Ethiopia

Sentence level Sentiment Analysis of Afaan Oromo text-based from social media using Deep Learning

Lelisa Bushu Teso

Advisor. T. GopiKrishna (Ph.D.)

A Thesis Submitted to the department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September 28, 2022

Adama, Ethiopia

Approval Board of Examiner

We, the undersigned, members of the Board of Examiners of the thesis by Lelisa Bushu Teso have read and evaluated the thesis entitled "**Sentence-level Sentiment Analysis of Afaan Oromo text-based from social media using Deep Learning**" and examined the candidate during the open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science and Engineering.

_____ ___ ____

Chairperson

_____ ___ ____

Internal Examiner

_____

External Examiner

Finally, approval and acceptance of the thesis are contingent upon the submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

Department Head	Signature	Date
School Dean	Signature	Date
Office of Postgraduate Studies	Dean Signature	Date

Recommendation

I/we, the major advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled "**Sentence level Sentiment Analysis of Afaan Oromo text-based from social media using Deep Learning**" prepared under my/our guidance by Lelisa Bushu Teso submitted in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science and Engineering. Therefore, I/we recommend the submission of a revised version of the thesis to the department following the applicable procedures.

Dr. T. GopiKrishna	_____	_____
Major Advisor	Signature	Date

Approval sheet of Master's Thesis

I/we, the advisors of the thesis entitled "**Sentence level Sentiment Analysis of Afaan Oromo text from Social media using Deep Learning**" and developed by Lelisa Bushu Teso, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Dr. T. GopiKrishna	_____	_____
Major Advisor	Signature	Date

Declaration

I hereby declare that this Master thesis entitled "**Sentence level Sentiment Analysis of Afaan Oromo text-based from social media using Deep Learning**" is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation

Lelisa Bushu Teso	_____	_____
Name of student	Signature	Date

ACKNOWLEDGMENTS

First, I want to express my gratitude to the Almighty God for giving me full of health throughout my journey. And finally, I want to thank my Advisor Dr. T. GopiKrishna, for pointing me in the correct direction, responding to timely feedback, providing constant assistance during my research, and making helpful suggestions about how to complete the thesis requirements. Next, I want to express my gratitude to the entire lecturers and staff at Adama Science and Technology's Department of Computer Science and Engineering. Next, I want to thank my parents for all of their help along the way; without them, I wouldn't be where I am now. Additionally, I want to thank all of my friends and classmates for their unwavering support during my journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

AI Artificial Intelligence

BBC: Britain Broadcasting Corporation

Bi-LSTM: Bidirectional Long Short-Term Memory

CBOW: Continuous Bag of Words

CNN: Convolutional Neural Network

DNN: Depp Neural Network

FBC: Fana Broadcasting Corporation

GB: Gradient Boosting

LR: Logistic Regression

LSTM: Long Short-Term Memory

MNB: Multinomial Naïve Bayes

NLP: Natural Language Processing

OBN: Oromia Broadcasting Network

ODP: Oromo Democratic Party

RF: Random Forest

SA: Sentiment Analysis

SVM: Support Vector Machine

TF-IDF: Term Frequency Inverse Document frequency

VOA: Voice of America NB: Naïve Bayes

Word2Vec: Word-to-Vector

ABSTRACT

The main objective of this research is to develop sentence-level sentiment analysis for Afaan Oromoo text using deep learning models. Finding out how using Emojis with Afaan Oromo text affects the labeling dataset and model's performance is the other task of the research. Sentiment analysis is a field of NLP that classifies the feeling of people into positive and negative sentiments. *The data (comments and feedback) that was presented on social media are a list of unorganized and unclassified raw data that made it challenging to accurately and quickly ascertain customer sentiment in real-time. To overcome these problems, this study assessed different techniques from data collection to model building. Data that contain Emojis and text was collected from Facebook public pages of OBN, FBC Afaan Oromo, BBC Afaan Oromo, and VOA Afaan Oromo, as well as the Vision Entertainment and Aanaa Entertainment YouTube channels. And this study, was prepared and preprocessed using various techniques of NLP and* proposed CNN, LSTM, and BiLSTM models with word2Vec feature extractions. According to the experiment's findings, using Emojis alongside Afaan Oromo text decreased the performance of the LSTM, BiLSTM, and CNN models from 73.34% to 72.03, 72.88 to 71.88%, and 74.58% to 72.66% respectively with skip-gram feature extractions. This study also evaluates the proposed models on a binary dataset and achieves an accuracy of 89.60% on LSTM, 88.32% on BiLSTM, 87.52% on CNN by using skip-gram, and 87.68% on LSTM, 87.36% on BiLSTM, and 88.64% on CNN with CBOW. Finally, by comparison of all the proposed models, the CNN model outperforms the other two models with an accuracy of 74.58% on the multi-class datasets with skip-gram feature extractions. Therefore, this study chose the CNN model for Afaan Oromo text-based Sentence level sentiment analysis classifications.

Keywords*: sentiment analysis, sentence level, deep learning, CNN, LSTM, BiLSTM, Afaan Oromoo, NLP, classification.*

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the Study

In the case of the spread of the Internet around the world, huge reviews and opinions can be forwarded on different social media platforms and the way people express their feeling and share information was changed. A vast quantity of user-generated data is available online due to the expanding scope of the internet and social media. Identifying and categorizing people's opinions as positive, negative, or neutral has become critical in a variety of fields such as business services, political issues, economic issues, and social activities.

People are expressing their opinions/emotions on various social media platforms such as Facebook, YouTube, Twitter, Instagram, Telegram, and others. According to the number of users and ease of use, Facebook and YouTube are the most popular social networking platforms[1]. People express their feelings on Facebook by posting and replying to comments in text, emoji, audio, and video formats. Online opinions include a varied range of themes, including government, organizations, politics, entertainment, products, and others, and they want to know what their customers think about their services.

Sentiment analysis is a field of NLP and data mining that investigates the feelings, responses, and judgments on different products and services (Bhatia, Sharma, and Bhatia 2018). Businesses frequently use sentiment analysis to classify sentiment in social media data, assess brand reputation, and better understand customers. It is also used to identify opinions from text, emojis, audio, and videos. Sentiment analysis examines a text's polarity (positive, negative, or neutral), as well as specific feelings and emotions (angry, happy, sad, etc.), urgency (urgent, not urgent), and intentions (interested v. not interested). The central idea of sentiment analysis is to classify the polarity of comments and reactions by eliminating the features and components mentioned in each text (Dave, Lawrence, and Pennock 2003).

[1] https://gs.statcounter.com/social-media-stats/all/ethiopia

Depending on how you interpret customer feedback, sentiment analysis can be classified in a variety of ways. The most common sentiment analysis types are graded sentiment analysis, emotion detection, and multilingual sentiment analysis. Graded sentiment analysis is applied if the polarity correctness is significant to the selected business. This type of sentiment analysis considers expanding the polarity and classifies the classes into very positive, positive, neutral, negative, and very negative. Emotion detection sentiment analysis detects emotions other than polarities, such as happiness, frustration, anger, and sadness. Many emotion detection algorithms make use of lexicons or complex machine learning algorithms. Multilingual sentiment analysis needs more resources and preprocessing.

Sentiment analysis is essential for people to express their thoughts and feelings. As a result, sentiment analysis is quickly becoming an essential tool for monitoring and understanding sentiment in all types of data. Automatically analyzing the client's feedback, on social media conversations, enables products to learn what makes customers joyful or unsatisfied, allowing them to tailor products and services to their customers' needs[2]

When sentiment analysis tries to categorize the feeling of people into its polarity rate, there are some challenges like the use of different writing styles; one word can be positive and negative in another, and in one sentence reviewers express positive and negative feelings which goes to a sarcastic sentence in sentiment analysis.

Sentiment Analysis is taken into consideration from a different perspective. Most commons are document level, sentence level, and aspect level. The sentiment analysis at the document level is the way of calculating the overall opinion of a sentence by assuming that every document has a single feeling result (Fikre, 2020). The sentiment analysis at the sentence level directly deals with what people like or dislike about each entity feature in words or sentences (Duwairi 2015). This level of analysis is extracting the appropriate aspects of the commented entity and classifying the sentiment of the corresponding opinion. Sentence-level sentiment analysis is another common sentiment analysis that focuses on the extraction of people's emotions at the sentence or phrase level. It is a finer analysis of the document. In this level, polarity is

[2]https://monkeylearn.com/sentimentanalysis/#:~:text=Sentiment%20analysis%20(or%20opinion%20mining,feedback%2C%20and%20understand%20customer%20needs.

determined for each sentence as an individual sentence is considered an isolated unit and each sentence may have a different opinion. The main task of sentiment analysis at the sentence level is subjectivity classification (Fikre, 2020). The subjective sentence holds a polarity of information either positive or negative. While the objective sentence holds fact information which is neutral sentiment or no opinion (Duwairi 2015).

Many pieces of research have been done for rich resource languages like English, Arabic, and other European languages. Though, sentiment analysis of Afaan Oromo languages did not get more attention. There are a limited number of researches have conducted on Afaan Oromo sentiment analysis using machine learning approaches (Kuyu 2021), (Rase 2020), (Wayessa and Abas 2020), (Oljira and Kekeba 2020). The reasons why many researchers did not consider on Afaan Oromo language may include morphological complexity, lack of resources, lack of availability of labeled data, less existence of research resources, absence of abbreviation rule, and other factors.

There are many problems with Afaan Oromo language sentiment analysis that should not be answered yet. Some of them are there is no available standard dataset to develop Afaan Oromo text-based sentiment analysis in different areas such as politics, economics, entertainment, sports, hotel data services, and transport area of data. Some of the studies focused only on the political data to classify the sentiment analysis. Since, sentiment analysis has abroad area, solving the problems differently is not advisable because it needs many tools and models when it is considered specifically.

We decide to conduct a sentiment analysis of the Afaan Oromo language after identifying the research gaps through a review of related studies. Based on the gaps in prior research, we choose LSTM, BiLSTM, and CNN deep learning models to build sentiment analysis at the sentence level of Afaan Oromo text-based on Facebook and YouTube social media. We also assess the effects of combining textual data with Emojis on the labeling data and the proposed model performance.

To accomplish our goal, this study focuses on a variety of techniques, from identifying the data source to developing the model that solves problems under investigation. This study used social media sites like Facebook and YouTube to gather data. The selection of those social media was

made in light of a study of social media usage trends in Ethiopia over the previous twelve months. The investigation showed that the chosen media outlets have a sizable following of subscribers and users[3].

Data was gathered from various Facebook public pages of VOA Afaan Oromo, FBC Afaan Oromo, OBN, and BBC Afaan Oromo as well as, Vision Entertainment and Aanaa Entertainment of YouTube channels. The collected data are prepared and preprocessed using the NLP preprocessing techniques. This study proposed the LSTM, BiLSTM, and CNN neural models with Word2Vec feature extraction. In the end, this study classified the classes of the sentiment analysis sentient as negative, very negative, neutral, positive, and very positive classes.

1.2. Motivation for the Study

Choosing to work a research on Afaan Oromo sentiment analysis motivated us by several considerations. To begin with, Afaan Oromo sentiment analysis is becoming increasingly important due to its already large-scale audience. In addition, the complication of the language in terms of both morphology and structure has created many challenges. Furthermore, because of its history, the strategic importance of its people, the territory they inhabit, and its cultural and literary heritage, the Afaan Oromoo language is both demanding and fascinating.

The previous researches on Afaan Oromoo have limitations such as considering only political data, classifying the sentient into binary classes, and conducting on traditional machine learning approaches. The availability of gaps in Afaan Oromo Sentiment analysis and the challenges of the targeted Language motivated us to develop Sentiment analysis for Afaan Oromo text-based using Deep learning models. In conclusion, by considering the aforementioned list above, we put some contributions to the selected language.

1.3. Statement of the Problem

Numerous well-known governmental and non-governmental organizations publish daily updates on their public Facebook pages about a variety of topics, including politics, the

[3] https://gs.statcounter.com/social-media-stats/all/ethiopia

economy, entertainment, and on other areas. The comments and feedback left on social media pages are a list of unorganized, unclassified raw data that do not provide the desired insight. Without knowing the customer's opinions, it would be difficult for governmental and non-governmental organizations to improve the standard of their services. Because of this, categorizing those comments without the aid of a sentiment analysis model makes it difficult to determine the precise sentient of the client.

Previous Afaan Oromoo language sentiment analysis researchers focused on texts for comment labeling and opinion classification. They did not give attention to emojis and different areas of data other than political data (Kuyu 2021), (Wayessa and Abas 2020), (Oljira and Kekeba 2020), and (Rase 2020). The other limitations of the previous research were conducted using traditional machine learning techniques and focusing on binary classes and three classes for sentiment classifications (Oljira and Kekeba 2020). The issue with using only these classes is that they do not accurately reflect people's opinions. This means that an opinion that should be assigned to the ‘very positive’ category has been crammed into the ‘positive’ category and the same for every negative and negative class.

As far as the knowledge of the researcher is concerned, there is no system developed for Afaan Oromo language on CNN, BiLSTM, and LSTM models at sentence-level sentiment analysis and also considering Emojis with Afaan Oromo text to evaluate its effects on the model performance.

1.4. Research Question

To solve the issues stated in the Statement of Problems, this study formulates three research questions. Here are the research questions:

- RQ1: Which feature extraction produces better performance for Afaan Oromo sentence-level sentiment analysis with the proposed deep neural networks?
- RQ2: Which deep learning models efficiently produce a better classification model for sentence-level Afaan Oromo sentiment analysis?
- RQ3: What is the effect of using Emojis on model performance when it is added with Afaan Oromo text?

1.5. Objectives of the Study

1.5.1. General Objective

The general objective of this research is to develop a Sentence-level Sentiment Analysis for Afaan Oromo text using deep learning.

1.5.2. Specific Objectives

The following are specific objectives of this study:

- Reviewing the papers on Sentiment analysis
- Collecting textual data and Emojis from selected social medias
- Preprocessing data using NLP data preprocessing techniques
- Build the proposed deep learning models for Afaan Oromo text SA
- Evaluate the proposed model's performance in the development of Afaan Oromo text-based sentence-level sentiment analysis.
- Select the most efficient deep learning models that develop Afaan Oromo text at sentence-level sentiment analysis based on experimented results.

1.6. Significance of the Study

This study assists organizations or individuals those of using Afaan Oromo Language by groping out the feeling of the customers accordingly. This helps to easily understand what our customer thinks about our products/services without reading all comments. Facebook and YouTube social media also advantageable by satisfying their users and if the user is satisfied by the service what the Facebook and YouTube giving, they prefer those social medias than the others. In conclusion, the target groups who benefited from this study was Facebook and YouTube media, governmental or non-governmental organizations, and individual those who use Afaan Oromo language.

1.7. Delimitation of the Study

1.7.1. Scope of the Study

The scope of the research is to develop sentiment analysis for Afaan Oromoo text at a sentence level and also to identify the effect of using Emojis written with text on the labeling data and

model performance measures using deep learning techniques. In this work, we focused on sentiment analysis of opinions of text formats, and evaluate the effect of Emojis added to Afaan Oromo text. The data was collected from the Facebook public pages of OBN, VOA Afaan Oromo, FBC Afaan Oromo, BBC Afaan Oromo, YouTube channels of Vision Entertainment, and Aanaa Entertainment. This study proposed three deep learning models such as CNN, BiLSTM, and LSTM with word2Vec feature extractions on the 7404 labeled datasets. In the end, the polarity checking was categorized as very Positive, Positive, Negative, very Negative, and neutral.

1.7.2. Limitation of the Study

This study conducted on deep learning models that needs large number of datasets. Due to constraints of time, we prepared and labeled 7404 comments which is not fit the proposed models. This may reduce the performance of the models due to small number of the datasets. This study focused on Afaan Oromo texts and assess the effects of using Emojis on labeling data and model performance. So, this study did not consider sarcastic sentiments, and Emojis to classify the polarity of the comments.

1.8. Thesis Organization

The general introduction to the topic is explained in the first chapter. This study makes an effort to examine the problem identified, the motivation for this study, the research question used to address the problem, the scope, and limitations study. A review of the literature and related works is included in Chapter two. The purpose of discussing tasks and literature review is to learn from one another's efforts, find supporting evidence, and identify any gaps. This study's methods and resources are described in chapter three. Data collection, preprocessing of the gathered data, dataset preparation, feature extraction selection, and deep learning classification algorithms are among the stages of this study's methods in this chapter. The design and applications of the suggested models are described in chapter four. The overall research architecture, proposed data preprocessing, proposed feature extraction, and proposed deep learning models and their implementations were the main topics to be covered in this chapter. The experiment, results, and discussions from this study are covered in Chapter five. The evaluation outcome determines which model is the best and answers the questions accordingly. Finally, the conclusion and future works are covered in the concluding section.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORKS

2.1. Overview of the Chapter

The chapter discusses relevant reviews of sentiment analysis were reviewed for further understanding and investigation of the research gaps. Sentiment analysis is a process of emerging a computerized system for extracting opinions from sources such as text, emojis, audio, and videos. Sentiment analysis identifies the polarity of the sentiment to different classes.

Like machine learning, there are four most common approaches of deep learning to develop automated sentiment Analysis systems. These are Supervised, Unsupervised, Semi-supervised, and Reinforcement approaches. Supervised Learning is the learning process considered under the watchful eye of observation variables (Wang et al. 2021). It is a method of developing artificial intelligence (AI) in which a computer program is trained on labeled input data for a certain output. The model is trained until it recognizes the underlying patterns and correlations between the input data and the output labels, allowing it to produce appropriate labeling results when presented with previously unseen data. Supervised learning proficiency in classification and regression challenges, such as determining the category of a news article belongs.

The supervised learning in neural network models is improved by regularly monitoring the model's outputs and fine-tuning the system to get closer to its target accuracy. The level of accuracy obtained is determined by two factors: the available labeled data and the algorithm used. In addition, supervised learning needs the data for training must be balanced and cleansed, the variety of the data impacts how effectively the algorithm performs with new cases reducing the risk of the model will falter and fail to produce accurate results[4].

Classification and regression are the two main outcomes of supervised learning systems. Based on the labeled data it was trained on, a classification algorithm attempts to arrange inputs into a specific number of categories or classes. Regression algorithms anticipate that the model will provide a numerical relationship between the input and output data.

[4] https://www.techtarget.com/searchenterpriseai/definition/supervised-learning

The bias and variance that exist within the algorithm, as well as the complexity of the model or function that the system is attempting to learn, should be considered when selecting a supervised learning method. To achieve acceptable performance levels in supervised learning, enormous amounts of accurately labeled data are required.

Unsupervised learning is defined as having only input data and no associated output variables. This type of learning approach seeks to understand the underlying structure or distribution of data to learn more about it. The data points are used as references to uncovering significant structures and patterns in the observations. It is often used for data exploration, extracting generating features, and finding significant patterns and groupings. Clustering is an unsupervised learning problem that attempts to identify intrinsic groupings in data, for as clustering customers based on purchasing behavior. Another unsupervised learning problem is an association, which indicates a rule learning problem in which rules describe large portions of considered data, such as people who buy X also tend to buy Y. When supervised learning is required but there is a paucity of quality data, semi-supervised learning may be the best option. Semi-supervised learning has been shown to produce reliable results and applies to many real-world applications where supervised learning algorithms would fail due to a lack of labeled data.

Reinforcement learning is a training method that rewards desired behaviors while punishing undesirable ones. A reinforcement learning agent, can understand and interpret its environment, act, and learn through trial and error. Following on the basics of machine learning techniques explained above are drawn in Figure 1 below. This study was worked out using the classification of supervised deep learning methods.

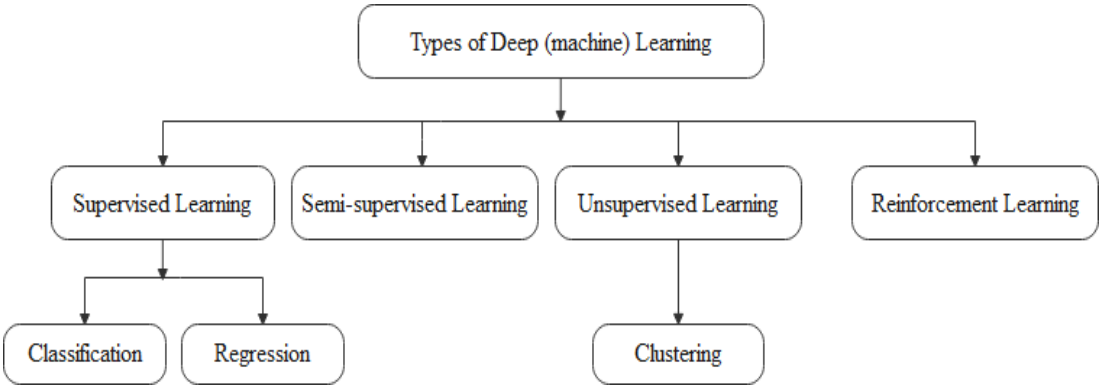


Figure 1: Deep Learning Techniques Types (Rajbanshi Sabita 2021)

2.2. Levels of Sentiment Analysis

Sentimental analysis is a polarity decision concerning the target object at different levels of sentiment analysis. The most common sentiment analysis levels are document level, sentence level, and Aspect based. All types of levels are discussed below.

2.2.1. Sentence level Sentiment Analysis

Sentimental analysis at the sentence level works with the classification of subjectivity which classifies the sentence given as a subjective or objective sentence (Fikre, 2020). A subjective sentence expresses personal feelings, views, emotions, or beliefs which carry a category of positive or negative polarity, whereas an objective sentence presents some factual information.

2.2.2. Document-level Sentiment Analysis

This approach extracts sentiment from the full review and classifies an entire opinion depending on the overall sentiment of the opinion holder. When a document is created by a single individual and reflects a sentiment about a single entity, document-level classification works best. It classifies the whole sentence as positive, negative, or neutral.

2.2.3. Aspect-based Sentiment Analysis

Aspect-based sentiment analysis (ABSA) is a text analysis technique that divides data into aspects and determines the sentiment associated with each aspect (Bethiha, 2021). Customer opinion can be analyzed using aspect-based sentiment analysis, which associates specific feelings with different characteristics of a product or service. This type of sentiment Analysis can extract through sentiment and aspects. Sentiments are discussed with positive or negative feelings concerning a specific topic, whereas Aspects describe a category, characteristic, or topic being discussed.

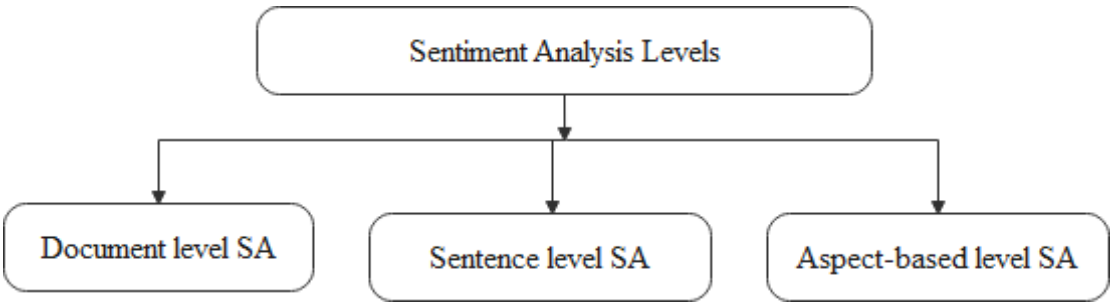


Figure 2: Level of Sentiment Analysis (Shelar and Huang 2018)

2.3. Sentiment Analysis Approaches

Different researchers studied sentiment analysis using various approaches. The common approaches are Lexicon (rule-based), machine learning-based, and hybrid approaches (machine learning and Rule-based). The deep learning approach of sentiment analysis is the recent method and can deeply understand the data than the machine learning approach. This approach needs a large amount of data with data quality to give a good model performance. Below, each approach is outlined.

2.3.1. Lexicon-based sentiment Analysis approach

In Lexicon-based sentiment analysis techniques, a sentiment dictionary with sentiment words is used to classify sentiment. To work with, it requires a set of facts or data sources and rules to manipulate that data. Rule-based sentiment analysis is an approach to text analysis without training or the use of machine learning models. The result of this approach is a set of rules by which text is labeled as positive, negative, or neutral. This approach is also called the lexicon-based approach. The most commonly used lexicon-based approaches are SentiWordNet, TextBlob, and VADER.

Examples of how rule-based works: define the polarity of words (positive words as Bareeda, Baay'ee Bareeda, dansa; and negative words as fokkisaa, gadhee, yaraa, etc), compute the number of positive and negative texts that exist in the sentence, and decides that as positive if the number of positive words occurs are greater than negative words; the proposed model predicts positive sentiment and vice versa. If not the model reflects neutral sentiment (Tesfaye 2011).

2.3.2. Machine Learning-based Sentiment Analysis Approach

Unlike Lexicon-based sentiment analysis, machine learning sentiment analysis use models to determine the sentiment of sentences. Many research was conducted on sentiment analysis using machine learning algorithms such as Logistic Regression (LR), Naïve Bayes (NB), Support Vector Machine (SVM), Maximum Entropy (ME), Random Forest, and others (Sachdeva, Abhishek, and V.K 2019).

2.3.3. Hybrid approach of Sentiment Analysis (ML and Lexicon-based)

A hybrid approach system works with the collaboration of machine learning and Lexicon-based

techniques to deliver a good result. The result acquired from this method is always more precise. The paper (Vaitheeswaran and Arockiam 2016) has shown that the combination provides improved performance in sentiment classification. As this study stated, sentiment lexicons are used as resources and detect sentiment polarities, and the results from the lexicon-based are used to train machine learning algorithms (Vaitheeswaran and Arockiam 2016).

2.3.4. Deep Learning approach of Sentiment Analysis

Deep learning methods are a type of machine learning that uses deep neural networks to learn (Ain et al., 2017). Using several layers of networks is one of the applications of artificial neural networks (ANN) for learning tasks. Deep Neural Networks are useful in a variety of fields, including sentiment analysis, speech recognition, language translation, information retrieval, and health care. Deep learning algorithms can classifyclassifytiment analysis with higher accuracy and performance.

From many deep learning algorithms, CNN (Convolutional Neural Network), RNN (Recurrent Neural Network), Long Short-Term Memory (LSTM), and Bidirectional Long Short-Term Memory (BiLSTM) are the most prevalent deep neural networks (Ruangkanokmas et al., 2016). The algorithms utilized in this investigation were CNN, LSTM, and Bi-LSTM.

2.4. Feature Extraction Techniques

The technique of turning raw data into numerical features that can be processed while keeping the information in the original data is known as feature extraction. Compared to directly using deep learning on the raw data, it produces better outcomes. The preprocessed data should be converted to a format that the neural network classifier can understand. TF-IDF, a bag of words, and word embedding are the three main techniques for word representation. The explanation of those feature extracts is discussed below.

2.4.1. Bag of words

The bag of words model takes raw data as input and outputs the number of times each word appears in a document. In the Bag of Words paradigm, word lists are matched with the number of times they appear in a document (Philippe 2016). For developing a documented vocabulary that is used to classify a document, a bag of words may employ Unigram, Bigram, Trigram, or N-gram (Philippe 2016). The ways for constructing a vocabulary include Unigram, Bi-gram,

Trigram, and N-gram. Each word is treated as a token in Uni-gram, while N-gram treats N pairs of words as a token. N might be 1, 2, 3, or any other number

2.4.2. Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is another method for transforming raw data into numerical features that can be fed into any machine learning algorithm to find hidden subjects. TF-IDF focuses on the relevance of words rather than their frequency count because a high-frequency count does not indicate that the document has a lot of information. As a result, the TF-IDF transformation is superior to the bag of words model (Philippe 2016)

2.4.3. Word Embedding

The most well-known used word representation strategy in NLP applications is word embedding, which allows words with similar meanings to have comparable representations (Philippe 2016). Word embedding is a type of feature learning that seeks to map words from a real-number vocabulary into a low-dimensional space.

For capturing both syntactic and semantic information about words, word embedding can be computed. Many NLP tasks, such as sentiment categorization of a document by adding semantic information to the documents, performed well with word embedding as input. Glove, FastText, and Word2Vec are the three most well-known word embedding methods (Philippe 2016).

Glove: a global vector for word representation developed by Pennington et al. for an unsupervised learning approach to obtain vector representation of words (Mandelbaum and Shalev 2016), (Philippe 2016), (van Paridon and Thompson 2021).

FastText: is a c^{++} package for efficient word and sentence classification representations in both supervised and unsupervised learning. Both the continuous bag of words (CBOW) and the Skip-gram models are supported by FastText (van Paridon and Thompson 2021).

Word2Vec: is an efficient word embedding model from raw text developed by Mikolov et al (Shao et al. 2017), (Mandelbaum and Shalev 2016). The word2vec model can be used to solve a variety of NLP challenges, including sentence categorization, sentiment analysis, and other issues. To handle the meaning of the text, word2vec vectors of the words in the sentence are provided to the classifier model as input. Because of the availability of many parameters in the

word2vector model, the performance of NLP applications is improved. During training, pre-trained word2vec does not require any updates that reduce the Overfitting of data.

For learning word embedding, Word2Vec employs a Continuous Bag of Words and Skip-Gram (Mandelbaum and Shalev 2016). Skip-Gram predicts a sentence's context using neighboring words and is more efficient with a small amount of training data, whereas Continuous Bag of Words predicts a single target word's context and is more efficient with a large amount of training data.

2.5. Deep Learning Classification

Deep learning techniques apply a multilayer approach to the neural network's hidden layers. Traditional machine learning approaches define and extract features either manually or using feature selection methods. On the other hand, deep learning models can learn and extract features automatically, resulting in higher accuracy and performance.

2.5.1. Recurrent Neural Network

Artificial Neural Networks include Recurrent Neural Networks as a subset. RNNs are a sort of DNN that uses sequential data to anticipate desired outcomes (Lipton et al., 2015). RNNs make decisions based on what they've learned in the past and what they're learning now. RNN has two types of input data: current and past recent inputs. When RNN makes a judgment, it considers both the data inputs learned in the prior input and the current inputs.

During RNN training, the data is handled in iterative ways, requiring neural network updates to recall the data input for accurate prediction outcomes. RNNs are used to address networks that have loops in them, allowing information to continue to flow.

Even if RNN has the power of predicting the long-term dependencies, the varnishing gradient enforces the RNN not to remember long-term dependencies (Lipton et al., 2015). The chain of loops shows that the RNNs are imitatively related to sequences, and lists and they are the natural architecture of Neural Networks used for such data.

2.5.2. Long Short-Term Memory (LSTM)

Long Short-Term Memory is a special kind of RNN, that is capable of learning long-term dependencies that solve the problem of long-term dependency. LSTM remembers the previous

information and uses it for processing the current input. LSTM contains information in the memory of the gated cell to choose whether to store or delete information (Lipton, Berkowitz, and Elkan 2015), (Understanding LSTM Networks -- colah's blog 2016). LSTM algorithm can store, add, delete, and read information to cell state using three gates such as input gate, forget gate, and output gates on its memory. Input gate to take new inputs, forget gate to remove unnecessary information, and output gate to check the impact of information on the current time step (Understanding LSTM Networks -- colah's blog 2016)

2.5.3. Bidirectional Long Short-Term Memory (BiLSTM)

Bidirectional Long-Short Term Memory (BiLSTM) is the process of constructing a neural network that can store sequence information in both directions (backward and forwards). The information in the traditional RNN and LSTM model can only be propagated forward, the state of time t only depends on the information before time t.

BiLSTM differs from regular LSTM in that the input flows in both directions. The input flow in one direction with a regular RNN and LSTM, either backward or forwards. In Bi-LSTM, however, the input flows in both directions to preserve future and past data.

2.5.4. Convolutional Neural Network (CNN)

CNN is a convolutional layer that extracts information from a larger piece of text, thus we use the convolutional neural network for sentiment analysis, and we create a simple convolutional neural network model and test it on a benchmark. The multilayer perceptron is used in a convolutional neural network. These layers can be entirely integrated or grouped.

Convolutional Feed-forward neural systems are compared to neural networks. Where the neurons can function by gaining a better understanding of points (weights) and predispositions. Each word's statement model is different in CNN. The input data is linked to a vector representation, which is made up of a dimensional vector. Convolutional neural networks also do well in semantic parsing and the detection of paraphrases. They're also used in picture categorization and signal processing.

2.6. Related Work

Many researchers conducted research that related to sentiment analysis in different languages. Some researchers tried to do some research on Afan Oromo sentiment Analysis which predicts the polarity of the emotions of people. Below we tried to list and discuss the work done on Afan Oromo sentiment analysis, and other Languages.

2.6.1. Afan Oromo Sentiment Analysis

2.6.1.1. Rule-based sentiment Analysis for Afan Oromo Language

Jamal Abate (Abate and Million 2019) applied the word lexicon method to unsupervised Afan Oromo sentiment analysis. He conducted three different experiments, considering unigram words as a feature as the first. For unigram feature extraction, he extracted 73% of the features and correctly extracted 79% of the extracted unigram features. In the bigram feature extraction and classification experiment, he used more informative words than in the unigram experiment. At this stage, the system extracted 70% of the features and extracted 82%of the bigram features correctly. In the third experiment, he used trigram feature extraction with more informatory words than unigram and bigram feature extraction with a performance of 66% to extract the usable words.

2.6.1.2. Machine Learning-based Sentiment Analysis for Afan Oromo Language

Negessa Wayessa and Sadik Abas use a supervised machine learning approach to construct a Machine Learning approach to multi-scale sentiment analysis of Afan Oromo text (Wayessa and Abas 2020). The researchers gathered 1810 corpus sizes from OBN tweeters. They employ the Support Vector Machine and Random Forest methods, as well as the TF-IDF feature extraction method. The constructed model achieved an accuracy of 90% for SVM and 85% for RF, respectively.

Sultan Jenbo created an automatic machine learning sentiment analysis from Afan Oromo text (Kuyu 2021). A sentiment analysis at the word, phrase, and sentence levels was performed by a researcher. According to this study, he used a variety of Machine Learning approaches, including Multinomial Nave Bayes, Logistic Regression, Support Vector Machine, and Random Forest with TF-IDF feature extraction. He gathered data from OBN, BBC News Afan Oromo, and FBC Afan Oromo. For MNB, LR, RF, and SVM, the constructed model attained accuracy

measures of 80.12%, 82.52%, 80.62%, and 84.62% respectively. The polarity of sentiment is predicted by three classes (positive, neutral, and negative).

According to the paper (Oljira and Kekeba 2020), researchers worked on sentiment analysis of Afan Oromo text using machine learning techniques. The research focused on document-level sentiment analysis with two polarity classes, namely positive and negative. The collected dataset size is 1452 corpus from ODP Facebook pages. The research uses a Multinomial Naive Bayes algorithm with TF-IDF feature extractions and achieved a performance of 93% accuracy.

The first paper uses a small dataset size which affects the performance of the model and also it is advised for a morphologically large language like Afan Oromo. The third paper predicts the sentiments only in two classes (positive and negative) that indicate the feeling of customers not performing as they needed.

More of the listed above papers use small datasets, though, their accuracy is a good achievement, but the area of their dataset is only politics. So, the trained data is politics and tested by political data the algorithm needs even a small amount of data to produce a good performance. If the area of the dataset is contained different scopes like politics, the economy, Entertainment, sport, and others the model need a very large amount of data and the performance may not be good as a single scope area of the dataset.

2.6.1.3. Deep Learning-based Sentiment Analysis for Afan Oromo Language

The author (Rase 2020) used deep learning algorithms to undertake a character-level Sentiment analysis of Afan Oromo literature. They acquired 2400 corpora from Facebook and Twitter platforms. The model trained differently for Facebook and Twitter data with CNN, Bi-LSTM, and a combination of CNN, Bi-LSTM (CNN-Bi-LSTM), and achieved an accuracy of 93.3%, 91.4%, and 94.1% on Facebook; and an accuracy of 92.6%, 90.3%, and 93.8% on the tweeter respectively. The developed model predicts an opinion based on five classes (2, 1, 0, -1, and -2) that assess sentiment polarity. Since they focus on document-level sentiment analysis, some aspects of customers' emotions are buried; this has an impact on products, services, and other activities.

2.6.2. Sentiment Analysis for Other Language

According to the paper (Kindeneh 2020), this study was conducted on Opinion Mining for Amhara Broadcasting Agency News using Machine learning approaches. The work focuses on sentence-level sentiment analysis and classifying its polarity into three classes called positive, neutral, and negative. The collected dataset size was 787corpus from the Amhara Broadcasting Agency Facebook page and YouTube channels. The method used for this study was Support Vector Machine and Naïve Bayes with TF-IDF feature extraction. As the researcher reported, the result achieved 83.24% and 78% for Support Vector Machine and Naïve Bayes respectively.

According to the paper (Bethiha 2021), the work done for the Aspect-based sentiment analysis model for Hotel services in the Amharic Language uses Machine learning approaches. This study focuses on aspect-level sentiment analysis and classifying its polarity into three classes called positive, neutral, and negative. The amount of data set collected from different hotel pages and websites is 2124 corpus size.

As the researcher reported, this study uses Support Vector Machine, Naïve Bayes, Logistic Regression, and Gradient Boosting Algorithms with unigram, bigram, trigram, and TF-IDF feature extraction. The model achieved an accuracy of 92.2%, 81.3%, 87%, and 92.8% for SVM, NB, LR, and Gradient Boosting respectively. A researcher did not apply to stem and used positive, negative, and neutral classes only.

A rule-based approach for effective sentiment analysis was developed by (Chin -Sheng Yang, and Hsiao-Ping Shih, 2012). The proposed system learns a set of effective rules for product features without training data manually. They collect data from Amazon.com and use the Minimum Support (min-supp), and Minimum Confidence (min-conf) methods to measure the performance of the system.

The author (Wondwossen Mulugeta 2014), delivered a supervised machine learning approach to investigate the multi-scale sentiment analysis model for Amharic online posts written in Ethiopic scripts. The work's main goal is to identify multi-scale sentiment sentences based on the polarity weight value of sentiment words. The author compiled a dataset of 608 social media posts. The author used five polarity rank scales to distinguish the polarity strength of the Amharic language.

To classify sentiment texts, the Naive-Bayes algorithm was trained on the entire corpus. The algorithm acquired knowledge from the training corpus using n-gram features. For the unigram, bigram, and hybrid language models, the author achieved accuracy rates of 43.6 percent, 44.3 percent, and 39.3 percent, respectively. However, the system's accuracy was very low because the training dataset was insufficient (Wondwossen Mulugeta 2014).

Mabrahtu Tedese used the word lexicon unsupervised method to conduct Trilingual sentiment analysis on social media (Amharic, Tigrigna, and English). He used precision, recall, and F-measure to evaluate system performance, and he achieved an average precision of 87.49%, recall of 84.78%, and F-measure of 85.99%

Sentiment mining from opinionated Arabic language combined with semantic orientation and machine learning was developed by Amira (Amira, 2013). Egyptian dialect tweets of 20,000 corpora were collected from Twitter, and those 4800 tweets were annotated manually. She tests a model in different ways; compares the performance of the model by testing the dataset before and after preprocessing, compares the performance of the combined approach against the performance of the approach separately; and evaluates the effectiveness of negation on the performance of the hybrid approach.

All papers were done from different sentiment analysis perspectives (document-level, sentence-level, and aspect-based level). All researchers focused only on texts, and they did not give any attention to Emojis to predict the polarity and also determine the effect of emojis written with the text on the labeling dataset and the performance measures (Wayessa and Abas 2020), (Kuyu 2021), (Oljira and Kekeba 2020). The summary of the reviewed papers is summarized in Table 1 below.

Table 1: Related works Summary

Author	Paper Title	Feature, Method, Accuracy	Gaps
Negessa Wayessa and Sadik Abas	Sentence-level sentiment analysis from Afan Oromo text based on supervised Machine Learning	TF-IDF; 90% of SVM, 89% of RF	did not consider Emojis, consider only political data, this study was done using ML
Sultan Jenbo	Developing an Automated Machine Learning Based Sentiment Analysis for Afan Oromo	TF-IDF, 80.12% MNB, 82.52% LR, 80.62% RF, 84.62% SVM	Emojis are not applied, no overfitting checker, consider only political data, used only 3 classes
Megersa Oljira et.al	Sentiment analysis of Afan Oromo using Convolutional Neural Network Bidirectional Long Short-Term Memory	Word embedding; 93.3% CNN, 91.4%, Bi-LSTM 94.1% CNN-Bi-LSTM	Emojis did not considerable, did not check to overfit, Consider only political data
Mebratu Tadesse	Trilingual sentiment analysis on social media	word lexicons, precision 87.49%, recall 84.78%, and F-score 85.99%	The datasets for three languages were not enough and the technique used is not advanceable for large dataset
Wondwossen Mulugeta	A machine Learning Approach to multi-scale sentiment Analysis of Amharic Online posts	n-grams, 43.6% of Unigram, 44.3% of a bigram, and 39.3% of trigram using NB	Low performance, work done using traditional ML
Gupta, j. Pruthi1,	Sentiment Analysis of tweets using the Machine Learning Approach	67.78% on both KNN, SVM, and	Using traditional ML algorithms, it may not be true for more dataset

and N. Sahu		76.17% on a hybrid of both algorithm	
Jamal Abate	Unsupervised Opinion Mining Approach for Afaan Oromo Sentiments	Word lexicons, bigram performance of 70% recall, 82% precision, and 76% F-score	The technique followed is difficult for a large dataset, and used 600 reviews only
Megersa Oljira	Sentiment analysis of Afan Oromo using Machine Learning (Document-level).	TF-IDF; 93% of MNB	Used binary class, did not consider emojis, worked on traditional ML algorithms, consider only political data

2.7. Afaan Oromo Language

Afaan Oromo is an Afro-Asian language that is mostly spoken by the Cushitic family by the Oromo peoples. Afaan Oromo language is the native tongue of the Oromo people. It is an official working language of the Oromia regional state of Ethiopia and now is recognized as the working language of the federal government including Afar, Somali, Tigrigna, and Amharic languages. Afaan Oromo language is also an academic language starting from primary school up to Ph.D. levels at higher institutions.

According to the Ethiopian Population Census Commission, Oromo people who speak the Afaan Oromo language account for 34.5 percent of the country's overall population (Commission, 2008). It is also the third most widely spoken language in Africa, after Arabic and Hausa languages, according to a paper (Tegegne, 2016). It is a prominent indigenous African language that is extensively spoken and understood in most of Ethiopia and parts of neighboring nations such as Kenya, Tanzania, Djibouti, Sudan, and Somalia.

2.7.1. Afaan Oromo Writing system

Afaan Oromo is a Latin orthography that has a writing script known as Qubee (Ibrahim Bedane 2015). As a result, with some modifications, the language shares many characteristics with English Language writing. The language's Afaan Oromo writing system, known as "Qubee

Afaan Oromo," is simple and is based on the Latin script. Thus, letters in English are also written in Afaan Oromo, except for how they are written. Afaan Oromo is a phonetic language, which means it is spoken in the same way it is written, according to paper (Sari 2017). In contrast to English or other languages.

Qubee contains 33 different characters; among those 26 characters of 21 consonants (Qubee Dubbifamaa) letters, 5 vowels (Qubee Dubbachiiftuu) letters, and 7 compounds (Qubee Dachaa) letters. Qubee was similar to English letters except for consonants like ‘p’, ‘v’, and ‘z’. Some words are employed in Afaan Oromo writing systems that relate to English words. For example, the English word “Police” is written by Afaan Oromo as “Poolisii”.

The writing system of ‘Qubee’ was the official script for Afaan Oromo since 1991. The vowels of Afaan Oromo follow two rules: short (one vowel letter) and long (two vowel letters) which is a maximum and gives a sound when they are combined with consonant letters. For example, consider the following words: lafa (earth) short vowel letters, and laafaa (soft) long vowel letters. Table 2 shows all Afaan Oromo Qubee and its International Phonetic Alphabet (IPA) taken from paper (Kuyu 2021).

Table 2: Qubee Afaan Oromo and its IPA

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

2.7.2. Afaan Oromo Punctuation marks

Afaan Oromo uses punctuation marks to determine the boundary of sentences, which is similar to other Latin languages. But, unlike in the English language, the apostrophe mark in Afaan Oromo is used to represent a glitch (hudhaa) sound (Tesema & Tamirat, 2017). Hudhaa (glitch) is used to write words of two different vowel letters like the word “har’a” to mean in English “today”. Most of the time glitch (hudhaa) in Afaan Oromo is interchangeable with the consonant

letter “h”. For Example, the word in Afaan Oromo like “ta’e”, “ga’e”, and “ba’e” is written as” tahe” (let it be), “gahe” (done), and “bahe” (out). But, not all Afaan Oromo words which contain glitch (hudhaa) are interchange with the letter “h”. For instance, the word “har’a” is not written as “harha” which gives a meaningless word in Afaan Oromo. Some of the most usually used punctuation marks in Afaan Oromo Language are Tuqaa (full stop), Mallattoo Gaaffii (question mark), Raajeffannoo (exclamation mark), Qoodduu (comma), Tuqlamee (colon), etc.

2.7.3. Grammatical Rules of Afaan Oromo

Afaan Oromo Language follows the grammatical word series of the subject-object-verb opposite of the English language which follows the subject-verb-object word sequence(Tesema & Tamirat, 2017). For example, the Afaan Oromo equivalent sentence of “Gadaa came yesterday” is written as “Gadaan kaleessa dhufe”.in this sentence Gadaa (subject), came (verb), and yesterday (object).

2.7.4. Sentiment in Afaan Oromo

The sentiment is the way to determine the feeling of people by considering the context and word tones in the sentence. For example, the “Marmaree jedhama nyaata naannoo Arsiiti” this sentence is a neutral polarity prediction, “Sirba aadaa baayyee sirriitti itti yaadamee hojjetamee fi qulqullinni viidiyoo fi yeedaloo baayyee Bareeda” this sentence is Positive polarity prediction. “Namni sirba Yoosaniif Muuziqaa taphate gahumsa waan qabu natti hin fakkaatu Muuziqaan qulqullina hinqabu” this sentence is a negative polarity prediction. “Ati Nama tahuukee iyyuu nan shakka sammuu doomaa kanaan biyyan hoggana jettee xurumbaa Afuuftu mataa xaasaa goote saree!” this sentence gives very negative polarity prediction. “Sirba baay’ee baay’ee bareedaa fi ajaa’ibaa kana ilaaluu kootti baayyeen gammada” this sentence is very positive polarity prediction.

2.7.5. Challenges of Afaan Oromo in sentiment analysis

There are some difficulties in analyzing and determining Afaan Oromo sentiment text in an automated manner. The most difficult challenge for Afaan Oromo in sentiment analysis and classification is a lack of resources, such as a prepared lexicon database and parts of speech that aid in predicting people's polarity opinions. Another difficulty is that the Afaan Oromo language is morphologically very rich, making it difficult to stem the entire word.

CHAPTER THREE

3. MATERIALS AND METHODS

3.1. Overview of the Chapter

This chapter explains how to predict sentiment analysis using various methods, approaches, tools, and research procedures. The methods described in this chapter were used to accomplish research objectives that addressed the issues mentioned in the statement of problems and provided an accurate response to the research question. The methodology followed the phases of data collection, preprocessing dataset (including annotation of the dataset, tokenization, stop word removal, cleaning dataset, and normalization), feature extraction techniques, model selection methods, and applying various evaluation techniques for assessing the performance of the desired sentiment analysis model. Beginning with section 3.3 all phases are explained.

3.2. Research Design

Research design is the technique that answers the research questions of this study. It is also the key strategy that integrates the process of working research logically to overcome the problem raised in effective ways. Research design includes the type of this study that the research procedure follows. So, this study follows an experimental research type that the result has been performed on an experiment to predict the sentiment analysis. Research design includes the activities such as research area identification, literature review, gap formulation, data collection and preprocessing, applying feature extraction, building the models, and predicting the polarity of the comments. The research design procedures are explained in Figure 3 below.

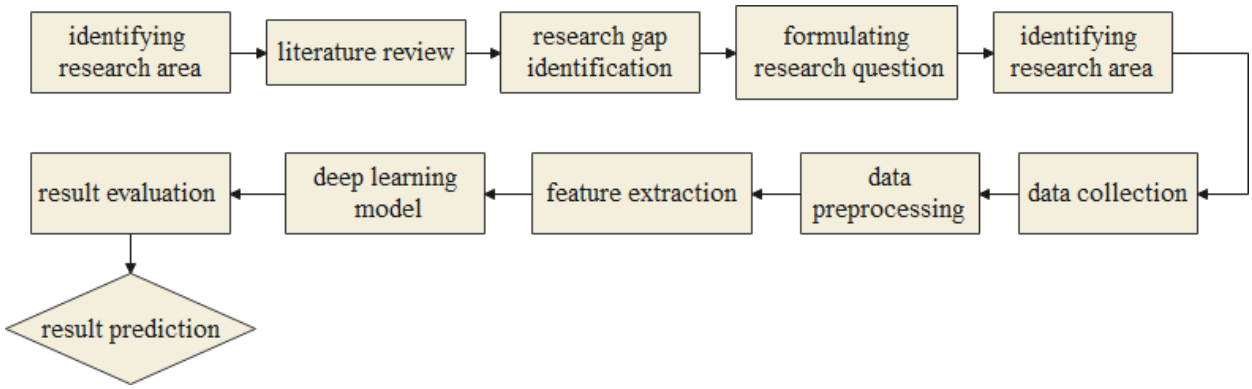


Figure 3: Research Design

3.3. Building Dataset

Since there is no standard dataset for Afaan Oromoo Sentiment Analysis, it needs to build a new Afaan Oromoo Sentence-based Sentiment Analysis dataset. To prepare the dataset we followed the process of Collecting textual and Emoji comments from Facebook and YouTube social media, preparing, cleaning, filtering the collected data, and annotating the dataset.

3.3.1. Data Collection

A researcher can assess their hypothesis based on the data collected. The most critical goal of data collection is to ensure that rich and reliable information is collected for statistical analysis so that data-driven decisions can be made for research. This study collects comments on Afaan Oromoo texts and emojis from Facebook public pages and YouTube platforms using the Face Pager software application and YouTube online comment scrapers respectively.

The reason why we selected Facebook and YouTube was that they have huge followers, subscribers, and the availability of resources. The below chart shows that the selected social media have a greater number of followers than the other social media for the past twelve months.

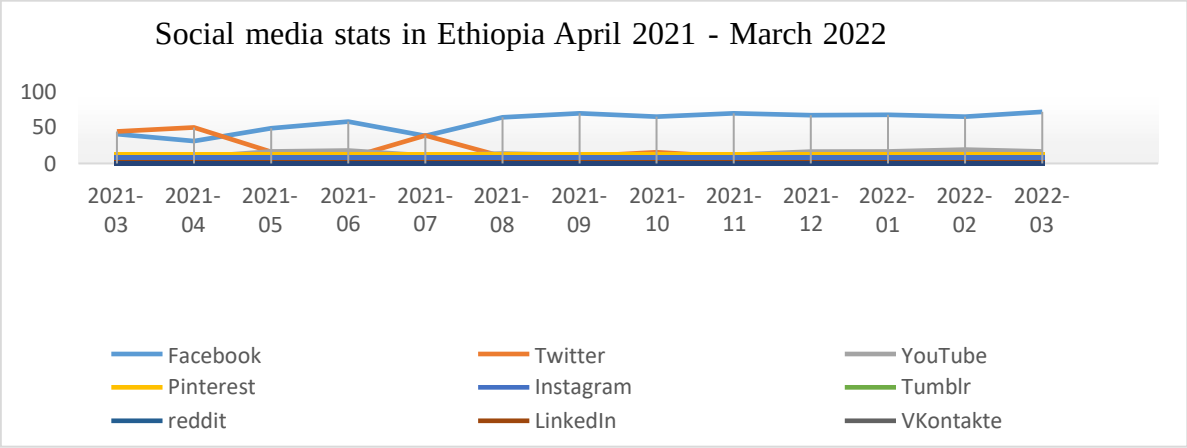


Figure 4: Social media Users Stats in Ethiopia

This study selected some public Facebook pages and YouTube channels. The criteria followed to choose Facebook Pages and YouTube channels are:

- The most popular Facebook public pages that regularly post in Afaan Oromo languages
- Pages with a lot of followers on their pages and channels
- Pages with news about politics, the economy, and entertainment.

Table 3: Shows the number of followers on the selected Facebook pages and YouTube channels

Name of Facebook pages and YouTube channels	Number of followers
BBC News Afaan Oromoo	1,176,953
OBN Afaan Oromoo	1,126,632
FBC Afaan Oromoo	796K
Vision Entertainment	459K
Aanaa Entertainment	315,109

In the above table BBC News Afaan Oromoo, OBN Afaan Oromoo, and FBC Afaan Oromoo are selected from Public Facebook pages. Those pages have a very huge number of followers as well as more opinions are commented on every post whether the information of posts is political or economic. Vision Entertainment and Aanaa Entertainment are selected from YouTube channels for entertainment purposes[5].

Data on politics, the economy, and entertainment were gathered via Facebook and YouTube, respectively. There were 7404 datasets in total, 2023 of which were for entertainment, 2000 for the economy, and 3,381 for politics. The data was gathered from both media between March 20, 2022, and May 2, 2022.

3.3.2. Dataset Preparation

The next step of data collection is the data preprocessing process followed by filtering, cleaning, checking the missing values, and combining the collected data into a single dataset. In preparation techniques, we used Microsoft Excel to prepare the Comma-Separated Values (CSV) dataset. To prepare the dataset for annotation we followed the below criteria:

- All non-Afaan Oromo texts, special characters, and numbers are unconcerned.
- All missing values, whitespace, and blank values are not considered.
- To ensure that each text in the dataset is unique and combining each data into one dataset.

[5] https://gs.statcounter.com/social-media-stats/all/ethiopia

3.3.3. Dataset Annotation

To create a dataset for sentiment analysis at the sentence level, the comments are labeled with their polarity weights. To determine the impact of using a dataset labeled with text and Emoji over just text alone, we have annotated the dataset in this study using labeled text only and text with Emoji.

This study gathered a total of 7,404 comments, all of which were at the sentence level. Text-only dataset and text plus emoji dataset are two different datasets included in the dataset. When annotating text with emoji comments, the annotator takes into account how the emoji will affect the sentence polarity rather than labeling text-only comments by removing the emoji.

To gauge the level of agreement between the two annotators, these collected datasets were divided up into 3,702 different comments for both of them, along with 500 comments that are the same for both annotators. The results of datasets that have text only and text with Emoji were described below within tables. The dataset labeled the comments into five classes such as very positive, positive, neutral, negative, and very negative classes. The guideline for those classes is discussed in Appendix A.

Table 4: Unique annotated dataset

Annotators	Very positive	positive	neutral	negative	Very negative	Total dataset
Annotator 1	338	1099	578	1066	621	3,702
Annotator 2	334	1107	574	1053	634	3,702

Table 5 showed that the common annotated dataset for both annotators. This helps to see the agreement between the annotators.

Table 5: Common annotated dataset

Annotators	Very positive	positive	neutral	negative	very negative	Total dataset
Annotator 1	39	127	109	110	115	500
Annotator 2	42	130	102	117	109	500

The common dataset was the same dataset that was shared for both annotators to identify the inter agreement between the annotators.

Table 6: Total annotated text only and text with Emoji dataset

Polarity	Total comments	text only dataset	text with Emoji dataset
Very negative	1242	995	247
negative	2132	1925	207
neutral	1156	1004	152
positive	2198	1898	300
Very positive	676	582	94
Total	7404	6404	1000

3.3.4. Guideline of Data Annotation

The annotation process is prepared by Linguistic experts. Annotators label each comment into five different classes' namely very positive, positive, neutral, negative, and very negative.

3.3.5. Annotation Procedure

The annotation procedure provided by the researcher and the annotators served as the basis for the preparation of the annotation process. The labeling procedure is involved two annotators. They were chosen from the Oromia State University and Dambi Dollo University College of Social Science and Humanities. Their selection was based on their proficiency in the Afaan Oromoo language. We only had two annotators due to a lack of resources (Budget and time. They randomly select an equal number of reviews, and they promptly submit the annotated data.

3.3.6. Inter Annotator Agreement

The inter-annotator agreement is useful for determining how well a dataset has been annotated. The agreement aids the annotator in determining if a sentient being falls under one or more categories. Cohen's kappa guidelines were employed in this study to gauge the degree of agreement between annotators. We distributed 500 of the same comments to two annotators to gauge the level of agreement between them, and we found a 0.685 value of Cohen's kappa. Therefore, the level of the agreement is substantial. There are some discrepancies among the annotators regarding the dataset's labeling.

Table 7: Sample of disagreement of the annotator

Comments	Annotator 1	Annotator 2
Duulli birmadummaa biyyaa eegsisuu karaa milkaahina qabuun ni raawwatamaa jedhame	neutral	Positive
Dhugaan ni tura malee hinuma baati yoo Waaqayyo jedhe.	neutral	Positive
Jabaadhuu Gadi guuri iccitii gantoota Sabaaf osoo taane warra garaaf jiraattu ergamtoota nafxanyaa.	Very negative	Negative
Jaalala na qabsiifte!	positive	Very positive

3.4. Dataset Preprocessing

Dataset preprocessing is a crucial phase in data analysis that includes a variety of techniques, including cleansing of data, data reduction, and data transformation. This phase assists in cleaning up the acquired data and making it acceptable for analysis.

Emojis, non-text data, and other items written entirely in the text were all eliminated from the dataset because they served no purpose for the Dataset. The emoji on the second dataset, which contains text and emoji, was left in place after the dataset was processed. Because it aids in determining how emojis affect the labeling dataset and the model's performance. Data preprocessing includes the techniques of cleaning, tokenization, stop word removal, Normalization and Stemming. These techniques are discussed below sections.

3.4.1. Tokenization

In Natural Language Processing, tokenization is a typical activity in both traditional NLP approaches and deep learning-based architectures. Tokenization is the process of breaking down a large chunk of text into smaller called tokens. The most common types of tokens are characters, words, and sentences.

Character tokenization is breaking words into characters. For example, the character tokenization of the word “token” is written as t-o-k-e-n. Word tokenization is splitting a sentence into words. For example, the word tokenization of the sentence [“performing tokenization is good for preprocessing dataset”] is tokenized as [‘performing’ ‘tokenization’ ‘is’ ‘good’ ‘for’ ‘preprocessing’ ‘dataset’]. Sentence tokenization is splitting the documents of the sentence into single sentences.

3.4.2. Stop-word removal

Stop words are words that have no significant meaning and are frequently removed from manuscripts. Stop words are removed from a term space to minimize their dimensionality. Articles, prepositions, and pronouns are all considered stop words in text documents. Since a Natural Language Toolkit (NLTK) does not recognize Afaan Oromoo stop words, we had to create our own stop words.

3.4.3. Normalization

Normalization is one of the procedures involved in obtaining clean data from unstructured textual data. Normalization is a technique for lowering the number of unique tokens in a text and eliminating variances; as well as tidying up the text by deleting unnecessary information. From two kinds of normalization such as character level and word level normalization, our study used word normalization. For instance, from the sentence “Baayyeeeeen jaalladhe Diraamaa keessan jabaadhaa!” the word “Baayyeeeeen” need normalization to give a correct meaning in Afaan Oromo. So, the word “Baayyeeeeen” was corrected to “Baayyee”. Finally, the sentence was corrected as “Baayyeen jaalladhe Diraamaa keessan jabaadhaa!”.

3.5. Feature Extraction Methods

The fundamental issue with working with language processing is that deep learning algorithms cannot work directly on raw data. As a result, feature extraction techniques are required to turn text into a matrix (or vector) of features. We used word2Vec as feature extraction for our study.

3.5.1. Word Embedding

From word embedding, word2Vec is used to handle the meaning of the text by preparing vectors of the words in the sentence to the classifier model as input. Word2vec is not a deep neural network, it turns text into a type of computation that deep neural networks can understand. Word2vec's goal and benefit are to collect vectors of the same words in vector space. Word2vec generates vectors based on numerical representations of word elements, such as individual word context.

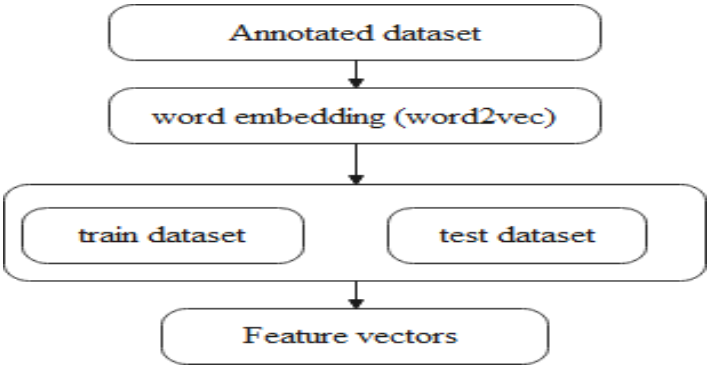


Figure 5: Feature Extraction process

3.6. Model Selection Techniques

In sentiment analysis, several deep learning techniques are employed. Before comparing different models, there is no definitive answer to say which one will perform the best for the particular challenges. A list of appropriate methods for Afaan Oromoo text sentiment analysis is provided in this study.

One reason is that a classification technique may be necessary depending on the nature of the issue. A multi-class classification problem that separates the reviewer's sentiment is necessary for the suggested sentiment analysis of Afaan Oromoo's text.

Another reason is to train the model with labeled data after creating a labeled dataset whose input and output are already known, then test it with unlabeled data. To forecast the sentiment analysis, supervised deep learning approaches were therefore suggested for this study.

The final justification for choosing a particular model is that deep learning algorithms excel at NLP tasks. Additionally, deep learning algorithms produce high-performance results using high-quality and substantial data. The deep learning algorithm that performs best should be chosen from those being employed.

3.7. Deep Learning models for sentence-level Sentiment Analysis

Deep learning algorithms are used in a variety of fields, including speech recognition, sentiment analysis, and healthcare. Deep learning also evolved in terms of Natural Language Processing. For this study, we chose the LSTM, BiLSTM, and CNN deep learning models. The discussion of the algorithm is listed below.

3.7.1. Long Short-Term Memory

Long Short-Term Memory is a type of RNN that can learn long-term dependencies and so solves the problem of long-term dependency. The past information is remembered by the LSTM and used to process the current input. LSTM has information in its gated cell memory that allows it to determine whether to store or discard information (Lipton et al., 2015). (Colah, 2015). The Step to develop LSTM network modules are:

Select the information to be deleted: - Decide the information to be removed from a cell using the sigmoid layer of the forgetting gate layer.

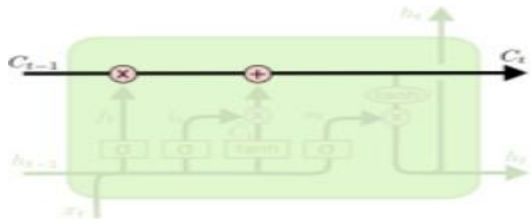


Figure 6: Forget layer of LSTM network (Understanding LSTM Networks -- colah's blog 2016)

$t = \sigma(\text{Wf}.[\text{ht} - 1, \text{xt}] + \text{bf})$ (1) (Understanding LSTM Networks -- colah's blog 2016)

Select new information to store: -decide what new information to store in the cell using the input gate layer and the tanh layer creates new candidate (Ct) values that are added to the state. Figure 103 shows how information goes in the input and tanh layers.

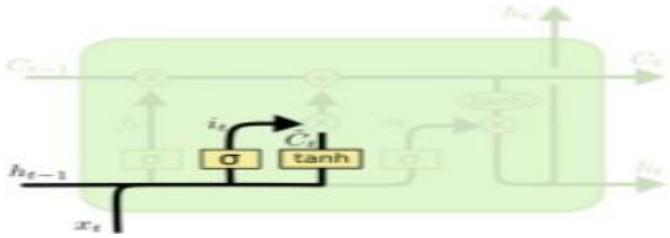


Figure 7: Input and tanh layer of LSTM network (Understanding LSTM Networks -- colah's blog 2016)

$it = \sigma(\text{Wi}.[\text{ht} - 1, \text{xt}] + \text{bi})$ (2) (Understanding LSTM Networks -- colah's blog 2016)

$Ct = \tanh(\text{WC}.[\text{ht} - 1, \text{xt}] + \text{bC})$ (3) (Understanding LSTM Networks -- colah's blog 2016)

Update the old cell state to the new cell state: - multiply the new cell state by what you forgot earlier, and add the new candidate cell state to the old cell state. The new candidate values are

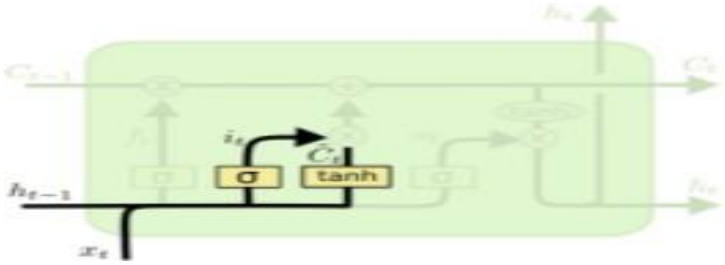


Figure 8: - Memory cell update layer of LSTM network (Understanding LSTM Networks -- colah's blog 2016)

$Ct = \text{ft} * \text{Ct} - 1 + \text{it} * \text{Ct}$ (4) (Understanding LSTM Networks -- colah's blog 2016)

Decide the Output:- select the output based on the cell state. The sigmoid layer decides what part of the cell state going to output and multiplies it with an output of the sigmoid layer (Understanding LSTM Networks -- colah's blog 2016).

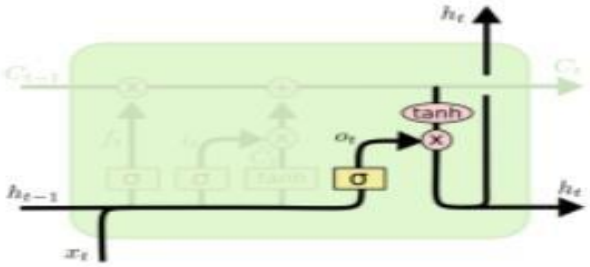


Figure 9: Output layer of LSTM network (Understanding LSTM Networks -- colah's blog 2016)

$Ot = \sigma(\text{Wo} . [\text{ht} - 1, \text{xt}] + \text{bo})$ (5) (Understanding LSTM Networks -- colah's blog 2016)

$ht = \text{Ot} * \tanh(\text{Ct})$ (6) (Understanding LSTM Networks -- colah's blog 2016)

3.7.2. Bidirectional Long-Short Term Memory (BiLSTM)

BiLSTM is another neural network model for this study to predict the sentiment analysis classification of Afaan Oromoo text. BiLSTM follows how LSTM works. The shortcomings of the LSTM model, which accesses future information, are being fixed by the BiLSTM model. The idea behind the BiLSTM model is to connect two separate LSTMs and send a signal across

each of their layers in the opposite direction so that it can flow in both directions (backward and forward). Figure 10 demonstrates how the BiLSTM algorithm functions.

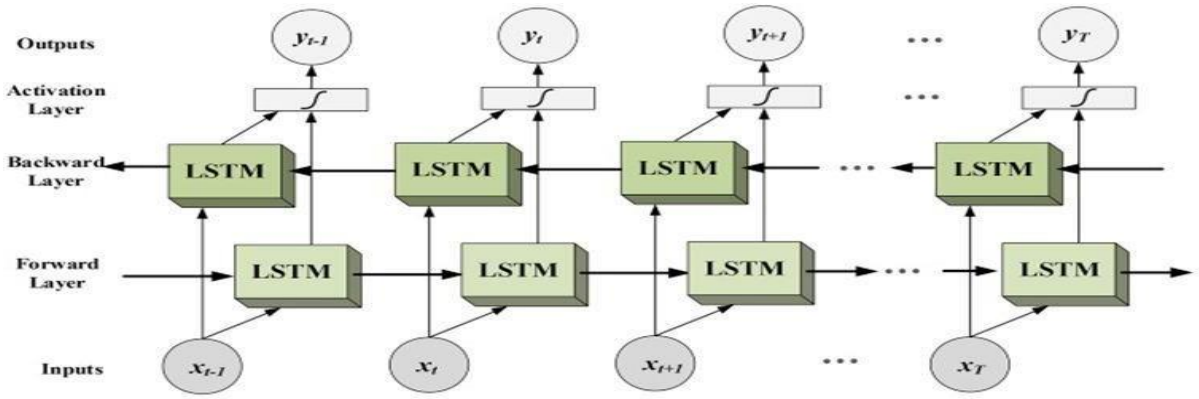


Figure 10: Bidirectional Long-Short Term Memory Architecture

The information flow in the diagram is backward and forwards. When sequence-to-sequence tasks are required, BiLSTM is used. Text classification, speech recognition, and forecasting models all use this type of neural network.

3.7.3. Convolutional Neural Network (CNN)

A particular deep learning approach called a convolutional neural network was created primarily for image processing. These days, CNN is also capable of doing NLP tasks, particularly in text categorization and sentiment analysis (Yao et al. 2016).

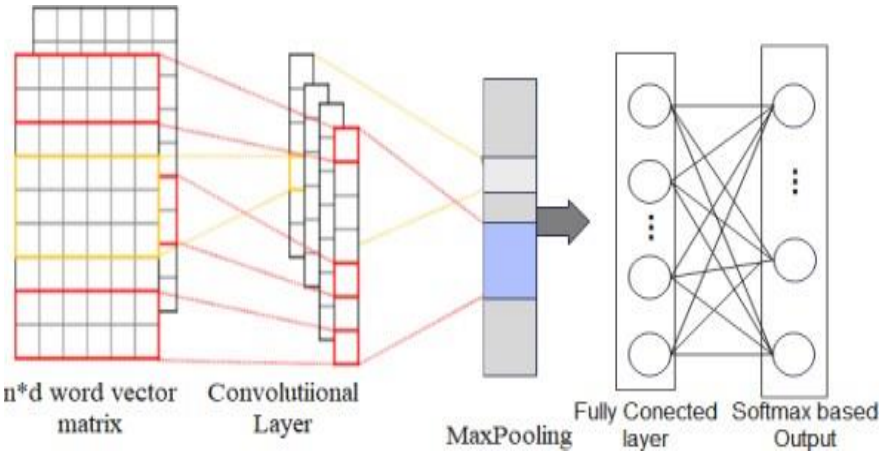


Figure 11: Architecture of CNN model (Yao et al. 2016)

From Figure 11 above, MaxPooling carries out the network to hold only the maximum value in a feature vector. The 1D convolutional layer shows kernel flow in one direction only.

3.8. Model Overfitting Techniques

Overfitting is by far the most prevalent issue with neural networks. Overfitting occurs when a model is created that performs well on training data but struggles to generalize to unobserved data. Another way of occurring of overfitting is when the dataset is inadequate, and the model's design grows complex. Deep learning model overfitting can be fixed using many methods, the most popular of which are regularization approaches, cross-validation, data argumentation, and early stopping. For this investigation, dropout approaches were chosen to address overfitting and underfitting. The rationale for using dropout techniques is to minimize training time and generalize error loss (Kurtis, 2021).

3.9. Model Evaluation Metrics

Evaluating how well the categorization model predicts the anticipated outcome is essential while developing and improving the model. A key component of creating an efficient deep learning model is model evaluation. Accuracy, precision, recall, and F-score are the four categories of evaluation metrics that are used most frequently. The important points listed below should be understood while contemplating the evaluation metric.

- True Positive (TP): when both actual class of data point and predicted are True
- True Negative (TN): when both actual class of data point and predicted are False
- False Positive (FP): when the actual class of data point was False and the predicted is True
- False Negative (FN): when the actual class of data point was True and the predicted is False

Accuracy: denotes the rate at which the method properly predicts results. The predicted value of accuracy was calculated as equation 7.

$$Accuracy = \frac{TP + TN}{total\ instance\ (TP + FN + TN + FP)} \qquad 7$$

Precision: a metric that shows the predicted values indicate how many times the model correctly predicts that value. The precision value was determined as equation 8.

$$Precision = \frac{TP}{TP + FP} \quad 8$$

Recall: the fraction of genuine positives correctly identified. Its prediction value was determined as equation 9.

$$Recall = \frac{TP}{TP + FN} \quad 9$$

F-score: sometimes known as F1-score, is a test accuracy metric that is defined as the weighted harmonic mean of the precision and recall of the test. When the dataset has an unequal class distribution, F1 is more useful than accuracy. The F-score prediction value was calculated as equation 10.

$$F - Score = 2 * \frac{Recall * precision}{Recall + precision} \quad 10$$

3.10. Tools and Materials

We used various tools and packages to implement sentiment analysis of Afaan Oromoo text. Some of the tools and packages are listed below

Table 8: software, hardware, and python libraries

Tools	Description
Environments	
Anaconda Navigator	It is a GUI, which makes it easy to launch applications, packages, and environments without using the command line
Jupyter notebook	It is an open-source web application to create and share documents that contain code, equations, and text. It is also used to clean data, data transformation, statistical modeling, and machine learning

Google Colab	To write and execute arbitrary python codes via browser and helpful for machine learning
Programming language tools	
Python (3.9.7)	Python is an Object-Oriented, high-level programming language rich in semantics. It is a powerful programming language to develop a machine learning to process Natural Languages easy to learn
Data collection tools	
Facepager (4.3.10)	To collect data from Facebook
YouTube online comment scraper	To collect data from YouTube
Data preparation tools	
Microsoft Office (2019)	For writing a documentation
Excel (2019)	For dataset preparation tasks and sorting of the collected data
System design tools	
Wondershare EdrawMax (10.5.2)	Software diagram tools that assist to design diagrams. The software built-in editable symbols and templates. This study used this software to draw system design, the architecture of this study, the level of SA, etc.
Some of the Packages used	
NLTK	A Python programming platform for working with human language data and applying statistical Natural Language Processing.

Pandas	Is fast, powerful, flexible, and easy to use. Used to clean and analyze data. It includes features for investigating, cleaning, changing, and visualizing data. Pandas used to read dataset
NumPy	A multi-size array and matrix data structures are included in this package. It can be used to perform a variety of mathematical operations on arrays.
Tensorflow (6.4.5)	is an open-source that includes a library for performing various machine learning tasks including building and training models with high performance
Regex (re)	A regular expression is a good tool for analysis of natural language processing and it defines a set of strings that matches it, performing string matching, replace, removal, etc. we used this tool for preprocessing of text.
Matplotlib	matplotlib is a python plotting library that is widely used for machine learning applications because of its numerical mathematics extension- NumPy to create static, animated, and interactive visualization.
Keras	is a high-level neural network library that runs on top of TensorFlow. Used for building and training models.
Gensim	It is used for modeling document indexing and retrieving similarities with large datasets. Gensim is used to construct the word2Vec model
Computer	Installed and work with Preprocessor: Intel(R) Core (TM) i7-7500U CPU @ 2.70GHz 2.90 GHz, RAM: 8.00 GB, system type: 64-bit operating system, x64-based processor, windows 10 pro.

CHAPTER FOUR

4. DESIGN AND IMPLEMENTATION

4.1. Overview of the Chapter

In this chapter, we discuss the Design and implementation of the proposed Sentiment analysis of Afaan Oromoo text using different deep neural networks such as CNN, LSTM, and BiLSTM. This study focuses on an experimental approach to select the best experimental result of the listed neural networks above with word embedding feature extractions.

4.2. Architecture of the Proposed Model

The architecture of research is a crucial tactic that unifies the method of conducting research logically to efficiently address the issue raised. This study contains all the processes including identification of data sources, Data collections, data annotation, data preprocessing, feature extractions, and deep learning models. The system uses several methods and components that increase the model's effectiveness and efficiency. Figure 12 below describes the overall system architecture for automatic SA of Afaan Oromoo text.

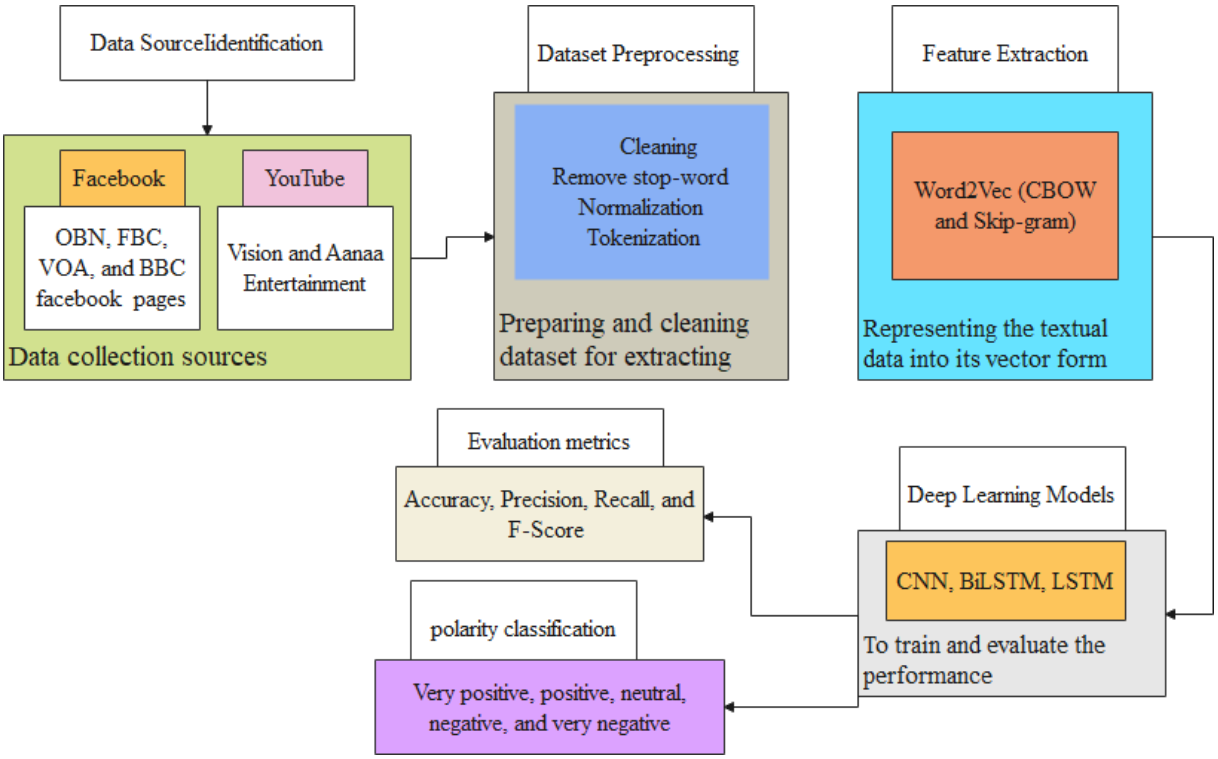


Figure 12: Architecture of the proposed model

4.3. Proposed Model Building

The features obtained from the feature extraction techniques of word2Vec are used to train algorithms on training data sets. Long-Short Term Memory (LSTM), Bidirectional Long-Short Term Memory (BiLSTM), and convolutional neural networks (CNN) are some of the deep neural network models that were chosen for this study.

All of the listed models were deep learning models, which can evaluate the data with high quality and accuracy, which is why those models were chosen. With the ability to accept a sequence of data and use multiple memories to recall previous inputs, LSTM is advantageous for comprehending the context of texts. The main benefit of using the LSTM algorithm is its capacity to retain past important information while eliminating all other irrelevant data.

To produce high accuracy and automatically identify key features, CNN prefers to use image data because our dataset includes emojis. By training two LSTMs on the input sequence of data, BiLSTM can perform better on problems involving sequence classification. Every element of the input sequence contains information from both the past and the present, so by combining LSTM layers from both directions, BiLSTM can produce more insightful outputs.

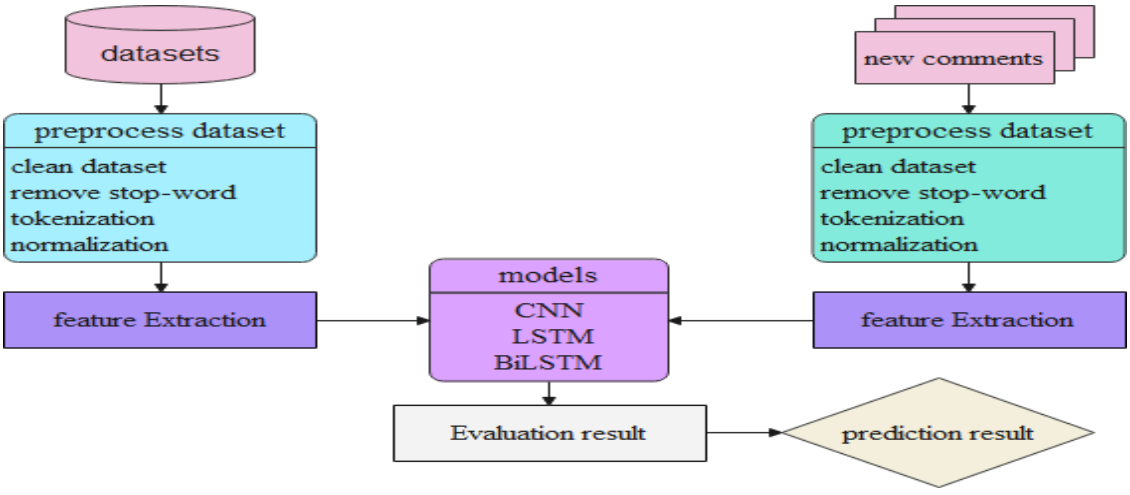


Figure 13: Model Building of Afaan Oromo text sentiment analysis

4.4. Proposed Data Preprocessing

Social media provides unstructured data that is full of extraneous information. Therefore, working with data and organizing it efficiently for deep learning models is essential. The data collected from Facebook and YouTube social media need the NLP preprocessing techniques to

clean the collected comments for the preparation of a quality dataset This study included a variety of preprocessing techniques including cleaning the dataset, removing stop words, tokenization of words, and normalization. The listed NLP preprocessing techniques are discussed below.

4.4.1. Data cleaning

Data cleaning is the process of eliminating unnecessary elements, special characters, white space, and other uncleaned information. it makes data clean to prepare the quality of the dataset and assists the models to deliver high-performance results.

```
INPUT: uncleaned dataset
 OUTPUT: cleaned dataset
 INIT: Read the dataset
 WHILE (it is not the end of the dataset):
    if data contains punctuations, numbers, special characters, white spaces, and other symbols ['0123456789','-','/',
      '(', ')','@','+','#', '&', '*','$','%','!',',','.','?',’|','/','a','aa','A','ka','b','B',’c','C','d','D','E','e',
     'F','e','G','g','H','h','ph','PH','TS','ts', 'I','i','j','J','K','k','L','l','M','m','n','N','o','O','P','p',
     'ch','CH','NY','ny','q','Q','r','R','s','S','t','T','I','U','u','v','W','V','w','y','Y','Z','z','J', ' ',]
     THEN remove symbols, numbers, punctuations, white spaces, special characters
     return cleaned dataset
ENDIF
HALT
```

4.4.2. Stop-word removal

Stop words are words that have no significant meaning and are frequently removed from manuscripts. Stop words are removed from a term space to minimize their dimensionality. Articles, prepositions, and pronouns are all considered stop words in text documents. Since a Natural Language Toolkit (NLTK) does not recognize Afaan Oromoo stop words, we had to create our own stop words.

```
INPUT: dataset
OUTPUT: dataset free from stop-words
INIT: Read the dataset
WHILE (dataset):
IF dataset not in afaanoromoostopwords:
Return dataset
ENDIF
HALT
```

4.4.3. Normalization

Normalization is one of the preprocessing techniques that follow the procedures involved in obtaining clean data from unstructured textual data. Normalization is a technique for minimizing the number of unique tokens in a text and eliminating variances; as well as tidying up the text by deleting unnecessary information.

Normalization is also applied to elongated words and abbreviations to make words meaningful. For instance, in the sentence “Ani baayyeen si jaalladhaaaa!” the word “jaalladhaaaa” need normalization to give a correct meaning in Afaan Oromoo. So, the word “jaalladhuuuu” was corrected to “jaalladha”. Finally, the sentence was corrected as “Ani baayyeen si jaalladha!”.

```
INPUT: unnormalized text
OUTPUT: normalized text
INIT: Read the text
WHILE (it is not the end of dataset):
IF text contains [umrii] THEN normalize into “umurii”
IF text contains [xoophiyaa] THEN normalize into “itiyoophiyaa”
IF text contains [titiisa] THEN normalize into “tisiisa”
Return normalized text
ENDIF
HALT
```

4.4.4. Tokenization

Tokenization is the process of breaking down a large chunk of text into smaller called tokens. The most common types of tokens are characters, words, and sentences. The proposed data preprocessing of Afaan Oromoo sentiment analysis is explained below.

The data preprocessing techniques are shown in Figure 14. After the dataset was fully preprocessed feature extraction was applied to the dataset to transform the textual data into its vector representation.

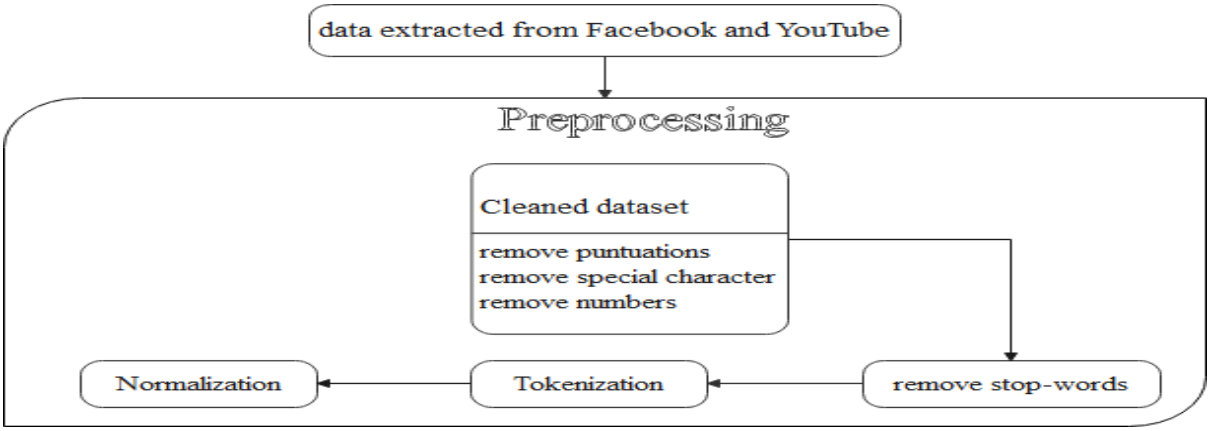


Figure 14: Data Preprocessing of Afan Oromo text sentiment analysis

4.5. Proposed Word Embedding (word2Vec)

The textual data is converted into vectors through word embedding, which comes after the dataset has been cleaned and preprocessed. The word2vec technique is used in this study

because it can comprehend the semantics of the words used in the sentence. It is also recommended for deep learning techniques to give good results by scaling the dataset.

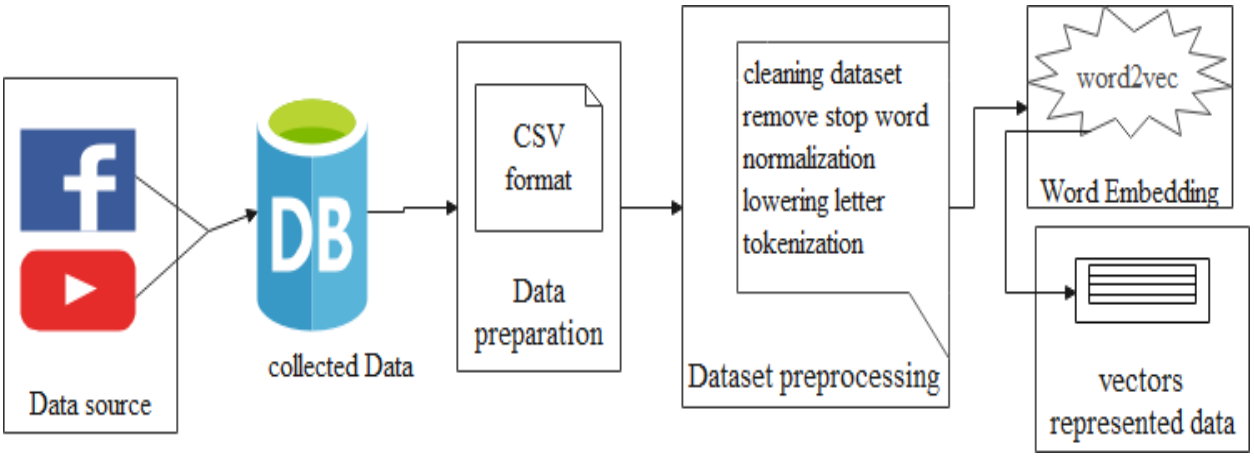


Figure 15: Proposed Data preprocessing and Feature Extraction

4.6. Proposed Deep Learning Model

For this investigation, deep learning models including CNN, LSTM, and BiLSTM were suggested. Those models have been proposed by the section 3.6 criteria. The decision to choose various models was made because each of them has distinct drawbacks. For instance, the LSTM model is restricted from learning the sentence order without access to future data. To determine the best model for Afaan Oromoo sentiment analysis, as shown in Figure 16 below, this study tested the model using new data and evaluated it using evaluation metrics.

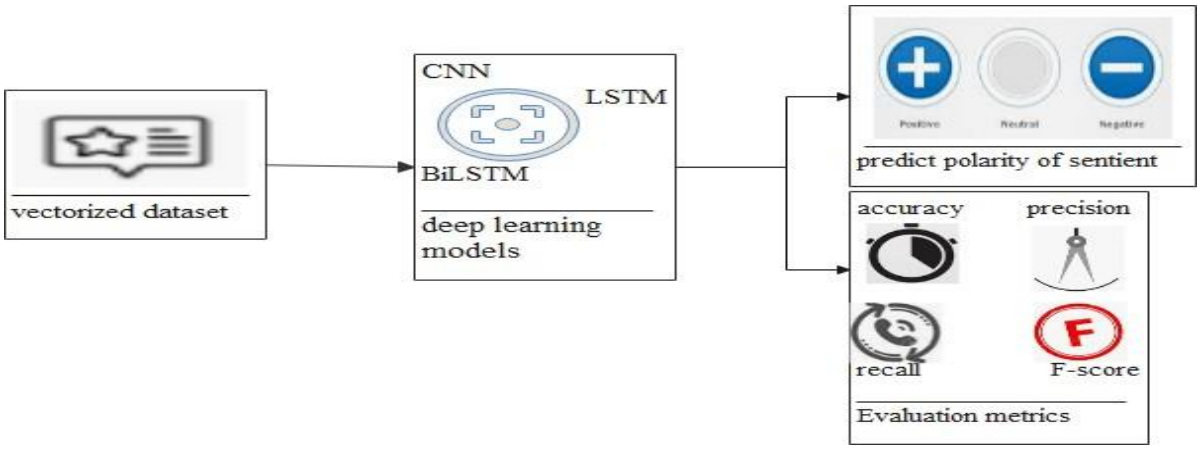


Figure 16: proposed deep learning models for Afaan Oromo SA

4.7. Implementation of Proposed Data Preprocessing

4.7.1. Loading Dataset

The python panda's library utilized the pd.read_CSV () method of CSV formats to load the dataset. The dataset in pandas' format is loaded, as evidenced by the sample code below.

```
df=pd.read_csv('input.csv', header = 0, encoding='unicode_escape')
```

4.7.2. Cleaning the Dataset

Data cleaning is accomplished by eliminating various things like punctuation, digits, white spaces, and special characters. The cleaning dataset was displayed in the sample codes below.

```
def clean_text_round(text):
    text=re.sub('\[.*?\]', '',text)
    text = re.sub(r'\s+', ' ', text).strip()
    text = re.sub('([@0-9_]+)|[^\w\s]|#|http\S+', '', text)
    return text
round1= lambda x:clean_text_round(x)
df1=pd.DataFrame(df.reviews.apply(round1))
```

4.7.3. Normalization

Normalization is important to reduce the number of unique tokens in a text, get rid of variations, and clean up the text by getting rid of extraneous details. The following provides evidence for the sample codes used to normalize the dataset.

```
def clean(text, stem_words=True):
text = re.sub(" mini", " miti ", text, flags=re.IGNORECASE)
text = re.sub(" baay'ee ", " baayyee ", text, flags=re.IGNORECASE)
text = re.sub(" ta'e ", " tahe ", text, flags=re.IGNORECASE)
text = re.sub(" ba'e", "bahe", text, flags=re.IGNORECASE)
text = re.sub(" ka'e", " kahe ", text)
text = re.sub("yuunbarsiitii", " yuunvarsiitii ", text, flags=re.IGNORECASE)
text = re.sub("umrii", " umurii ", text, flags=re.IGNORECASE)
text = re.sub("bilxiginnaa", " badhaadhina ", text, flags=re.IGNORECASE)
data['reviews'] = data['reviews'].apply(clean)
```

4.7.4. Tokenization

Tokenization is used to break up phrases into words known as tokens once the dataset has been cleaned, normalized, and stop words have been eliminated.

4.7.5. Removing stop words

Stop words are unnecessary in sentiment sentences and their removal helps to minimize the dimensionality. Below is a sample of code for deleting stop words and tokenization words.

```
from nltk.corpus import stopwords
stop_words=set(stopwords.words('afaanoromoo.txt'))
from nltk.tokenize import word_tokenize
def remove_stopwords(sentence):
    word_tokens = word_tokenize(sentence)
    clean_tokens = [w for w in word_tokens if not w in stop_words]
    return clean_tokens
df['reviews'].apply(remove_stopwords)
```

4.8. Implementation of proposed Word Embedding

Word Embedding assists in converting textual data into a vector format that the models can comprehend. The word2Vec feature extraction sample code is shown in the code below.

```
import gensim
from gensim.models import Word2Vec
from gensim.models import FastText
from gensim.models import KeyedVectors
Min_count=2, Size=100, Workers=4, Window=5
model=gensim.models.Word2Vec(sentences=stopw,vector_size=Size,sg=0, workers=Workers,min_count=Min_count, window=Window)
model.build_vocab(stopw, progress_per=1000)
words=list(model.wv.key_to_index)
```

4.9. Proposed Model Implementation

Vector represented dataset was utilized to train the model and develop the deep learning methods. This study proposed various including CNN, BiLSTM, and LSTM. The proposed models' source code was included in Appendix I.

4.9.1. LSTM Model Implementation

Two hidden layers were used in the LSTM model for sequential data. To produce the findings, the model was trained with 150 filters, 0.2 dropouts, activation relu, and dense 128. Here is a sample LSTM model program.

```
modells = Sequential()
modells.add(embedding_layer)
modells.add(LSTM(128,  return_sequences=True, activation = 'relu' ))
modells.add(Dropout(0.25))
modells.add(LSTM(64,  return_sequences=False, activation = 'relu'))
modells.add(Dropout(0.2))
modells.add(Dense(6, Activation ='softmax'))
```

4.9.2. BiLSTM Model Implementation

BiLSTM employed two LSTM hidden layers for both directions to better understand the context of past and future sentences. Here is a sample of BiLSTM model codes.

```
modells = Sequential()
modells.add(embedding_layer)
modells.add(Bidirectional(LSTM(128,  return_sequences=True, activation = 'relu' ))
modells.add(Dropout(0.25))
modells.add(Bidirectional(LSTM(64,  return_sequences=False, activation = 'relu'))
modells.add(Dropout(0.2))
modells.add(Dense(6, Activation ='softmax'))
```

4.9.3. CNN model Implementation

Layers with convolutions Conv1D with 64 filters, the Adam optimizer, a maximum dropout of 0.25, maximum pool size 1, dense of 128, and the relu activation function were employed in this study to construct the CNN model. Below is a sample of the code for the CNN model.

```
modelcnn = Sequential()
modelcnn.add(embedding_layer)
modelcnn.add(Conv1D(64, 1, 1, input_shape=(8, 1, 1), activation='relu'))
modelcnn.add(Conv1D(32, 1, 1, activation='relu'))
modelcnn.add(Dense(6, activation='softmax'))
```

CHAPTER FIVE

5. RESULTS AND DISCUSSIONS

5.1. Overview of the Chapter

This chapter explains, using an experiment, the impact of utilizing emojis on a labeling dataset and the model performance outcomes, and also the impact of proposed deep learning models for Afaan Oromo text on sentence-level sentiment analysis. This study analyzes and contrasts the outcomes of several proposed deep learning models that are offered in this study with those from other studies that were carried out on Afaan Oromo sentence-level sentiment analysis, which is another crucial point depending on the dataset that was created. The examination of how the presented study responds to the research questions is the chapter's main topic.

5.2. Results

Based on experiments, various deep learning models were used to show this study's findings. Emojis used within text comments also have an impact on model performance and dataset labeling. The results of each model, the effects of increasing the number of classes, and the effect of Emojis are explained below sections.

5.2.1. The effect using of Emojis on Labeling Afaan Oromo texts

This study experimented to find out the effect of Emoji on labeling Afaan Oromo sentiment analysis. As we discussed in the introduction part Emojis are considered to be reliable indicators of sentiment to enhance sentiment classification. This study concluded that an exploratory experiment to show the labeled Emojis with text on the dataset and the number of labeled comments with class

Table 9: The of effect using Emojis on dataset distribution

Comments	Number of comments
Comments are written with text only	6404
Comments are written with text and Emojis	1,000
Total	7, 404

From Table 9, the total number of comments collected and labeled in the dataset was 7,404 Among the total numbers 14.3 % were the dataset collected from text with Emojis. There was a very huge number of comments replied to on the selected Facebook pages and YouTube channels with Emojis only. Since our target was identifying the effect of Emojis that are used with texts, we did not consider only Emoji comments. A large number of Emoji comments shows that a large number of reviewers use Emojis to express their feeling.

Using Emojis for comment labeling affects the polarity of a sentient and affects the number of comments classified as positive, neutral, and negative. For instance, the word "Mootummaa naannoo Oromiyaa" in English "Oromia regional government" can be labeled as neutral, but if the Emoji is added to the text the class of the text was changed. So, the word "Mootummaa naannoo Oromiyaa ❤❤" is classified as positive since it expresses positive sentiment. In the same way the sentence "sirba haaraa Andualem Gosa gadhiifame" is classified as neutral, but the sentence "Sirba haaraa Andualem Gosa gadhiifame ❤🤦‍♀️🤦‍♀️" can be grouped as a positive class. According to the above example, Emojis have the power to change the class of the comments.

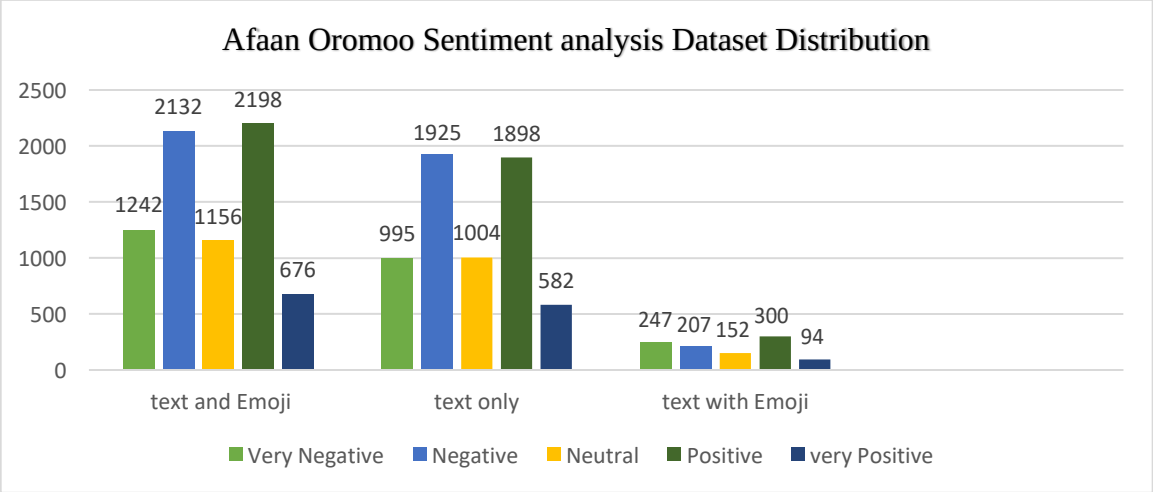


Figure 17: Afaan Oromoo Sentiment Analysis Dataset Distribution

From Figure 17, the number of labeled comments as very positive, positive, neutral, negative, and very negative were different on text and Emoji, text only, and text with Emojis. The chart shows that from a total of 7,404 labeled comments, 6,404 were text-only labeled comments, and 1000 were text with Emoji comments. This shows that Emojis play a significant role in data labeling.

5.2.2. Result of LSTM Model

The LSTM model was built with a neural network of hyperparameters that were trained with the prepared dataset. The model achieved an accuracy of **73.73**% and **70.31**% on text only dataset and text with Emoji dataset respectively with a CBOW and an accuracy of **72.03**% and **71.66**% on text only dataset and text with Emoji dataset with skip-gram feature extraction. The below figure showed the accuracy and loss of training and validation of the LSTM model on the multi-classes (five).

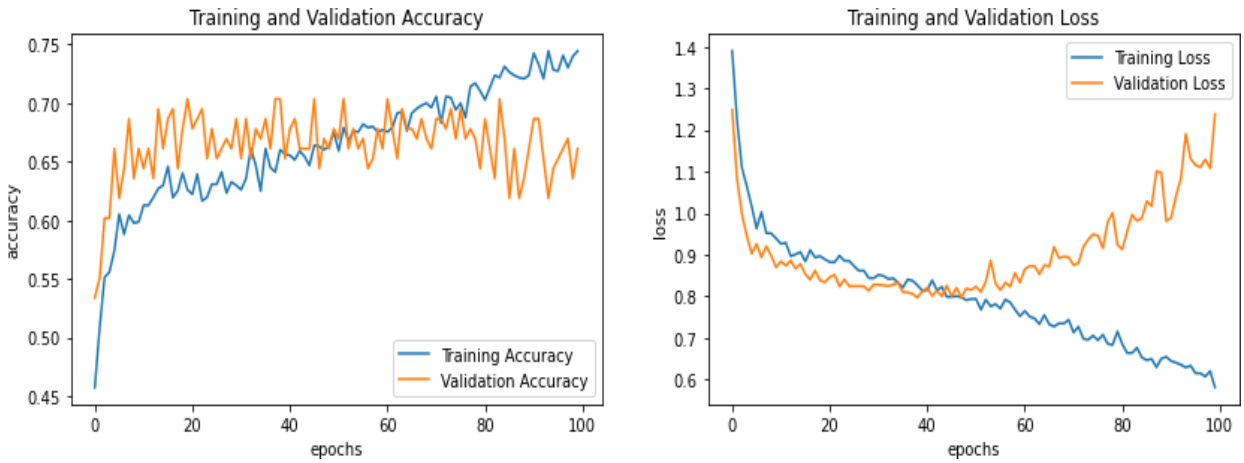
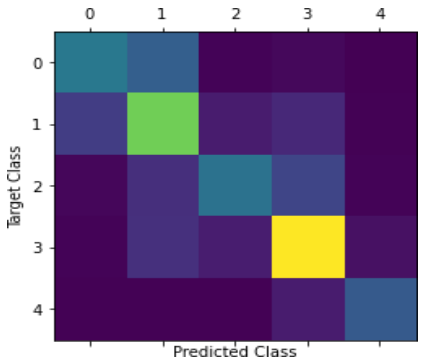


Figure 18: Training and validation accuracy and loss of LSTM on multi-class (five)

As we discussed the evaluation metrics in chapter three, the proposed models were evaluated by evaluation metrics such as precision, recall, and f-score on the LSTM model on the multi-classes. Here are the results of that accuracy in the figure below.



Classification Report

	precision	recall	f1-score	support
0	0.65	0.54	0.59	127
1	0.57	0.68	0.62	200
2	0.68	0.50	0.58	129
3	0.70	0.78	0.74	220
4	0.82	0.75	0.78	65
accuracy			0.66	741
macro avg	0.69	0.65	0.66	741
weighted avg	0.67	0.66	0.66	741

Figure 19: Evaluation metrics of the LSTM model

5.2.3. Result of BiLSTM Model

BiLSTM is another neural network model that follows a sequential data series. Instead of using the standard LSTM model, it is better to use the BILSTM model which can handle the information from both forward and backward directions. Using the BiLSTM model this study achieved an accuracy of **72.88**% and **71.88**% on text only dataset and text with Emoji dataset respectively with a skip-gram and an accuracy of **70.34**% and **72.66**% on text only dataset and text with Emoji dataset with a CBOW. The below figure shows the accuracy of training and validation and the loss of training and validation of the BiLSTM model on the multi-classes (five).

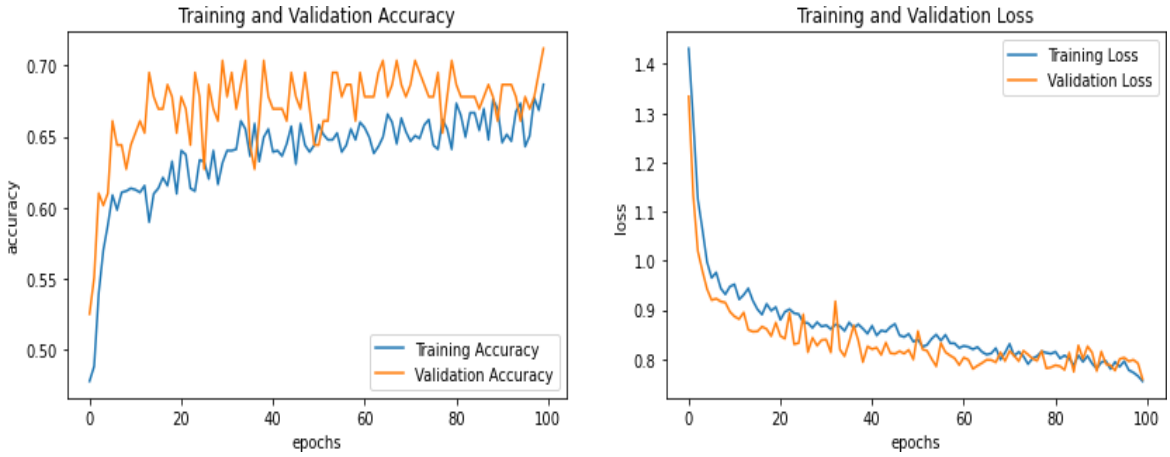
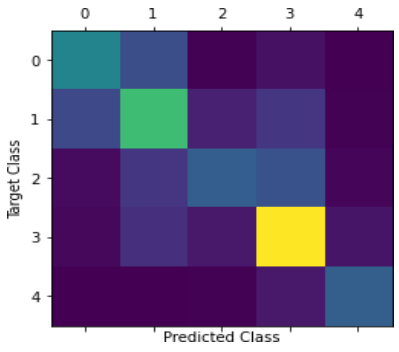


Figure 20: Accuracy and loss of the BiLSTM model on multi-class

The below figure shows the result of the evaluation metrics such as precision, recall, and f-score on the BiLSTM model on the multi-classes.



Classification Report

	precision	recall	f1-score	support
0	0.62	0.61	0.61	127
1	0.56	0.59	0.58	200
2	0.63	0.40	0.49	129
3	0.66	0.78	0.71	220
4	0.79	0.80	0.79	65
accuracy			0.63	741
macro avg	0.65	0.63	0.64	741
weighted avg	0.63	0.63	0.63	741

Figure 21: Result of the evaluation metrics of the BiLSTM model on multi-class

5.2.4. Result of CNN Model

CNN model was built with the optimal network hyperparameters that were trained with the prepared dataset. The model's accuracy on the text-only dataset and the text with Emojis dataset was **74.58**% and **71.09**% respectively with a skip-gram, and it was **73.73**% and **72.66**% when using a CBOW. The below figure shows the accuracy of training and validation and the loss of training and validation of the BiLSTM model on the multi-classes (five).

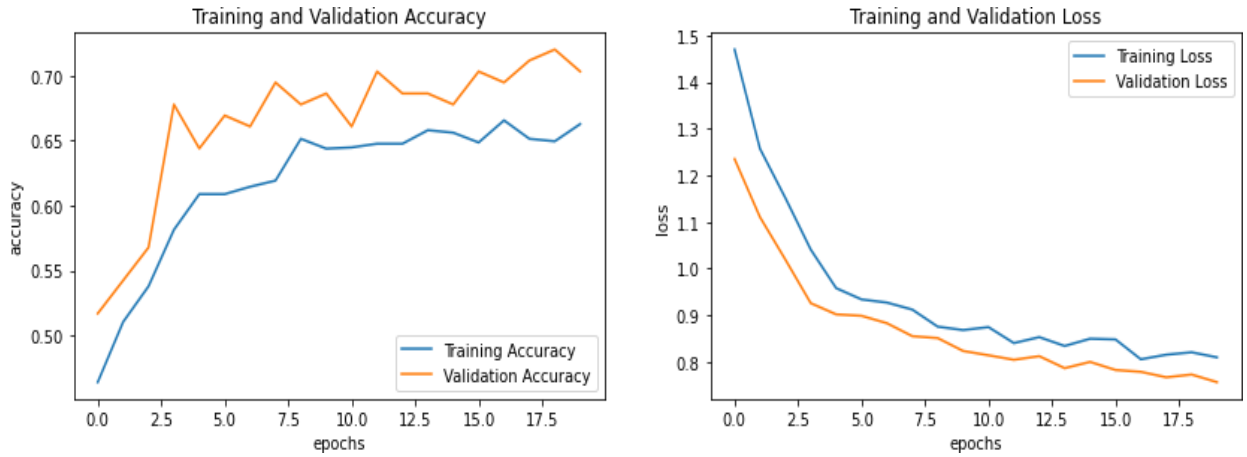
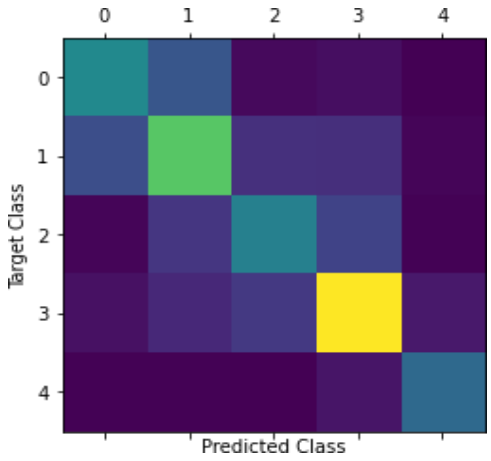


Figure 22: Evaluation metric of the CNN model for multi-class (five)

The evaluation result of the precision, recall, and f-score of the CNN model on multi-class was shown in the below figure.



Classification Report

	precision	recall	f1-score	support
0	0.61	0.59	0.60	127
1	0.58	0.58	0.58	200
2	0.57	0.53	0.55	129
3	0.70	0.72	0.71	220
4	0.79	0.83	0.81	65
accuracy			0.64	741
macro avg	0.65	0.65	0.65	741
weighted avg	0.64	0.64	0.64	741

Figure 23: Evaluation metrics of the CNN model

5.2.5. The effect of using Emojis on the Model Performance

Investigating the impact of including Emojis in text datasets on the performance of deep learning models is one of the goals of this research. By feeding distinct datasets of text only and text including emojis into CNN, LSTM, and BiLSTM models, the impacts of considering Emojis were assessed. The impact of emojis on experiments was demonstrated in Table 10 below.

Table 10: Comparison of deep learning model's performance with and without Emoji

Datasets	**LSTM**	**BiLSTM**	**CNN**
Text only	73.34	72.88	**74.58**
Text and Emoji	72.03	71.88	72.66

The result from Table 10, showed that the accuracy of the CNN model falls from **74.58**% to **72.66**%, on the BiLSTM model drops from 72.88% to 71.88%, and on the LSTM model from 73.34% to 72.03% when Emojis are used for comment labeling with Afaan Oromo text. The experiment's findings presented that adding Emojis to Afaan Oromo text reduces the CNN, BiLSTM, and LSTM models by **1.92**%, **1.00**%, and **1.31**% respectively. Thus, adding Emojis to Afaan Oromo text decreases the model's performance.

5.2.6. Comparison of the result of skip-gram and CBOW

The proposed deep learning models used word2vec's skip-gram and CBOW feature extraction to produce the neural networks' performance measurements. The findings of the trials are reviewed and shown in Table 11 below.

Table 11: Comparison result of a skip-gram and CBOW on the multi-class

Word2Vec	**LSTM**	**BiLSTM**	**CNN**
Skip-gram	72.03	72.88	**74.58**
CBOW	73.73	70.34	73.73

According to Table 11, the experiment's findings show that skip-gram surpasses a CBOW on the CNN and BiLSTM models with the accuracy of **74.58**% and **72.88**% respectively. However, the CBOW outperforms the skip-gram on the LSTM model with an accuracy of **73.73**%.

5.2.7. The effects of increasing the number of classes on model performance

The other objective of this study was to identify the effects of increasing the number of classes on the model's performance. To identify the effects of a different number of classes, this study did experiments by reducing the number of the classes from the multi-class (five) to three and binary classes.

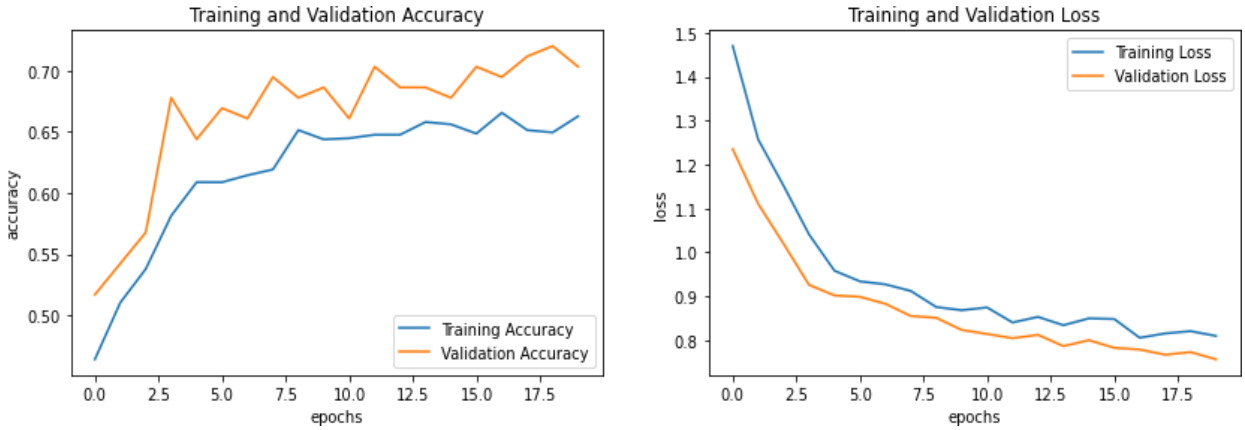


Figure 24: Evaluation results of the CNN model on multi-class (five)

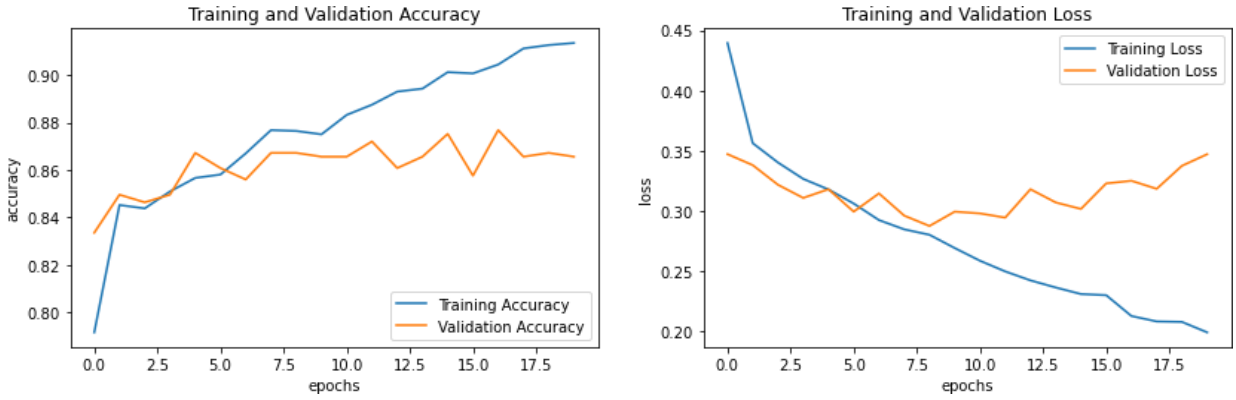
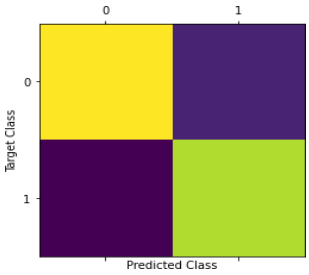


Figure 25: Evaluation results of the CNN model for binary class

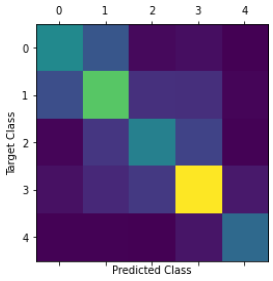
The impact of adding more classes is depicted in Figures 22 and 23 above. The results of CNN's multi-class experiment are shown in the first figure, whereas the results of the binary class experiment are shown in the second figure.



Classification Report

	precision	recall	f1-score	support
0	0.91	0.84	0.87	341
1	0.82	0.90	0.86	284
accuracy			0.87	625
macro avg	0.87	0.87	0.87	625
weighted avg	0.87	0.87	0.87	625

Figure 26: The evaluation result of the CNN model on binary classes



Classification Report

	precision	recall	f1-score	support
0	0.61	0.59	0.60	127
1	0.58	0.58	0.58	200
2	0.57	0.53	0.55	129
3	0.70	0.72	0.71	220
4	0.79	0.83	0.81	65
accuracy			0.64	741
macro avg	0.65	0.65	0.65	741
weighted avg	0.64	0.64	0.64	741

Figure 27: The evaluation results of the CNN model on the multi-class (five)

Figure 26 and 27 above shows the evaluation result of the CNN model on binary classes and multi-class that identifies the effects of increasing the number of classes.

Table 12: Result summary of using a different number of classes

Number of classes	**Skip-gram**			**CBOW**		
	LSTM	**BiLSTM**	**CNN**	**LSTM**	**BiLSTM**	**CNN**
Binary class	**89.60**	88.32	87.52	87.68	87.36	**88.64**
Three class	82.05	79.35	**87.04**	79.08	**79.89**	79.22
Multi-class (five)	72.03	72.88	**74.58**	73.73	70.34	73.73

From Table 12, the experimental findings showed that the performance of the proposed model declines as the number of dataset classes increases. For instance, the accuracy of the BiLSTM model with skip-gram feature extraction on the same dataset for the binary class, three classes, and multi-class (five) was **88.32**%, **79.35**%, and **72.88**%, respectively. The result of binary class was increased by **15.44**% when compared with the result of multi-class(five). The full summary of the result was available in table 12 above. In conclusion, the performance of the proposed models was negatively impacted by the increasing number of classes.

5.2.8. Hyperparameter Tuning

This study searches for the optimal hyperparameter for the proposed neural network models using the grid search hyperparameter tuning method. This study searched the combination of possible optimal hyperparameters. We search many hyperparameters such as activation, optimization, number of epochs, batch size, and others. The hyperparameters proposed models are explained in Table 13 below.

Table 13: Hyperparameter for the proposed model

Models	**Filters**	**Hidden layers**	**Activation**	**Dropout**	**Optimizer**	**Epochs**	**Batch size**
LSTM	128	2	Relu	0.25	Adam	100	42
BiLSTM	128	2	Relu	0.25	Adam	50	42
CNN	64	2	Relu	0.25	Adam	50	64

5.3. Discussions

The main goal of this study was to classify a sentiment analysis for Afaan Oromo text from Facebook and YouTube using Deep learning approaches. This study undergoes several experiments with a newly collected and prepared dataset on three proposed deep learning models to measure the performance of Afaan Oromo text sentiment analysis. The CNN, LSTM, and BiLSTM models have been used and evaluated using skip-gram and CBOW of word2Vec as feature extraction.

The paper (Wayessa and Abas 2020) has conducted a Sentence-level sentiment analysis from Afaan Oromo text-based using Machine Learning. The researcher used the SVM algorithm and TF_IDF feature extraction and achieved an accuracy of 90% on the politics datasets. Another paper (Oljira and Kekeba 2020) has conducted a sentiment analysis of Afaan Oromo using Machine Learning techniques. The researcher considered document-level sentiment analysis of Afaan Oromo and achieved an accuracy of 93% on the MNB algorithm on binary class polarity classification.

In general, the previous research on Afaan Oromo text sentiment analysis focused and delivered on a specific area of the dataset, did not consider Emojis, classify the sentiment analysis into binary or three classes (Oljira and Kekeba 2020), (Rase 2020), (Wayessa and Abas 2020),

(Kuyu 2021). Using a specific area of data and a lower number of classes helps the model to perform better performance because it only learns from the same area of data and the probability to classify the class is high. But when the area of the dataset and class of the dataset grows it possibly reduces the model's performance.

This study considered the effect of Emojis on the labeling dataset and model performance, including the data on different areas (political, economic, and entertainment) to prepare the standard dataset for Afaan Oromo sentiment analysis which helps the researcher for future works, considered a multi-class sentiment classification. In the end, we evaluated the effects of considering the Emojis on the labeling comments and also on the model performances, evaluated the impacts of increasing the number of classes on the model performance, selects the feature extraction that better performs with the proposed deep learning models, and also selects the models that outperform than the others for classification of sentiment analysis for Afaan Oromo text-based at sentence level form.

To achieve our goal, we formulated the research questions depending on the problems of this study area. So, this study discussed and found the solutions for the research questions such as the effects of the Emojis when included with the textual data on the labeling dataset and model performances, the effect of increasing the number of the classes on the model performances, which feature extraction was a better fit with deep learning models, and select the neural networks that produce the better performance for Afaan Oromo text at sentence level sentiment analysis. Finally, this study answered the research questions that are addressed under chapter one of section 1.4. This study addressed three research questions, below is the answer to those questions.

RQ1: Which feature extraction produces a better performance on Afaan Oromo text sentence level SA with the proposed deep neural networks? This research question compares the two types of word2Vec (skip-gram and CBOW) feature extractions that perform better performance with the proposed neural networks. Based on the experiment, the skip-gram is better performance on the BiLSTM and CNN neural models with an accuracy of **74.58**% and **72.88**% respectively. On the LSTM model, the CBOW performs better than the skip-gram, with an accuracy of **73.73**%. The researcher suggests a skip-gram feature extraction for the BiLSTM

and CNN models to classify sentiments for Afaan Oromo text-based at sentence-level, and a CBOW for the LSTM model, based on the findings of the experiment results.

RQ2: Which deep learning models efficiently produce a better classification model for sentence-level Afaan Oromo sentiment analysis? This research question compares the proposed neural networks on the prepared Afaan Oromo text-based sentiment analysis dataset. Depending on the outcome of the experiment, the CNN model performs better than the other two models with a **74.58**% accuracy rate. So, this study suggested CNN neural models for the classification of sentiments for Afaan Oromo text-based at the sentence level.

RQ3: What is the effect of Emojis on model performance for Afaan Oromo text-based sentence-level sentiment analysis? This research question deals with the effects of Emojis on the performance of the proposed model when it was written with Afaan Oromo texts. Based on the experiment, the accuracy of the CNN model falls from **74.58**% to **72.66**%, the BiLSTM model from 72.88% to 71.88%, and the LSTM model from 73.34% to 72.03% when text with Emoji dataset was used. The experiment's findings presented that adding Emojis to Afaan Oromo textual data reduces the CNN, BiLSTM, and LSTM models by **1.92**%, **1.00**%, and **1.31**% respectively. Depending on the outcome of the experiment, this study concludes that adding Emojis to Afaan Oromo textual data decreases the performance of the model.

CONCLUSION AND RECOMMENDATION

Conclusions

Sentiment analysis is a branch of NLP that determines how people feel. It classifies the sentiment of the people in the form of texts, Emojis, audio, and video as positive, neutral, or negative. Few studies have used traditional machine learning techniques to analyze sentiment in Afaan Oromo languages. Only one study was conducted Afaan Oromo sentiment analysis at the document level using deep learning models classifying the polarity into binary classes.

There hasn't been any research on Afaan Oromo sentiment analysis for different categories of data, like entertainment, economics, and politics, how emojis impact labeling datasets, and model performance when it was used with Afaan Oromo text. And also, the impact of class size on the performance of the models. To address the aforementioned issues, research in the field of sentiment analysis of Afaan Oromo languages is required.

The main objective of this research is to classify the sentence-level sentiment analysis for Afaan Oromoo text using deep learning models. Finding out how using Emojis with Afaan Oromo text affects the labeling dataset and model's performance is the other task of the research. To achieve this study's goal, the researcher gathered information from the OBN, BBC Afaan Oromoo, FBC Afaan Oromoo, and VOA Afaan Oromoo public Facebook pages, as well as the Vision Entertainment and Aanaa Entertainment YouTube channels. The dataset, which has 7,404 comments, was preprocessed using several NLP preprocessing techniques, including the normalization of words, tokenization, removing stop words, and cleaning the data.

This study proposed three deep learning models, such as CNN, LSTM, and BiLSTM word2Vec feature extraction that transform the textual data into its vector representations forms. Since there were differences between the number of dataset classes, this study used the dropout regularization technique to avoid overfitting. This study also used hyperparameters to improve the performance of the proposed models.

Emojis alter the class of the data in the labeling dataset. Emojis change the classification of the dataset from neutral to positive, neutral to negative, and so forth when they are used with Afaan Oromo texts. Emojis have an impact on the models' performance as well. According to the experiment's findings, using Emojis alongside Afaan Oromo text decreased the performance of

the LSTM, BiLSTM, and CNN models by **1.31**%, **1.00**%, and **1.92**% respectively, compared to the dataset that contains Afaan Oromo text only. Emoji use with Afaan Oromo text lowers the models' performance, per the investigated results.

The experiment's findings show that the performance of the model was impacted by the size of the class. This study created a different dataset for binary, three, and five classes to evaluate the models' performance. The experiment's results showed that as the number of classes rises, the proposed model performs worse. This study evaluates the effectiveness of each model with two different word2Vec types: skip-gram and CBOW. Compared to the other two models, the CNN model performs better, and therefore, this study chose the CNN model for Afaan Oromo text-based Sentence level sentiment analysis.

The contribution of this study was preparing a dataset that contains 7404 instances, identifying the effects of emojis when written with Afaan Oromo texts on the labeling of datasets and models performance, and identifying the effect of the size of the dataset classes on the performance of the models, classifying the sentiments into multiclass polarity. Finally, this study develops Afaan Oromo text sentiment analysis using CNN, BiLSTM, and LSTM models.

Recommendation

Developing the full functional and efficient sentiment analysis requires collaborative team works from different areas such as computer Science (AI) experts, linguistic experts, and other support workers. So, to develop Afaan Oromo sentiment analysis, we recommend interested researcher research on:

- Researchers conduct sentiment analysis studies for the Afaan Oromo language by specifying its level as sentence, document, and aspect levels. Instead of doing that, it’s better to conduct researches that include all level of the sentiment analysis on one model.
- The other recommendation is that consider the different areas such as political, economic, entertainment, sports, transportation, and others by preparing the standard Afaan Oromo datasets.
- The final suggestion is that for a country like Ethiopia which is rich in linguistic diversity and resources is more helpful if researchers consider more than one language to classify the sentiments to make multilingual

Future works

The researchers direct the future works as follows:

- This research paper is limited to comments written in the Afaan Oromoo language to classify the sentiment of the people. However, most users write comments using different languages like English, Afaan Oromoo, Amharic, and others. For the next work adding these languages helps to classify sentiment analysis on a multilingual system.
- This study was conducted on text-based sentiment analysis, in the future we consider audio and video data to classify the sentiments.
- In this study, Emojis are considered as identifying their effects on the labeling dataset and the model performance when it is included with Afaan Oromo text. In the future, we consider it to classify the sentiment as positive or negative.

REFERENCES

Abate, Jemal, and Meshesha Million. 2019. “Unsupervised Opinion Mining Approach for Afaan Oromoo Sentiments.”

Bhatia, Surbhi, Manisha Sharma, and Komal Kumar Bhatia. 2018. “Sentiment Analysis and Mining of Opinions.” *Studies in Big Data* 30(May): 503–23.

Dave, Kushal, Steve Lawrence, and David M. Pennock. 2003. “Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews.” *Proceedings of the 12th International Conference on World Wide Web, WWW 2003*: 519–28.

Duwairi, Rehab M. 2015. “Journal of Information Science.” (March).

“Fighting Overfitting With L1 or L2 Regularization_ Which One Is Better_ - Neptune.”

Fikre, Turegn, and Addis Ababa. 2020. “Effect of Preprocessing on Long Short Term Memory Based Sentiment Analysis for the Amharic Language.”

Ibrahim Bedane. 2015. “The Origin of Afaan Oromo : Mother Language.” *Double Blind Peer Reviewed International Research Journal* 15(12).

Kindeneh, Maru. 2020. “Opinion Mining for Amhara Broadcasting Agency News.”

Kuyu, Sultan Jenbo. 2021. “Developing an Automated Machine Learning Based Sentiment Analysis for Afaan Oromoo.” (February).

Lipton, Zachary C, John Berkowitz, and Charles Elkan. 2015. “A Critical Review of Recurrent Neural Networks for Sequence Learning ArXiv : 1506 . 00019v4 [Cs . LG] 17 Oct 2015.” : 1–38.

Mandelbaum, Amit, and Adi Shalev. 2016. “Word Embeddings and Their Use In Sentence Classification Tasks.” http://arxiv.org/abs/1610.08229.

“No Title.” 2021. (August).

Oljira, Megersa, and Kula Kekeba. 2020. “Sentiment Analysis of Afaan Oromo Using Machine Learning Approach.” *International Journal of Research Studies in Science, Engineering and Technology* 7(9): 7–15.

van Paridon, Jeroen, and Bill Thompson. 2021. “Subs2vec: Word Embeddings from Subtitles in 55 Languages.” *Behavior Research Methods* 53(2): 629–55.

Philippe, Rémi. 2016. “Word Embeddings for Natural Language Processing.” 7148.

Rajbanshi Sabita. 2021. “Everything You Need to Know about Machine Learning.” *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2021/03/everything-you-need-to-know-about-machine-learning/#h2_1.

Rase, Megersa Oljira. 2020. “Sentiment Analysis of Afaan Oromoo Facebook Media Using Deep Learning Approach.” *New Media and Mass Communication*: 7–22.

Sachdeva, Saransh Jitendra, Raj Abhishek, and Dr. Annapurna V.K. 2019. “Comparing Machine Learning Techniques for Sentiment Analysis.” *Ijarcce* 8(4): 67–71.

Sari, Sefriani amelia. 2017. “No Title549 ,7(November): 40–42.

Shao, Liqun, Hao Zhang, Ming Jia, and Jie Wang. 2017. “Efficient and Effective Single-Document Summarizations and a Word-Embedding Measurement of Quality.” *IC3K 2017 - Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management* 1: 114–22.

Shelar, Amrita, and Ching-Yu Huang. 2018. “Sentiment Analysis of Twitter Data.” *Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018* 3(1): 1301–2.

Tesema, Workineh, and Duresa Tamirat. “Investigating Afan Oromo Language Structure and Developing Effective File Editing Tool as Plug-in into Ms Word to Support Text Entry and Input Methods.” www.pubicon.in.

Tesfaye, Debela. 2011. “A Rule-Based Afan Oromo Grammar Checker.” 2(8): 126–30.

“Understanding LSTM Networks -- Colah’s Blog.” 2016. : 1–8. http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

Vaitheeswaran, G, and L Arockiam. 2016. “Machine Learning Based Approach to Enhance the Accuracy of Sentiment Analysis on Tweets.” *Journal of Advance Research in*

Computer Science and Management Studies 6(6): 185–90.

Wang, Lizhi et al. 2021. "A Deep-Forest Based Approach for Detecting Fraudulent Online Transaction." *Advances in Computers* 120: 1–38.

Wayessa, Negessa, and Sadik Abas. 2020. "Multi-Class Sentiment Analysis from Afaan Oromo Text Based On Supervised Machine Learning Approaches." *International Journal of Research Studies in Science, Engineering and Technology* 7(7): 10–18.

Wondwossen Mulugeta. 2014. "A Machine Learning Approach to Multi-Scale Sentiment Analysis of Amharic Online Posts." *HiLCoE Journal of Computer Science and Technology* 2(2): 80–87.

Yao, Cuili et al. 2016. "A Convolutional Neural Network Model for Online Medical Guidance." *IEEE Access* 4: 4094–4103.

APPENDICES

Appendix A: Guideline for data annotation

Define clear rules

The consistent rule aids in classifying each group of people's sentient beings according to their proper polarity. Sort the sentences into their polarity classes based on the sentences' degrees.

Sarcastic remarks are disregarded because they may simultaneously include positive and negative polarity. As a result, it is challenging to categorize the sarcastic sentence according to polarity rates. The annotator should take the following concepts into account before categorizing the polarity of the sentences. Which are:

- Determine the statement's polarity. Statements are categorized as positive, neutral, or negative depending on how they express their feelings.
- Sentence Topic Identification: When choosing a topic, annotators should exercise caution. Politics, the economy, and entertainment news are the chosen topics.
- Recognize sarcastic comments by looking for expressions that lack a clear indication of the correct polarity class. Additionally, it might not have accurately captured their semantics.
- The statement should be at the sentence level in terms of its level. Due to this study's emphasis on sentiment analysis at the sentence level.

Each annotator is required to adhere to the dataset annotation guidelines. Below is a list of the basic guidelines for annotating the provided sentence.

- Very positive: is a positive statement that has a high degree on the sentences than the positive class.
- Positive: the statement can be considered positive if it shows an explicit or implicit clue in the text. Showing support, and positive thoughts.
- Neutral: the statement can be considered neutral if it shows there is no explicit or implicit indicator of the speaker’s emotional state. This class is operational sentence neither support nor oppose.
- Negative: the statement can be considered negative if it shows an explicit or implicit clue in the text. This class is opposing the other’s ideas, limitations on the services, etc

- Very negative: the statement can be considered very negative if it shows a sentence that indicates, that targeting an individual or group because of ethnicity, sexual or gender, religion, and insults or hate are categorized under very negative class

In the end, comments should be classified to the class labels depending on the clear rule of sentiment analysis, and guidelines for labeling classes.

- Very negative represented **0**
- Negative represented **1**
- Neutral represented **2**
- Positive represented **3**
- Very positive represented to **4**

Appendix B: Sample codes

Sample codes for importing python libraries

```
# used for handling the dataset
import pandas as pd
 # regex
import re
# solving punctuation problems
from string import punctuation
import numpy as np
from numpy import array
from keras.callbacks import ModelCheckpoint
import keras
from tensorflow.keras.optimizers import SGD
import tensorflow as tf
from sklearn import preprocessing
# import xgboost as xgb
from keras import backend as K
from sklearn.preprocessing import OneHotEncoder, LabelEncoder
from keras.preprocessing.text import Tokenizer , text_to_word_sequence
from keras.preprocessing.sequence import pad_sequences
from keras.layers.embeddings import Embedding
from keras.models import Sequential, model_from_json, load_model
from keras.layers import LSTM, Dense, Input, concatenate, Concatenate, Activation, Flatten, Dropout
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Conv1D
from tensorflow.keras.layers import MaxPooling1D
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPooling2D
from keras.models import Model
```

Sample codes of cleaning the dataset

```
def clean_text_round(text):
    text=re.sub('\[.*?\]', '',text)
    text = re.sub(r'\s+', ' ', text).strip()
    text = re.sub('([@0-9_]+)|[^\w\s]|#|http\S+', '', text)
    return text
round1= lambda x:clean_text_round(x)
df1=pd.DataFrame(df.reviews.apply(round1))
```

Sample codes of removing stop words and tokenization

```
from nltk.corpus import stopwords
stop_words=set(stopwords.words('afaanoromoo.txt'))
from nltk.tokenize import word_tokenize
def remove_stopwords(sentence):
    word_tokens = word_tokenize(sentence)
    clean_tokens = [w for w in word_tokens if not w in stop_words]
    return clean_tokens
df['reviews'].apply(remove_stopwords)
```

Sample code for word2Vec feature extraction

```
#train word2vec model using genism
import gensim
from gensim.models import Word2Vec
from gensim.models import FastText
from gensim.models import KeyedVectors
Min_count=2
Size=100
Workers=4
Window=5
# fasttext
# model=gensim.models.FastText(sentences=stopw,vector_size=Size,sg=1, workers=Workers,min_count=Min_count, window=Window)
# model=gensim.models.FastText(sentences=stopw,vector_size=Size,sg=0, workers=Workers,min_count=Min_count, window=Window)
# model=gensim.models.Word2Vec(sentences=stopw,vector_size=Size,sg=1, workers=Workers,min_count=Min_count, window=Window)
model=gensim.models.Word2Vec(sentences=stopw,vector_size=Size,sg=0, workers=Workers,min_count=Min_count, window=Window)
model.build_vocab(stopw, progress_per=1000)
words=list(model.wv.key_to_index)
print(words)
```

Sample codes of BiLSTM

```
model = Sequential()
model.add(embedding_layer)
model.add(Bidirectional(LSTM(64,  return_sequences=True )))
model.add(Dropout(0.25))
model.add(Bidirectional(LSTM(32,  return_sequences=False )))
model.add(Dropout(0.25))
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(5))
model.add(Activation('softmax'))
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam',metrics=['accuracy'])
history = model.fit(X_train_padded, y_train, epochs=100, batch_size=64, validation_data=(X_test_padded, y_test))
print('\n# TEST DATA #\n')
score = model.evaluate(X_test_padded, y_test)
print("%s: %.2f%%" % (model.metrics_names[1], score[1]*100))
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss=history.history['loss']
val_loss=history.history['val_loss']
plt.figure(figsize=(14,4))
plt.subplot(1, 2, 1)
plt.plot(acc, label='Training Accuracy')
plt.plot(val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')
plt.xlabel("epochs")
plt.ylabel("accuracy")
plt.subplot(1, 2, 2)
plt.plot(loss, label='Training Loss')
plt.plot(val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.xlabel("epochs")
plt.ylabel("loss")
plt.show()
```

Sample codes for the CNN model

```
modelcnn = Sequential()
modelcnn.add(Embedding(input_dim=len(word_index) + 1,
                       output_dim=Size,
                       weights=[embedding_matrix],
                       input_length=max_length,
                       trainable=False))
modelcnn.add(Conv1D(128, 1, 1, input_shape=(8, 1, 1), activation='relu'))
modelcnn.add(MaxPooling1D(pool_size=1))
modelcnn.add(Conv1D(64, 1, 1, activation='relu'))
modelcnn.add(MaxPooling1D(pool_size=1))
modelcnn.add(Dropout(0.25))
modelcnn.add(Flatten())
modelcnn.add(Dense(64, activation='relu'))
modelcnn.add(Dropout(0.25))
modelcnn.add(Dense(32, activation='relu'))
modelcnn.add(Dense(5))
modelcnn.add(Activation('softmax'))
modelcnn.compile(loss='sparse_categorical_crossentropy', optimizer='adam',metrics=['accuracy'])
historycnn = modelcnn.fit(X_train_padded, y_train, epochs=100, validation_data=(X_test_padded, y_test))
print('\n# TEST DATA #\n')
#y_test=y_test.astype(float)
score = modelcnn.evaluate(X_test_padded, y_test)
print("%s: %.2f%%" % (model.metrics_names[1], score[1]*100))
```

Sample code for the LSTM model

```
modells = Sequential()
modells.add(embedding_layer)
modells.add(SpatialDropout1D(0.25))
modells.add(LSTM(128,  return_sequences=True ))
modells.add(Dropout(0.25))
modells.add(LSTM(64,  return_sequences=False ))
modells.add(Dropout(0.25))
modells.add(Dense(128, activation='relu'))
modells.add(Dropout(0.25))
modells.add(Dense(64, activation='relu'))
modells.add(Dropout(0.25))
modells.add(Flatten())
modells.add(Dense(2))
modells.add(Activation('softmax'))
modells.compile(loss='sparse_categorical_crossentropy', optimizer='adam',metrics=['accuracy'])
history = modells.fit(X_train_padded, y_train, epochs=200, validation_data=(X_test_padded, y_test))
```

Sample codes for testing the score

```
print('\n# TEST DATA #\n')
#y_test=y_test.astype(float)
score = modells.evaluate(X_test_padded, y_test)
print("%s: %.2f%%" % (modells.metrics_names[1], score[1]*100))
```

Sample codes for metric evaluation

```
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report, confusion_matrix
def report(X_data, y_data):
    #Confution Matrix and Classification Report
    Y_pred = np.argmax(modells.predict(X_data), axis=-1)
    y_test_num = y_data.astype(np.int64)
    conf_mt = confusion_matrix(y_test_num, Y_pred)

    print(conf_mt)
    plt.matshow(conf_mt)
    plt.ylabel('Target Class')
    plt.xlabel('Predicted Class')
    plt.show()
    print('\nClassification Report')
    target_names = ["v+","+", "-", "vn"]
    print(classification_report(y_test_num, Y_pred))
    print("Classification Report for Test Data\n")
report(X_test_padded, y_test)
```

Sample code of text the unseen data

```
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping # es = EarlyStopping(monitor = 'val_loss', mode = 'm
mc = ModelCheckpoint('mym/saved_model.pb', monitor = 'val_accuracy', mode = 'max', verbose = 1, save_best_only = True)


txt = ["sirba haaraa afaan oromoo kan ahdualem qulqullina bareeda qaba!"]

seq = tokenizer.texts_to_sequences(txt)
padded = pad_sequences(seq, maxlen=max_length)
pred = modelcnn.predict(padded)
labels = ['positive', 'negetive']

print(pred)
print(np.argmax(pred))
print(labels[np.argmax(pred)-1])
```

```
[[0.00143474 0.99856526]]
1
positive
```

```
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping # es = EarlyStopping(monitor = 'val_loss', mode = '
mc = ModelCheckpoint('mym/saved_model.pb', monitor = 'val_accuracy', mode = 'max', verbose = 1, save_best_only = True)


txt = ["oromoon maaliif waan gadhee akkasi dhiisuu didee?!"]

seq = tokenizer.texts_to_sequences(txt)
padded = pad_sequences(seq, maxlen=max_length)
pred = modelcnn.predict(padded)
labels = ['positive', 'negetive']

print(pred)
print(np.argmax(pred))
print(labels[np.argmax(pred)-1])
```

```
[[0.9853821  0.01461788]]
0
negetive
```

Appendix C: Afaan Oromo stop words

Sample of Afaan Oromo stop words

Akkana	akkam	akkamitti	akkasumas	kana	kanaafuu	fi
Irraa	jala	hennaa	yommuus	kanaaf	isaan	ishee
Obbo	aadde	amma	immoo	ani	nuyi	isa
akkuma	booda	dura	asitti	achii	eegaa	garuu
Inni	ittumallee	jechuu	kan	keenya	koo	maal