



WALLAGA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
COLLEGE OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE
**POLITICAL STANCE DETECTION FOR AFAAN OROMO TEXT
USING DEEP LEARNING APPROACH**

MSc Thesis

By: Chala Merga Lemu

Advisor: Dr. Obsa Gilo (Ph.D.)

June, 2025

Nekemte, Oromia, Ethiopia



WALLAGA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
COLLEGE OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE
POLITICAL STANCE DETECTION FOR AFAAN OROMO
TEXT USING DEEP LEARNING APPROACH

A Thesis submitted to the college of Engineering and Technology in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

By: Chala Merga Lemu

Advisor: Dr. Obsa Gilo (Ph.D.)

June, 2025

Nekemte, Oromia, Ethiopia



WALLAGA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

P.O. Box: 395, Nekemte, Ethiopia

APPROVAL SHEET FOR SUBMITTING OF FINAL THESIS

As members of the Final MA/MSc. thesis open defense examining board, we confirm that we have reviewed and assessed Chala Merga Lemu thesis titled “Political Stance Detection For Afaan Oromo Text Using Deep Learning Approach”. We recommend that the thesis be approved as meeting the requirements for the Master of Science degree in computer science.

_____	_____	_____
Chairperson	Signature	Date
_____	_____	_____
Internal Examiner	Signature	Date
Naol Bakala		_____
_____	_____	_____
External Examiner	Signature	Date
Final Approval and Acceptance		
Thesis Approved by		
_____	_____	_____
Department PGC	Signature	Date
_____	_____	_____
Dean of College	Signature	Date
_____	_____	_____
Dean of SGS	Signature	Date

STATEMENT OF THE AUTHOR

I, Mr. Chala Merga, attest that the thesis, "Political Stance Detection for Afaan Oromo Text Using Deep Learning Approach," is entirely original work of mine, overseen by Dr. Obsa Gilo(Phd). All scholarly ethics have been followed in the planning, collection, and analysis of the data for this thesis. Citations have been used to give credit to all of the scholarly sources cited in the thesis. Every original source has been correctly cited and attributed by me. In my submission, I further affirm that I have upheld to all academic honesty and integrity principles and have not forged, misrepresented, or falsified any facts, data, ideas, or sources. The criteria for the Master of Science in Computer Science degree are being partially met by this thesis, which is being submitted to Wallaga University. I also declare attest that this thesis is not granted a degree, diploma, or certificate by any other higher learning institution. I hereby declare I understand that the violation of any of the above conditions will invite disciplinary action from the institution. that breaking any of the aforementioned rules will result in disciplinary action from the institution. Furthermore, inadequate citation or seeking the required authorization can lead to sanctions.

Name: **Chala Merga Lemu** Signature: _____ Date _____

Department of Computer Science

College of Engineering and Technology

Declaration

I confirm that Chala Merga's thesis, "Political Stance Detection for Afaan Oromo Text Using Deep Learning Approach," which has been submitted as a requirement for the award of a Master of Science in Computer Science, is of university standard and at the right level of quality and originality.

Dr. Obsa Gilo (Ph.D.)

Name of Advisor

Signature

Date

Table of Contents

STATEMENT OF THE AUTHOR	ii
Declaration.....	iii
Table of Contents.....	I
List of Figures	V
List of Acronyms/Abbreviations.....	VI
Acknowledgment	VII
Abstract	VIII
CHAPTER 1: INTRODUCTION	1
1.1 Background of the study	1
1.2. Motivation.....	2
1.3. Statement of the Problem.....	2
1.4. Research Questions.....	4
1.5. Objectives of the Study	4
1.5.1. General Objective	4
1.5.2. Specific Objectives	4
1.6 Scope of the Study	4
1.7 Limitation of the study.....	5
1.8 Significance of the Study	5
CHAPTER TWO:	6
2. LITERATURE REVIEW AND RELATED WORK	6
2.1 Natural language processing (NLP).....	6
2.2 Afaan Oromo	6
2.2.1 Afaan Oromo Writing System	7
2.2.2 Afaan Oromo Punctuation Marks	7
2.2.3 Word Structure of Afaan Oromo	8
2.2.4 Sentence Structure of Afaan Oromo	8
2.3 Natural Language Processing.....	8

2.4 Stance Detection	9
2.5 Approaches of Stance Detection	9
2.6 Deep learning	10
2.7 Related Work	10
2.8 Research Gap	12
CHAPTER THREE: METHODOLOGY	14
3. Introduction.....	14
3.1 Data Collection	14
3.2 Data Preprocessing.....	16
3.3 Development environment and tools	16
3.4 Evaluation Metrics	17
CHAPTER FOUR: DESIGN OF THE STUDY	19
4. Introduction.....	19
4.2 Preprocessing	21
4.2.1 Tokenization	21
4.2.2 Stop Word Removal.....	22
4.2.3 Stemming	22
4.2.4 Handling special characters	23
4.2.5 Padding	23
4.3 Encoding	23
4.4. Word Embedding.....	24
4.5. Building model.....	25
4.5.1 Deep Learning.....	25
4.5.2 Neural Networks	26
4.5.3 Convolutional neural networks	30
4.5.4 Recurrent neural networks (RNNs).....	31
4.5.5 Long short-term memory (LSTM) and gated recurrent unit (GRU).....	32
4.5.6 Deep belief networks (DBN)	32

4.5.7 Deep stacking networks	32
4.5.8 Bidirectional Recurrent Networks (BRNN).....	33
4.5.9 Gated Recurrent Units (GRUs)	34
4.6 Training the model	34
4.7 Stance Classification	34
CHAPTER FIVE: RESULT AND DISCUSSION	35
5.1. Overview	35
5.2. Data gathering and preparation	35
5.3. Development Environment	35
5.3.1. Computer used	35
5.3.2. Development Software.....	35
5.4. Implementation	36
5.4.1. Experiment using Attention-based LSTM	38
5.4.2. Experiment using Attention based RNN model.....	39
5.5. Discussion and result	41
5.6. Performance Evaluation and Validation	42
5.7. The Model Prototyping	44
CHAPTER SIX: CONCLUSION AND FUTURE WORK.....	46
6.1. Conclusion	46
6.2 Contributions of the Study	47
6.3. Recommendation	48
Reference	43
Appendixes	48

LIST OF TABLES

Table 1: Summary of related works and Research gaps	13
Table 2: Confusion Matrix with the evaluation metrics	17
Table 3: Description of Hyper-parameters	38
Table 4: Training and performance of the proposed model.....	41

List of Figures

Figure 1: Sample of collected data.....	16
Figure 2: confusion matrix.....	18
Figure 3: The System architecture	20
Figure 4: Weight from model	25
Figure 5: Single-layer perceptron	27
Figure 6: Artificial neuron for a multilayer perceptron	28
Figure 7: Fully connected multilayer feed-forward neural network topology.....	29
Figure 8: This architecture will act as first layer in our attention based RNN model	37
Figure 9: Accuracy/loss Graph of Attention-based LSTM for stance classification.	39
Figure 10: Accuracy/loss Graph of attention based RNN for stance classification.....	40
Figure 11: Attention based LSTM Evaluation performance.....	42
Figure 12: Attention based RNN evaluation performance	43
Figure 13: Graphical User interface design for our model	44
Figure 14: Stance detected as 'Against' by our model	44
Figure 15: Stance detected as 'Supporter' by our model	45
Figure 16: Sample of predicted result for both “Against and Supporter “by our models.....	45

List of Acronyms/Abbreviations

APIs.....	Application Programming Interfaces
BERT.....	Bidirectional Encoder Representations from Transformers
BOW.....	Bag Of Words
CNN.....	Convolutional Neural Networks
CTC.....	Connectionist Temporal Classification
DL.....	Deep learning
DSR.....	Design Science Research
FN	False Negatives
FP	False Positives
GPU.....	Graphics Processing Unit
GRU.....	Gated Recurrent Unit
IR.....	information retrieval
LR.....	Logistic Regression
LSTM.....	Long short term memory
MDL	Multi-Dataset Learning
NLP.....	Natural language processing
NLTK.....	Natural Language Toolkit
RNN.....	Recurrent Neural Network
SVM.....	Support vector machine
TF-IDF.....	Term Frequency-Inverse Document Frequency
TN	True Negatives
TP	True Positives

Acknowledgment

My greatest gratitude is to God Almighty, who has given me strength and health and has constantly supported me throughout my life. Additionally, I want to express my profound gratitude to Dr. Obsa Gilo (Ph.D.), my advisor, for his leadership, generous, and encouragement during this thesis. His guidance and support helped me overcome several difficult situations and finish this thesis.

Furthermore, the department of computer science, College of Engineering and Technology deserves my sincerest gratitude for this opportunity and for their dedication to offering access to a wealth of educational resources.

I am particularly grateful to Mrs. Sinkinesh Mekuria (MA), an Afaan Oromo professional, for her insightful comments, suggestions, and contributions to this research. I also like to thank my family and friends for their part in the success and their dedication. I also prefer to thank Wallaga University for assisting me with this task and for funding.

Abstract

Social media and other technology advancements are becoming more and more important tools for connecting with people around the globe, including governments, social media activists, and political figures. Social media has recently been used by people to voice their thoughts on a variety of topics or issues. In the area of natural language processing, stance detection or classification has several applications. For example, automatically determining if a community supports or opposes a specific position on political and religious issues, either in support of or against the stated objectives. We created our own dataset for this study, which included 10,136 Afaan Oromo comments, targeting the OFC, OLF, and Oromia Prosperity Party. The data were preprocessed and subjected to morphological analysis following their collection and annotation in accordance with annotation rules. This research utilized a dataset comprising 10,136 Afaan Oromo text comment sourced from three political parties' verified Facebook pages OLF, OFC, and Oromia Prosperity parties. To address the challenges of Afaan Oromo text stance categorization, we proposed an attention-based RNN and an attention-based LSTM in our work. Our experimental results show that the Attention-based Long Short-Term Memory (LSTM) model achieves 98.6% training accuracy and 97.5% validation accuracy. In comparison, the second model in this study, the Attention-based Recurrent Neural Network (RNN) model, achieved 95.82% training accuracy and 93.47% validation accuracy. The greatest training accuracy of 98.6% and the highest validation accuracy of 97.5% show that the Attention-based LSTM model outperforms the others. We used previously unseen data to assess our model's effectiveness by applying accuracy, recall, and F-Score measures to categorized text using validation data.

A precision, recall, and f-score of 0.83, 0.83, and 0.83 were attained by the attention-based RNN, respectively. The precision, recall, and f-scores of the attention-based LSTM f-measure were higher than those of the attention-based RNN, with respective values of 0.94, 0.94, and 0.93. Consequently, this implies that the stance classification performance of the attention-based LSTM model is superior.

Keywords: Natural language processing, Stance classification, Deep learning, Attention mechanism.

CHAPTER 1: INTRODUCTION

1.1 Background of the study

Natural language processing (NLP), which combines linguistics and artificial intelligence, aims to teach computers how to analyze words or assertions in human language. Its two components are natural language understanding (also known as linguistics) and natural language production (which enhances the ability to understand and write text). They were created to simplify user tasks and to meet the demand for natural language communication between software and users. Linguistics, or the study of language, encompasses phonology, which deals with sound; morphology, which deals with word production; syntax, which deals with sentence structure; semantics; and pragmatics, which deals with comprehension (Chowdhary, K., & Chowdhary, K. R. 2020).

Political stance detection is the subtask of opinion mining that aims at identifying an individual's or an organization's position on a political candidate or issue based on written or verbal text. Recognition of political stances can yield useful information for a variety of applications, including public opinion surveys, social media monitoring, opinion mining, and election prediction.

Political stance detection is a subfield of Natural language processing (NLP) that uses written or spoken language to identify political beliefs of individuals or groups (Oshikawa, R., et al, 2018). This technology has grown in importance recently, as political divisiveness and ideological divides have increased in many nations.

Political stance detection aims to evaluate vast amounts of text data, such as social media posts, news stories, or political speeches, to establish the author's or speaker's political leanings (Bergam, N., et al., 2022). This can be accomplished through a variety of methods, including deep learning algorithms, sentiment analysis, and subject modeling. Stance identification is the process of identifying and assessing an individual's or group's attitudes, views, beliefs, or viewpoints on a specific topic or issue (AlDayel, 2021). It is frequently employed in natural language processing (NLP) and machine learning to assess the sentiment or emotional tone of text data, including social media updates, news stories, or political addresses. (Zubiaga, A. et al, 2018).

1.2. Motivation

Stance categorization is a relatively recent study area in natural language processing. Because this research area is young, there isn't much literature on the subject. However, as social media platforms become more widely available, people are increasingly using them to express their opinions on a variety of issues, whether they support or oppose a specific matter. The impetus for addressing the problem of political position detection in Afaan Oromo text using deep learning method originates from the growing the existence of political dialogue and the restricted advancement of linguistic resources in Afaan Oromo.

Afaan Oromo is a language that lacks resources, as demonstrated by the limited availability of specifically language processing tools is for this purpose. Afaan Oromo is a widely used genre, but its limited natural language processing resources make it difficult to develop successful computer programs for those who speak Afaan Oromo. The lack of development of applications and resources in Afaan Oromo prompted us to initiate this study.

In summary, what motivated us to conduct this research is:

- The lack of development of applications and resources in Afaan Oromoo.
- Growing political discourse on social media, especially in under-researched languages.
- lack of NLP resources and applications on political stance detection for Afaan Oromo, w/c widely spoken language
- To foster informed decision-making and more inclusive communities by understanding political viewpoints.
- to develop the first deep learning models for political stance detection in Afaan Oromo, which help people better understand each other's political viewpoints, make informed decisions, and contribute to building more inclusive and democratic communities

1.3. Statement of the Problem

The widespread availability of the Internet and mobile technology has caused in a considerable growth for communication in the usage of social media platforms and the dissemination of varied perspectives, enabling more participation and the sharing of thoughts and opinions. Despite its

benefits, this development does present certain obstacles (Edosomwan, 2011).

The intricacy and nuance of political ideology is one of the most difficult aspects of determining political opinion. A variety of elements, including social, economic, and cultural forces, as well as personal experiences and beliefs, frequently determine political opinions. As a result, correctly identifying political positions necessitates a thorough comprehension of language and context. Several issues have surfaced in our country, including people being displaced as a result of civil war, high unemployment rates, and rising living costs. These challenges come from the political parties' lack of knowledge of the people's interests and perspectives on policies and public concerns.

In spite of these constraints, political stance detection has numerous uses in politics, journalism and social media analytics. Political campaigns are able to target specific types of voters, journalists are able to interpret public opinion, and obtain valuable insights on political polarization and online discourse due to political stance detection.

Researchers and analysts are able to discover a lot about the multifaceted and wide realm of politics through interpretation of large volumes of text data using advanced natural language processing (NLP) methods.

Existing research on posture classification in social media has primarily concentrated on European and English languages. Some studies have also been undertaken on the Amharic language. However, we cannot apply this approach directly to Afaan Oromo because of variations in the characteristics of the language, writing system, morphological structure, and encoding style in social media communication. As a result, there is a significant research vacuum in the field of detecting political stances in Afaan Oromo using deep learning approaches. To the best of our knowledge, several experts have conducted sentiment analysis study on Afaan Oromo. Although sentiment analysis has been studied for Afaan Oromo, sentiment analysis models cannot accurately represent a user's attitude toward a target or issue. Also, most researchers used BOW and TF- IDF for data representation that means indirectly there is no semantic similarity between data. So we solved semantic similarity problem by using word embedding.

There have been no studies on political position detection in Afaan Oromo using a deep-learning approach. As a result, we will consider this a significant gap in future study. The goal of this research is to address a research vacuum by applying deep learning techniques to evaluate Afaan Oromo comments on the official Facebook pages of the Prosperity Party (Paartii Badhaadhinaa), the

Oromo Federalist Congress (OFC), and the Oromo Liberation Front (OLF/ABO) in order to better understand people's attitudes toward political issues. As per existing knowledge, this is the first study to employ a deep-learning technique for political positions detection in Afaan Oromo.

1.4. Research Questions

1. Which feature extraction method is most suited to represent the dataset of Afaan Oromo political stances?
2. Which deep learning algorithm is best for political stance classification area?
3. How effective is the proposed model in detecting stances?

1.5. Objectives of the Study

1.5.1. General Objective

The general objective of this study is to develop Political stance detection model for Afaan Oromo text using deep learning approach.

1.5.2. Specific Objectives

In order to accomplish our research's main goal, we have determined a number of specific goals. These include;

- ✧ A comprehensive analysis necessitates reviewing prior studies and determining suitable approaches and tactics.
- ✧ To gather and prepare the dataset
- ✧ To find the best features to extract for our classification.
- ✧ To develop a deep learning model that is suitable for the suggested task
- ✧ To select architecture for a deep learning model that is suitable for text categorization tasks.
- ✧ The resulting classification model will be compared and evaluated using various metrics.

1.6 Scope of the Study

The goal of this research is to create a deep learning-based model for detecting political stances in Afaan Oromo text. The concentrate on determining the best feature extraction technique and deep learning algorithm for effectively classifying political opinions. The study focus on a corpus of Afaan Oromoo text samples with different political stances. Dataset was collected from Facebook

comments of Oromia prosperity party, OFC and OLF. The dataset is annotated into two classes (supporter or against). The study will also look into how diverse characteristics such as complexity of language, text length, and text source affect classification of the model's accuracy.

1.7 Limitation of the study

People on social media can convey their feelings about publicly publicized topics using emojis (emotion icons); nevertheless, this study does not evaluate emojis.

The study only looks at the direct meaning of stances and does not take into account the indirect meaning of words (idiomatic phrases). It is also limited to Afaan Oromo texts on social media. And also, Cannot determine who wrote a text of his/her stances.

1.8 Significance of the Study

The study is significant because it has the potential to help develop more efficient and accurate Afaan Oromo position of political detection systems. With the increased usage of social media platforms and online news sources, it is critical to precisely identify individuals' and groups' political opinions. This can help with election prediction, campaign direction, and understanding public feelings and opinion on political issues.

The study of deep learning-based political position recognition in Afaan Oromo text has significant ramifications for several fields.

- ❖ Analyze the Afaan Oromo for political stance detection.
- ❖ Adjust campaign strategies based on potential voters' stances.
- ❖ Learn how political positions and opinions are created and shared in Afaan Oromo-speaking communities.
- ❖ Learn about people's political party stances. - Improve understanding of others' political views.
- ❖ Create excellent computer based system used to detect political stances for Afaan Oromo speakers.
- ❖ This study aims to develop deep learning-based systems for detecting political stances in Afaan Oromo.
- ❖ Assists political parties in understanding their supporters' opinions and interests.

CHAPTER TWO:

2. LITERATURE REVIEW AND RELATED WORK

2.1 Natural language processing (NLP)

Natural language processing (NLP) is a field of computer science that seeks to empower machines to understand language in the same way that humans understand it (Chowdhary, KR1442, and K. R. Chowdhary., 2020). Typical activities include question formulation, text sentiment analysis, and speech recognition. Applications of natural language processing (NLP), a fast developing field of study, have significantly boosted artificial intelligence (AI)(Beysolow, 2018).

2.2 Afaan Oromo

Afaan Oromo is Africa's third most widely spoken language, with over 50 million speakers in Ethiopia, Kenya, Somalia, and Egypt. The Oromo people are Ethiopia's biggest ethnic group, who account for around 40% of the population (Dulume, 2022). Although they can be found across Ethiopia, they are most common in the southwest regions of Go Jam, Wollega, Shoa, Illubabor, Jimma, Arsi, Bale, Hararghe, and Wollo. People utilize Afaan Oromo, another name for the Afaan Oromo on social media. (Arega 2020).

Afaan Oromo is the working language in Oromia and several other Ethiopian federal states. This, too, is an official working language in Ethiopia. The Oromia Zone encompasses Harari, Dire Dawa, and the Amhara Region. It is the primary language of teaching in Oromia, Harari, Dire Dawa, Benishangul-Gumuz, Addis Ababa, and the Oromia Zone of the Amhara region (Ikome, 2019). Federal websites employ it alongside Tigrigna as an internet language (Olbasa, 2018). Afaan-Oromo (which translates to "Oromo language") is a Cushitic language family (Alemayehu O, FenetB, 2019) that uses the Latin script and has around 50 million speakers in Ethiopia, Somalia, Egypt, and Kenya (Erena, 2021). The Afaan Oromo alphabets listed below have both capital and lowercase letters.

A B C CH D DH E F G H I J K

a b c ch d dh e f g h i j k

L M N NY O P PH Q R S SH T TS U

l m n ny o p ph q r s sh t ts u

V W X Y Z ZY

v w x y z zy

2.2.1 Afaan Oromo Writing System

The Latin-based alphabet used for the Afaan Oromo is called Qubee (Debela, 2019). It was legally recognized as the Afaan Oromo script about 1991. With certain adjustments, the language has many similarities to the English writing system. As a result, the letters used in English are likewise utilized in Afaan Oromo, with the primary distinction being how they are expressed orthographically. Afaan Oromo is categorized as a phonetic language, implying that its written and spoken forms are similar. Unlike English and other Latin-based languages, this language lacks any silent or deleted letters or sounds. The 33-letter Qubee Afaan Oromo typing system includes all 26 letters of the English alphabet as well as seven additional compound consonants called "Qubee Dachaa." This group includes ch, dh, sh, ts, ny, ph, and zy. In "Qubee," every English vowel (e, o, a, u, and I) is considered a vowel. The vowels in this language have two qualities that allow them to convey diverse meanings. These features are divided into two categories: short vowels and long vowels. When a vowel appears only once, it is considered short; when it appears twice, the maximum number of times, it is referred to as a long vowel. Examples could include lafa (ground) and laafaa (soft). Phonetically, the emphasis is greatly increased when a consonant is doubled. For example, damee (branch) versus dammee (sweet).

2.2.2 Afaan Oromo Punctuation Marks

In order to facilitate reading and make information understandable, punctuation is utilized in the text. The following is a list of the most frequently used punctuation marks in Afaan Oromo, which are similar to those in English. .

1. **"Tuqaa," full stop (.)**: This punctuation is used at the end of sentences and in abbreviations, just like in English.
2. **"Mallattoo Gaaffii,"** which ends a direct question with an interrogative or question mark (?).
3. To complete command and exclamatory phrases, use the exclamation point (!) **"Raajeffannoo".**
4. **"Qoodduu," Comma (,)**: is used to separate a list of parts or a sentence.
5. **"Tuqlamee," a colon (:)**: is used to begin and end sentences, lists, and quotes. It also has a number of other widespread uses.

6. In contrast to English, the apostrophe (') is present in Afaan Oromo's words as a placeholder for punctuation marks. Some examples are Har'a (now), re'ee (cow), and others.

2.2.3 Word Structure of Afaan Oromo

One or more recognizable, regular sounds can also make up a linguistic unit. There is a great range of terminology in the Afaan Oromo, from a few monosyllabic terms to ones with up to seven syllables (Bedane, 2015). Because the writing system is both written as it is read and read as it is written, it is simple.

Whitespace characters are also used in Afaan Oromo to differentiate words. Hence, white space characters enable the word segmentation (tokenization) technique, which includes adding suitable word borders for information retrieval reasons to an input text. The smallest component of language is the word. One of many ways to distinguish words from one another is Most languages use the blank letter, or space, to denote the conclusion of a word; "/" and ". " should not be used to indicate word limits as they are used to abbreviate lengthy words. The usual parentheses, brackets, quotations, and other characters in the Afaan Oromo show word limits.

2.2.4 Sentence Structure of Afaan Oromo

There are differences between the syntax and sentence structures of English and Afaan Oromo. The subject-object-verb (SOV) structure of the Afaan Oromo is the outcome of its flexible or changeable word order. In Afaan Oromo, an adjective is in agreement with its head noun; in English, however, this is not the case. While adjectives usually appear before a noun in English, they come after a noun or pronoun in Afaan Oromo. Following the word *miicaayyoo* is "*miicaayyoo bareedduu*" (beautiful girl).

The ending of Afaan is reminiscent of the Latin writing system in English and other languages, as Oromo sentences use a The full stop (.) in a statement, the question mark (?) in an interrogative, and the exclamation mark (!) in command and exclamatory phrases indicate that the sentence has come to an end. The semicolon denotes a break that is stronger than a comma but less definitive than a full stop balance.

2.3 Natural Language Processing

Natural language processing (NLP), an interdisciplinary field that falls under the umbrella of linguistics, computer science, and artificial intelligence, examines how computers interact with

human language and, more specifically, how to teach computers to process and analyze large amounts of natural language data (Kedia, A., et al., 2020).

The broad and diverse field of natural language processing (NLP) has expanded since then to encompass a wide range of tasks and applications, such as speech recognition, natural language understanding, natural language generation, machine translation, sentiment analysis, text summarization, question answering, information extraction, and more.

Natural language processing is a fascinating and challenging discipline with many benefits and uses in a variety of industries and domains, such as social media, business, healthcare, education, and entertainment. To get around these challenges and increase its effectiveness and quality, NLP therefore requires ongoing research and innovative development (Sadiku, M. N., et al., 2022).

2.4 Stance Detection

The technique of figuring out how a person's or people's feels about a specific topic, issue, or object is called "stance detection. Examining textual content to determine if the author is neutral, opposed to, or supportive of a certain topic is what it comprises. To ascertain public opinion and mood, stance detection is frequently employed in market research, political discourse analysis, and social media analysis. For a variety of applications, including sentiment analysis, opinion mining, and text classification, stance detection aims to automatically categorize the author's stance as either positive or negative (Küçük, D., & Can, F., 2020).

2.5 Approaches of Stance Detection

The goal of the deep learning task known as "stance detection" is to determine the placement of an entity or person inside a textual output toward a target (a concept, idea, event, etc.) that is either explicitly mentioned in the text or simply indicated. The three most common stance classifications are none, against, and favor. Can, F., and D. Küçük, 2022). Depending on the input representation, model architecture, and output format, there are many stance detection strategies. Here are a few typical methods:

In supervised learning, a labeled dataset containing examples of political positions and their labels is used to train a machine learning algorithm. Decision trees, logistic regression, and support vector machines (SVMs) are a few supervised learning methods that can be used to determine political stances.

For a variety of datasets, the majority of stance detection studies that employed supervised techniques like SVM, Random Forest, and Logistic Regression were shown to be the most effective (Aldayel, 2021).

Unsupervised learning involves training a machine learning system on an unlabeled dataset containing unlabeled political positions. Unsupervised learning seeks out organic groups (clusters) in the data.

The machine learning algorithm is trained using a smaller labeled dataset and a larger unlabeled dataset in semi-supervised learning. The unlabeled dataset is used to improve the model's performance after the approach is trained using the labeled dataset, which may improve the detection of political attitude.

The effectiveness of deep learning-based models for detecting political stances is largely dependent on the size and quality of the training data, the precision of data labeling, the suitability of feature selection, and the choice of the best algorithm for the given problem (Bencsik, B., et al., 2023).

2.6 Deep learning

The neural network of the brain is simulated by virtual computer systems in a type of artificial intelligence known as deep learning. It is specifically referred to as "deep learning" while being under the more general category of machine learning since it makes use of deep neural networks. These networks have multiple layers that allow the computer to analyze and learn from data. The model's level of sophistication is determined by its number of layers. For deep learning, a neural network makes learning easier. (Gurus 99, 2021).

2.7 Related Work

The main objective of this related work analysis is to critically assess the corpus of information pertaining to past research on the proposed study topic. An overview of the pertinent works is provided below.

Using a gradient decision tree and RF-based technique, C. Liu et al. (2016) proposed a supervised method for stance detection that integrated classifiers into an ensemble system that demonstrated good performance in attaining minority classifications. However, when it comes to topic-specific tweets and intense political tweets, where the same words might mean different things depending on the point of view, their approach is useless. Additionally, their model includes a significant number

of irrelevant attributes that were explicitly chosen to aid in the evaluation of tweets with ambiguity issues.

With a reported categorization F-score of 63.6%, the approach finds a wide variety of viewpoints. (M. Diab and H. Elfardy, 2016) used the SemEval-2016 dataset to propose a supervised stance detection method. To determine a tweeter's position for predetermined targets, their method used a variety of characteristics, including lexical, sentiment, semantic, latent, and frame semantic. The classification's reported F-score was 63.6%.

(E. Zotova, et al., 2019) suggested a stance identification method that used SVM and TF-IDF, with lemmatization used to generate the feature vector. The tweet's terms that lacked roots were eliminated and lemmatized.

Tokenizing each tweet's word list and translating it to the TF-IDF score using a dictionary built from the unigrams in the training dataset allowed for the construction of the feature vector. An F-score of 56% was obtained when the approach was evaluated on the SemEval-2016 dataset.

A stance identification method using the BERT architecture as its feature vector was created by (B. Schiller et al. in 2021). Google created the open-source NLP pre-training model known as BERT. Ten datasets from five distinct regions were used to train the MDL model as part of their methodology. Their model's evaluation on the SemEval-2016 dataset yielded an F-score of 71.62%.

(Tadesse, 2023) Used machine learning to show how to determine political stances in Amharic text. They employed four different feature extraction techniques: Bag-of-Words (BOW), TF-IDF, N-gram, and word2vec. Three machine learning algorithms were used. They were Support Vector Machine (SVM), Random Forest (RF), and Logistic Regression (LR). They achieved an 82% F-score by employing the TF-IDF feature extraction methodology in conjunction with the SVM machine learning method.

In contrast, an SVM-based learning system was created in a prior study by L. Barbosa and J. Feng (2010). It first classifies tweets into subjective or objective perspectives before classifying them as negative, positive, or neutral.

Unigram and feature-based models were employed in this investigation; their respective F-scores were 72.4% and 68%. Their kernel and sent-features model produced the best performance, 60.83%.

The authors used a dataset from (W. Ferreira and A. Vlachos, 2016) that consists of 2,595 news stories and 300 statements that were assessed for veracity and given the appropriate labels. A headline summarizing each article indicates whether it supports, refutes, or observes the assertion. According to the observation label, the article only restates the assertion. This dataset is a useful source of information for a range of fact-checking NLP activities.

2.8 Research Gap

The majority of research on using social media attitude classification to determine people's feelings toward a particular subject has focused on English and European languages, according to a review of related literature. Local languages, including morphologically rich ones like the Afaan Oromo, are where people write their feelings. As far as we are aware, no work has been done in this area before. In this proposal, some of the efforts that have been made to classify Afaan Oromo writings according to their political perspective will be reviewed. The machine learning model will be significantly impacted by the domain in which natural language processing is being employed. Learned models usually show poor adaptation across domains or across different text genres because they often rely on domain-specific attributes from their training data (N. Alturayef, et al, 2016). This study's main objective is to bridge the gap by compiling Afaan Oromo-specific data sets and assessing public stance on political matters, which will be mostly utilized by the government to develop successful policies.

Summary of related works and Research gaps

Authors	Year	Title	Method	Result	Research gap
C. Liu, W. Li, B. Demarest, Y. Chen and S. Couture	2016	An Ensemble Model for Stance Detection in Twitter	RF-based method with gradient decision trees	Accuracy of 63.6%,	The model is not work for Afan Oromo texts
H. Elfardy and M. Diab	2016	Ideological Stance Detection in Informal Text	lexical and all semantic features	F-score of 63.6%.	It doesn't apply to texts from Afan Oromo.
E. Zotova, R. Agerri and G. Rigau	2019	Automatic stance detection on political discourse in Twitter	TF-IDF and SVM	F-score of 56%	Afan Oromo texts are not supported.
B. Schiller, J. Daxenberger and I. Gurevych	2021	Stance Detection Benchmark: How Robust is Your Stance Detection?	BERT	F-score of 71.62%.	Not used for Afan Oromo texts.
Tadesse, S.	2023	Political stance detection on amharic text using machine learning	RF, LR, and SVM With word2vec,& TF-IDF, N-gram, and BOW	Acquire an F-score of 82%. Using SVM with the TF-IDF	Used exclusively for Amharic texts; not for Afan Oromo or other regional languages

Table 1: Summary of related works and Research gaps

CHAPTER THREE: METHODOLOGY

3. Introduction

The methodical and arranged process of gathering data, evaluating and interpreting information, and formulating conclusions in light of the findings is known as research methodology (Kapur, 2018). The specific techniques used in a research process to accomplish the study's goals are referred to as a research methodology. It comprises a collection of methods, approaches, and resources utilized to carry out research in a specific area. The validity, ethics, and dependability of study findings are all enhanced by a clearly established research technique. This usually include deciding on the research issue, suitable study design approaches, data gathering and analysis strategies, and suitable sample sizes and sampling methodologies. Methods for data collection, analysis, and ethical considerations should also be included of the process (Kaplan, B., & Maxwell, J. A., 2005). This chapter will examine the use of deep learning methods for political position detection and classification. In particular, we will suggest a method that entails generating a fresh dataset of opinions collected from social media, which we subsequently use as the main source of data for our categorization system.

3.1 Data Collection

We used datasets with ground truth labels for positions on the chosen issue in order to analyze stances on social media sites. Facebook will be used to collect these datasets from various social media sites. Using the Facebook API, a developer account will be created, and the Facepager text processing tool will be used to scrape the official Facebook accounts of the Oromo Federalist Congress (OFC), Oromo Liberation Front (OLF/ABO), and Prosperity Party (Paartii Badhaadhina). Face pager is software that uses web scraping and APIs to extract publicly accessible content from many online sites, such as YouTube, Twitter, and others (Jünger, J., & Keyling, T., 2019).

The selection of Facebook as the primary platform for data collection was driven by its unparalleled reach and unique characteristics that align with the objectives of this political stance detection research in Afaan Oromo. Firstly, Facebook exhibits a pervasive adoption rate and high user penetration, particularly in developing regions and parts of Africa, making it a critical hub for online communication and opinion dissemination ((Kende-Robb, M., & Salami, S., (2019))This broad accessibility positions Facebook as a primary arena for public political discourse, enabling a wider segment of the population to express and share their perspectives on various issues.

Secondly, and crucially for this study, major political entities relevant to the Afaan Oromo linguistic community, specifically the official pages of the Prosperity Party, Oromo Federalist Congress, and Oromo Liberation Front, maintain active and dynamic presences on Facebook. These pages serve as significant forums for direct public engagement and the exchange of current political commentary in Afaan Oromo. This active participation ensures a rich, authentic, and high-volume source of data directly relevant to the study's focus on political stance.

Furthermore, the text-centric nature of Facebook comments and posts provides the necessary linguistic data for training and evaluating deep learning models, unlike visually-dominant platforms such as Instagram or TikTok. While other text-based platforms like Twitter exist, Facebook often facilitates more extensive and nuanced discussions through its comment threads, and in the Ethiopian context, tends to command a broader and more diverse user base actively engaged in local and national political debates (Abdi, A., Gebresilassie, H., & Shiferaw, T., 2019,2020). Consequently, Facebook offers an unparalleled repository of public political opinion within the Afaan Oromo linguistic landscape, making it the most suitable and comprehensive source for this research.

This process involved collaboration with a leading Afaan Oromo linguistic expert, Mrs. Sinkinesh Mekuria (MA), who provided invaluable guidance throughout the data collection phase, ensuring grammatical accuracy and appropriateness of the collected materials. Furthermore, a rigorous annotation process was conducted, involving other Afaan Oromo experts, to accurately label each text as either "Against" or "Supporter," thereby creating a high-quality annotated corpus suitable for training deep learning models. This robust dataset, coupled with advanced deep learning architectures, aims to develop a highly accurate and efficient Afaan Oromo political stance detection system. The study also explores innovative data annotation strategies tailored to the linguistic nuances of Afaan Oromo, addressing challenges such as the identification and handling of ambiguous expressions and the incorporation of contextual information crucial for accurate stance classification.

The data was collected from 2018 to 2024; this period was particularly relevant due to several key political events and societal changes in Ethiopia. This timeframe includes the election period, during which there were notable political developments, such as the appointment of Dr. Abiy Ahmed as the new Prime Minister of Ethiopia. Additionally, the COVID-19 pandemic led to the

postponement of elections, creating various challenges for political parties in Oromia. Throughout this period, political parties, including the Prosperity Party, the Oromo Liberation Front (OLF), and the Oromo Federalist Congress (OFC), actively communicated their stances to the Oromo people via social media platforms, particularly Facebook. This context is essential for understanding the dynamics of political discourse during this time and the relevance of the data collected for our research on stance detection. Overall we have gathered 10,136 Afaan Oromo texts in order to accomplish our objective. Out of total 10,136 datasets annotated, 5136 of them were labeled as “Supporter”, and the rest 5000 were labeled as “Against”.



	Post	text	stance
0	mariin mana maree bakka bu'oota uummataa 6ffaa...	sagantaa ashaaraa magariisaa bara 2011 tiin gu...	supporter
1	mariin mana maree bakka bu'oota uummataa 6ffaa...	dr abiyyi akka biyyaatti tokkummaan obbolummaa...	supporter
2	mariin mana maree bakka bu'oota uummataa 6ffaa...	amanna masjiida kenya ya abiichu akuma wayii m...	Against
3	mariin mana maree bakka bu'oota uummataa 6ffaa...	jijiramaa attin fiddee ethiophiaa qofaa osso t...	supporter
4	badhaadhinni qormaatilee gara carraatti jijjii...	xaa'oo qonnaan bulaa dhorkitanii hattoota haad...	Against

Figure 1: Sample of collected data

3.2 Data Preprocessing

Data preparation is the process of transforming data into a format appropriate for the learning task based on the chosen deep learning approach (Zheng et al.,2018). Data preparation is the process of removing data noise to make categorization easier. Text pre-processing, which includes eliminating unnecessary characters, tokenizing, eliminating stop words, and stemming transformation using HornMorpho, will be the next stage of data collecting in this study.

3.3 Development environment and tools

Guido van Rossum developed the popular high-level programming language Python in the late 1980s, and, is utilized to create the system. By emphasizing code clarity and simplicity, this language enables programmers to construct apps quickly. Python code, like all high-level programming languages, is analogous to English, which is not directly comprehensible by computers. Installing a specialist program called the Python interpreter is required in order to interpret Python programs. We can't write, test, and run our Python code until this installation is complete (Maru Tesfaye Addis, 2022). Several open-source data science tools were used to create the proposed system's prototype. We selected these technologies because they are open-source and

have a large, vibrant community of supporters and contributors. We used Pandas, Matplotlib, SciPy, Keras, WordSpace, and Scikit-learn with TensorFlow as the backend to build the layers and the model. With its extensibility, modularity, and ease of use, the Keras deep learning API, which is based on TensorFlow, seeks to speed up experimentation.

3.4 Evaluation Metrics

The procedures used to determine the efficacy or worth of a specific initiative, program, good, or service are referred to as evaluation techniques. These methods assist discover areas for development and are used to assess if a given project is successful in reaching its intended goals (Siau, K., et al., 2011).

The fundamental performance evaluation techniques used in this research are accuracy, precision, recall, and F-measure.

Measure	Formula
Recognizability rate and accuracy	$\frac{True\ Positive + True\ Negative}{Total\ positive + Total\ Negative}$
Misclassification rate or error rate	$\frac{False\ Positive + False\ Negative}{Total\ Positive + Total\ Negative}$
Sensitivity(true positive rate, recall)	$\frac{True\ Positive}{Total\ positive}$
Specificity(or true negative rate)	$\frac{True\ Negative}{Total\ Negative}$
Precision	$\frac{True\ Positive}{True\ Positive + False\ Positive}$
F- score	$\frac{2 \times Precision \times Recall}{precision + Recall}$
F_{β} , where β is non- negative real number	$\frac{(1 + \beta^2) \times Precision \times Recall}{\beta^2 \times precision + Recall}$

Table 2: Confusion Matrix with the evaluation metrics

True Positives (TP), which are properly recognized positive tuples.

True Negatives (TN): These are the negative tuples that the classifier picked up properly.

False positive (FP) refers to negative tuples that have been incorrectly categorized as positive.

False negatives (FN) happen when positive tuples are mistakenly classified as negative.

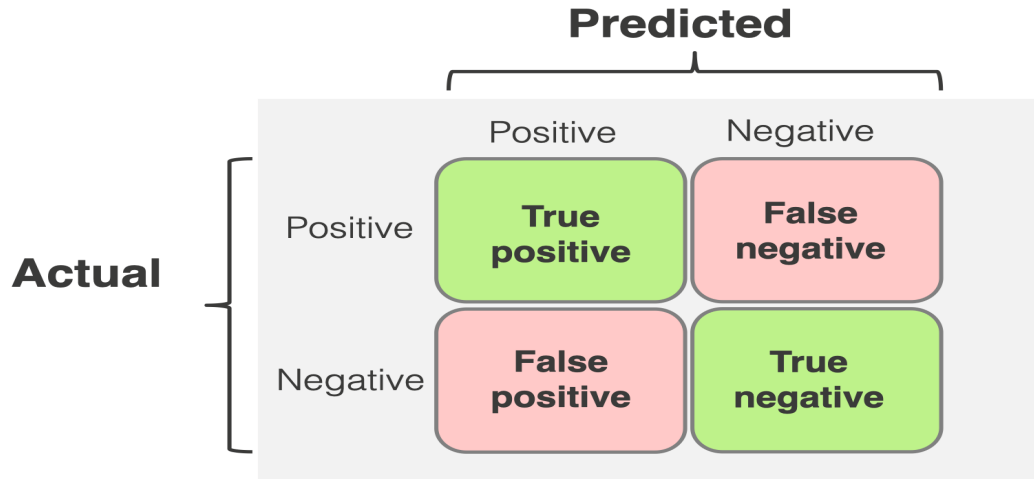


Figure 2: confusion matrix

Evaluation metrics can be stated as follows:

Accuracy: The ratio of accurately predicted labels to the total dataset is known as **accuracy**.

$$\text{Accuracy} = \frac{\text{True Positives (TP)} + \text{True Negatives (TN)}}{\text{Total}}$$

Precision: The number of accurate positive forecasts is measured by precision. The following formula can be used to determine system precision:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True positives (TP)} + \text{False positives (FP)}}$$

Recall: is the ratio of instances that should have been in a class to all of the examples that actually belong in the class.

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True positives (TP)} + \text{False Negatives (FN)}}$$

Precision and **recall** on the positive class are balanced in the **F1-score**.

$$\text{F1-score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}$$

We evaluated the model using evaluation metrics like Accuracy, precision, recall, and F-measure

CHAPTER FOUR: DESIGN OF THE STUDY

4. Introduction

Research design aims to maintain a balance between procedural efficiency and alignment with the study's goals by setting standards for data collection and analysis. You have the option to decide on the study design to employ, the method for collecting data from your participants, the criteria for selecting participants, the approach to analyzing the gathered data, and the way you will present your findings using a research design. Explaining the rationale behind each decision that affects your answers to the "how" of the inquiry is another need of your study design. You must use the examined material to critically support your claims and arguments. It's also essential to reassure yourself and others that the course you've recommended will yield genuine and reliable results (Kumar & Ranjit, n.d.).

"Design science" is an interdisciplinary field of study that aims to provide ideas, methods, tools, and knowledge for designing and evaluating processes, systems, and objects. It focuses on creating and applying design concepts and methodologies to difficult problems in a range of domains, such as information systems, engineering, and healthcare (Vom Brocke *et al.*, 2020). Formulating a problem statement, looking into potential solutions, creating and evaluating a design artifact, and sharing information and findings are all steps in the methodical approach to problem-solving known as design science. It is frequently combined with other research techniques, such case studies and empirical investigations, to produce creative and useful answers to real-world issues (Papalambros, 2015).

A methodology called Design Science Research (DSR) is used to create and assess novel products to solve particular issues (Teixeira, J. G., et al., 2019). The stages of Design Science Research (DSR) include identifying the problem, evaluating the value of a solution, and then establishing realistic goals for the solution. Additionally, it entails the design and production of a functional artifact, its subsequent use in instances of problem-solving through demonstrations, and a comprehensive assessment that gauges its efficacy in relation to the original goals. DSR offers a methodical approach to creating and assessing artifacts to address certain issues. DSR could be utilized to create a novel solution to a critical issue in the context of "Political stance detection for Afaan Oromo text using Deep learning approach."

This chapter will cover the creation of the model that the researcher would create using a deep learning approach for the proposed political stance recognition system. The creation of a deep learning-based political position recognition system for Afaan Oromo text is generally summarized in this chapter. It breaks down a starting raw data set into more manageable processing groups and covers the system architecture, data preparation techniques, in-depth descriptions of deep learning methods, and the feature extraction approach.

4.1 Design of the System

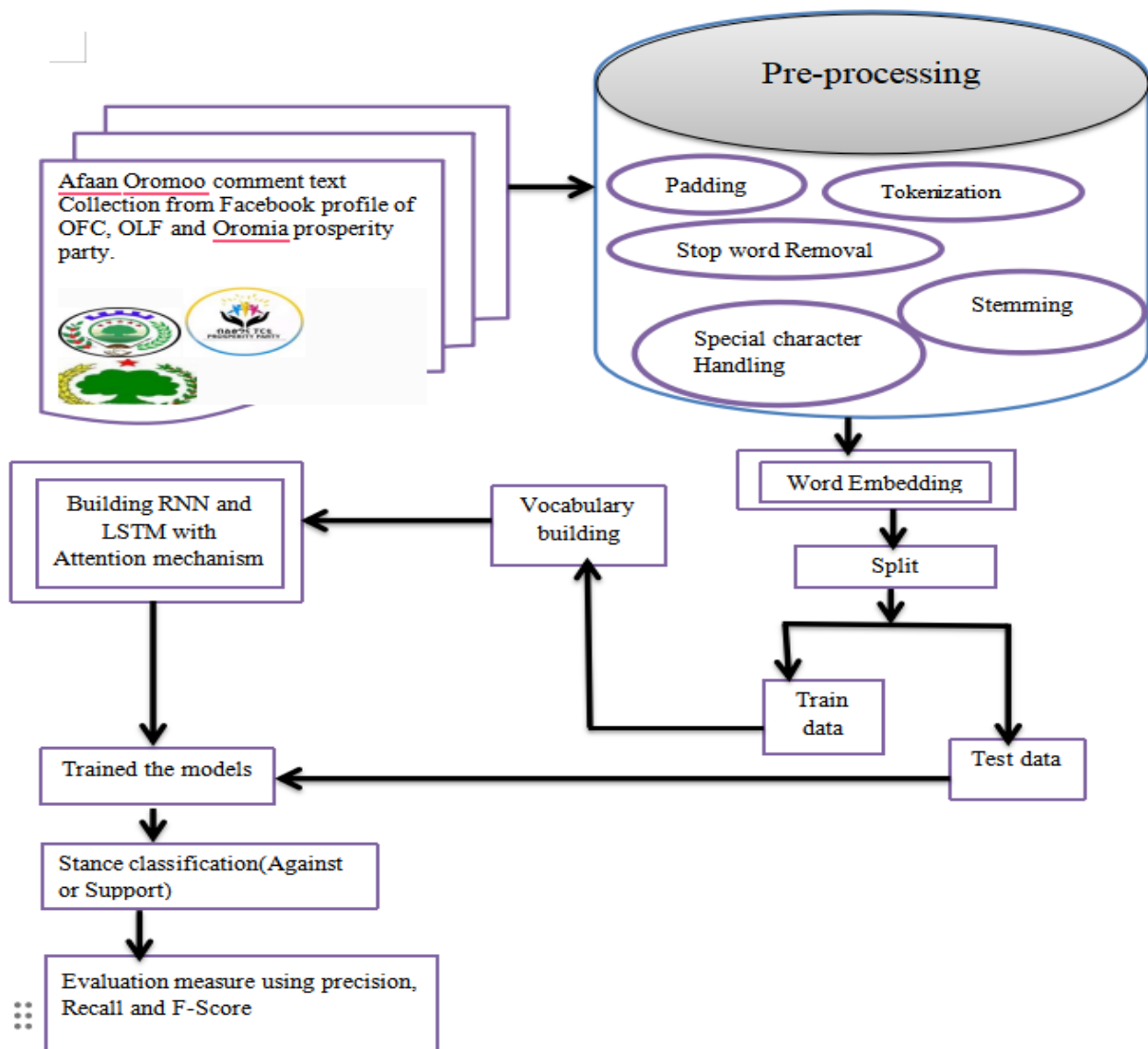


Figure 3: The System architecture

4.2 Preprocessing

According to (Kannan, S., & Gurusamy, V. ,2014)Any NLP system must include text pre-processing because the characters, words, and sentences that are identified at this point serve as the building blocks for all subsequent processing steps, from applications like information retrieval and machine translation systems to analysis and tagging tools like morphological analyzers and part-of-speech taggers. Text documents are pre-processed by a series of operations. Since the text data frequently includes unique formats, such as date and number formats, Common terms like pronouns, articles, and prepositions that are unlikely to help with text mining can be eliminated.

Text preprocessing is necessary in NLP systems in order to:

1. Minimize the size of the text documents' indexing (or data) files

- I) In a given text document, stop words make up 20–30% of the entire word count.

- II) Stemming can cut the indexing size by 40–50%.

2. To increase the IR system's efficacy and efficiency

- I) Stop words can confuse the retrieval mechanism and are not helpful for text mining or searching.

- II) Stemming, this is used to match terms that are similar in a text document.

4.2.1 Tokenization

Breaking up a stream of text into meaningful units called tokens—which might be words, phrases, or symbols—is known as tokenization. The purpose of tokenization is to analyze the words in a sentence (Rai, Abigail, and Samarjeet Borah, 2021). The token list is used as input for additional processing, such parsing or text mining. Both well and linguistics benefit from tokenization, which is a type of text segmentation. Additionally, the tokenizer may accommodate document consistency. Finding the relevant keywords is the primary application of tokenization. Inconsistencies may arise from disparate time and numerical formats. Another issue is that acronyms and abbreviations need to be changed to a uniform format.

Challenges in Tokenization

Language-specific tokenization challenges differ. The phrase "space delimited" refers to the fact that most words in languages like English and French are separated from one another by white spaces. Words in languages like Chinese and Thai are said to be unsegmented because they lack distinct boundaries. In order to tokenize sentences in unsegmented languages, more lexical and morphological information is needed. Additionally, the writing system and word typography have

an impact on tokenization. Language structure can be divided into three groups:

Isolating: Words don't break apart into bits and pieces. For instance, Mandarin Chinese

Words that are agglutinative split into smaller pieces. Tamil and Japanese, for instance.

Inflectional: There are unclear and ambiguous grammatical meaning boundaries between morphemes. Latin is one example.

4.2.2 Stop Word Removal

Although many terms in papers are employed in sentences, they are basically used to join words. useless. It is well known that stop words don't add anything to the content or context of written papers. Because they appear so frequently, their existence in text mining makes it difficult to comprehend the documents' substance. Stop words are common words like "and," "are," "this," and so on that are used a lot. When it comes to document classification, they are useless. Thus, they need to be taken out. However, it is challenging and inconsistent among textual sources to compile such a list of stop words. Additionally, this procedure lowers the amount of text data and enhances system performance. These terms, which are not required for text mining applications, are present in every text document.

4.2.3 Stemming

Stemming is the process of integrating a word's several forms into a single, shared representation called the stem. It is possible to condense the words "presentation," "presented," and "presenting," for instance, to the word "present." The idea behind this method, which is frequently employed in text processing for information retrieval (IR), is that asking a question that includes the word "presenting" suggests that the user is interested in documents that also contain the phrases "presentation" and "presented." (Singh, Jasmeet and Gupta, Vishal, 2016)

There stemming are mainly two errors in

1. Excessive stemming
2. Under stemming

If two words with distinct stems are stemmed to the same root, this is known as over-stemming. Another name for this is a false positive.

When two words that ought to be stemmed to the same root are not, this is known as under stemming. Another name for this is a false negative.

4.2.4 Handling special characters

Data cleaning is the process of identifying and eliminating duplicate, erroneous, or empty entries from a group of records to prepare the data for analysis (Ganti, Venkatesh, and Anish Das Sarma, 2022). News websites that post articles in Afaan Oromo provided the corpus for our investigation. After reviewing the information, we clean the text using the language features. Among other things, we removed quote marks, special characters, user names, and URLs. Any texts that weren't in Afaan Oromo were also removed.

4.2.5 Padding

Sequential machine learning in the real world allows dataset entries to be of any length. The statements in our instance include a variety of tokens. Due to the limited length of the sentence inputs in our model implementation, we will either need to split up each dataset phrase or fill it with a dummy element of a similar size. This stage of preprocessing is often called "padding." By padding shorter sequences with special tokens typically zeros to match the length of the longest sequence in the dataset, we can give our model consistent input sizes.

4.3 Encoding

A popular type of input data for deep learning and machine learning models are integer vectors. Text representation sometimes referred to as word representation, can be approached in a variety of ways. There are several approaches to text representation, often known as word representation (Grzegorzczuk, 2019). Word representation can be used to convert the written text into an output that can be read by a computer. Simple writing can be understood by people but not by computers. Raw text must be encoded with numerical values. The process of turning text into a numerical representation is known as text encoding. TF-IDF, BOW, and Vector Space Model are popular word representation techniques. However, these methods have trouble communicating semantics in programs. BOW, for example, has two disadvantages: high dimension and high sparsity. These defects are large and do not show the syntactic structure of the programs (Yu et al., 2021). We used word embedding for word denotation in order to effectively extract semantic information from programs.

4.4. Word Embedding

NLP has demonstrated the power of deep learning. Deep learning is frequently used for Part-of-Speech (POS) labeling, character creation, and word embedding. Word embedding models are frequently trained on large text corpora using Word2Vec, GloVe, Fast Text, and Keras embedding layer techniques. In order to objectively identify word similarities, Word2Vec learns the context of nearby words (Asudani *et al.*, 2023). Word2Vec creates vectors, which are scattered numerical representations of word characteristics, such as the context of a word. After being trained using a corpus of text as input data, Word2Vec produces a set of word vectors, or embeddings, as the output model.

In addition to useful capabilities like vector arithmetic, the produced word embedding includes spatially encoded word meanings and relationships. Prior to producing the individual word representations, the system builds a vocabulary using the input training data. Unlike other vectorization techniques, we are not first producing a vector for every page. However, the output dataset for Word2Vec is the unique set of words in the corpus that are each assigned a vector following training on the input corpus. Each vector in this output contains the context (or usage) of the word. Word2Vec uses a feed forward Neural Network Language Model (NNLM) to build these contextual word vectors in an unsupervised learning fashion. In practice, this two-layer neural network was trained using SGD (Gibson and Patterson, n.d).

The important task of converting incoming text input into word embedding that convey the semantic meaning of the words is carried out by the embedding layer. These word embedding are generated throughout the training process and serve to depict words in a continuous vector space. The Keras framework offers an embedding layer to represent words as distinct integers. The word embedding framework is utilized to convert the output from the sequence padding element into a real dense format for our research objectives. In this system, the main goal of the embedding layer is to transform the input size into a lower-dimensional space. The keras library offers an embedding layer to represent words using distinct integers. Consequently, the word embedding structure is utilized to transform the output of the sequence padding element into a tangible dense representation for our research. The main function of the embedding layer in this architecture is to translate input size into a low-dimensional space.

In our research, we prefer using word embedding, specifically Word2Vec, for detecting political stances in Afaan Oromo text, as opposed to traditional methods like Bag of Words (BOW) and Term Frequency-Inverse Document Frequency (TF-IDF). The primary bottlenecks of BOW and TF-IDF lie in their inability to capture the semantic meaning and contextual relationships between words. BOW treats words as independent entities, disregarding their order and context, which can lead to a loss of important information. Similarly, while TF-IDF provides a measure of word importance, it still fails to account for the nuances of word meanings and their interrelations. Word2vec's architectures, CBOW and Skip-gram, allow for more dynamic and context-aware embedding, enabling the model to discern subtle variations in meaning and usage essential for stance detection.

By leveraging Word2Vec, we can enhance the model's ability to detect subtle political stances in Afaan Oromo text, ultimately leading to more accurate and meaningful analyses.

The weight representation and semantic links between the terms in our dataset are shown in the picture below.

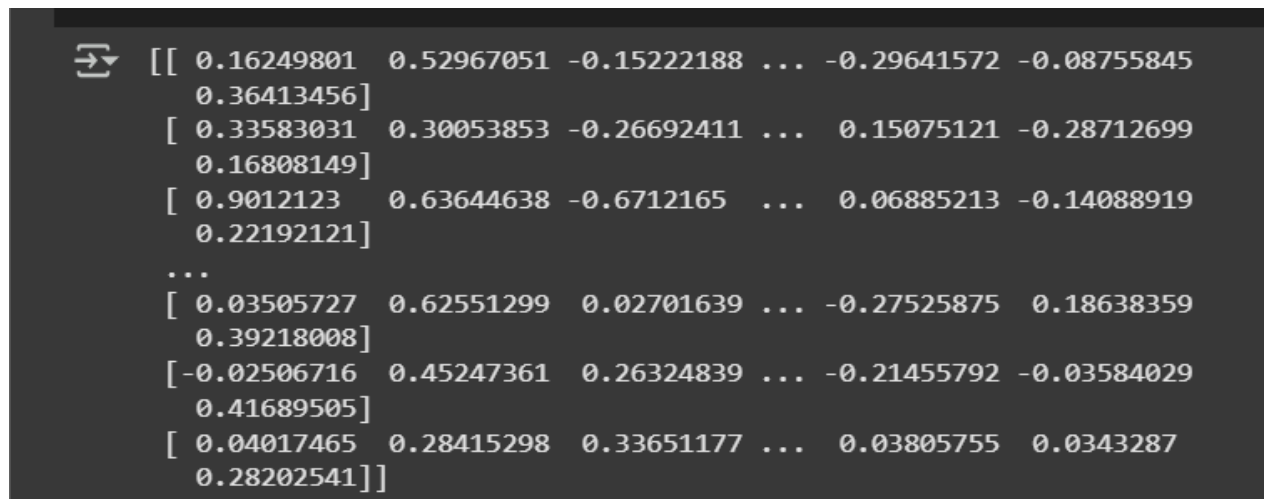


Figure 4: Weight from model

4.5. Building model

4.5.1 Deep Learning

Method of artificial intelligence known as neural networks with numerous hidden layers has been in and out of style for more than 70 years, and the term "deep learning" refers to machine learning algorithms. Neural networks were first proposed by Walter Pitts and Warren McCulloch in 1943,

marking the beginning of deep learning. Mathematician Walter Pitts and neurophysiologist Warren McCulloch created a computer model based on the neural networks of the human brain. Warren McCulloch and Walter Pitts used a combination of mathematics and algorithms to simulate the process, which they named threshold logic. The MCP neural model was their model. The MCP Neural Model and the modern perceptron share similarities, but they differ in many important ways. For example, the MCP Neural Model was originally constructed as electrical circuits, and its weights are fixed, whereas the weights of the modern perceptron are adjustable. For data processing, human speech comprehension, and visual object recognition, Deep Learning (DL) employs several layers of algorithms. As information moves through each layer, the output of the previous layer becomes the input for the next layer. Rather than being a single technique, deep learning is a class of algorithms and topologies that may be used to a variety of scenarios. Despite not being new, deep learning is expanding quickly because it uses graphical processing units (GPUs) to speed up its execution and heavily layered neural networks. Big data has also contributed to this increase. Since deep learning relies on supervised learning algorithms those that use example data to train neural networks and reward them based on their performance the more data, the better these deep learning structures may be constructed. Among the several types of deep learning architectures include recurrent neural networks (RNNs), convolutional neural networks (CNNs), deep belief networks (DBNs), long short-term memory (LSTM)/gated recurrent units (GRUs), and deep stacking networks. (Sifen Dadi Degefa, 2021) (DSNs).

4.5.2 Neural Networks

Neural networks are a computational model that shares some traits with the human brain, where many basic units function simultaneously without a centralized governing body, claim Josh Patterson and Adam Gibson (August 2017). The main mechanism for the long-term retention of information in neural networks is the connections between these units. The basic mechanism by which the neural network learns new information is the alteration of these connections.

A network's architecture can be described (somewhat) by the following parameters: • Number of layers

- Number of neurons
- The type of connections that exist between layers.

The feed forward multilayer neural network is the simplest and most commonly utilized neural

network. This structure includes one input layer, one or more hidden layers, and a single output layer. Every layer contains a varying number of neurons, and all layers are completely interconnected. The Perceptron The perceptron functions as a binary classifier based on a linear model, exhibiting a simple relationship between input and output, as illustrated in the accompanying image. The "net input" is sent to a step function with a predetermined threshold value of 0.5 after n inputs have been added together and multiplied by their respective weights.

This function will output a single binary value (0 or 1) that is real-valued, depending on the input.

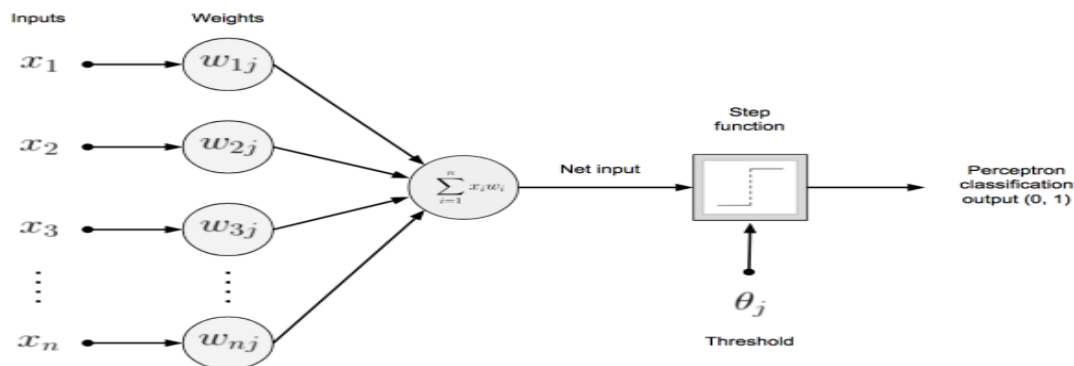


Figure 5: Single-layer perceptron

The perceptron learning procedure is employed to modify the weights of the perceptron model until all input records are accurately recognized. Even if the input for learning is not linearly separable, the method will still operate effectively. When we can identify the parameters of a hyperplane that effectively divides the dataset into its two categories, we state that the dataset is linearly separable. Feed-Forward Multi-Layer Systems Neural networks consist of multilayer feed-forward architectures that include input, output, and multiple hidden layers. Each layer contains a minimum of one artificial neuron. These artificial neurons, like their progenitor perceptrons, have distinct activation functions based on the network layer's particular mission.

Later in the chapter, we'll look more closely at the many kinds of multilayer perceptrons. For now, let's examine this advanced artificial neuron, which was developed to address the limitations of the single-layer perceptron.

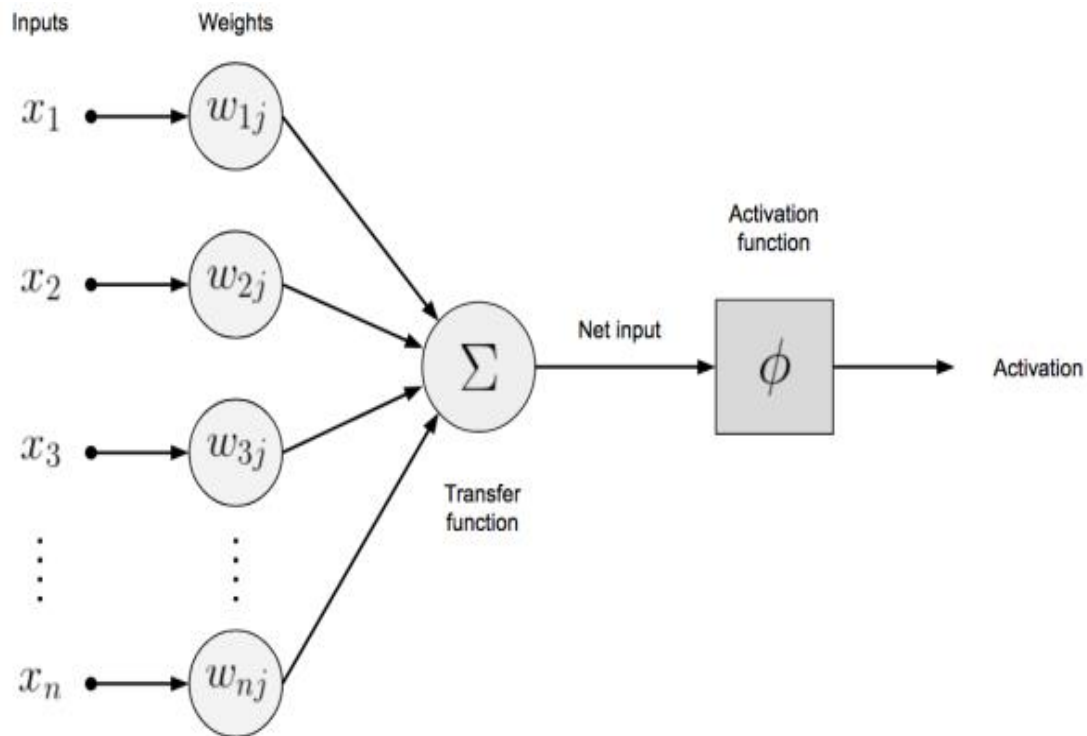


Figure 6: Artificial neuron for a multilayer perceptron

Artificial neurons are characterized by the transformation (activation function) they employ to generate output and the continuous or binary input they can process. Each neuron in a layer performs the same activation function.

Connection weights: These factors scale (increase or decrease) the input signals to a specific network neuron in a neural network. These are the edges of the mathematical graph, the lines or arrows that connect points in typical neural network representations. The letter "w" is frequently used to denote connections in mathematical representations of neural networks.

Biases: To ensure that a minimum of specific nodes per layer are active regardless of signal strength, scalar values known as biases are applied to the input. By providing the network with a task to perform in the event of a weak signal, biases enable learning. They allow the network to experiment with novel interpretations or behaviors. Biases, which are commonly represented by the letter b, are modified throughout the learning process, much like weights.

Functions of activation: By altering inputs, weights, and biases, activation functions regulate the behavior of artificial neurons through forward propagation. Neural networks utilize nonlinear

transforms to convert data into useable ranges. To show that they are active, active neurons send nonzero values to other neurons.

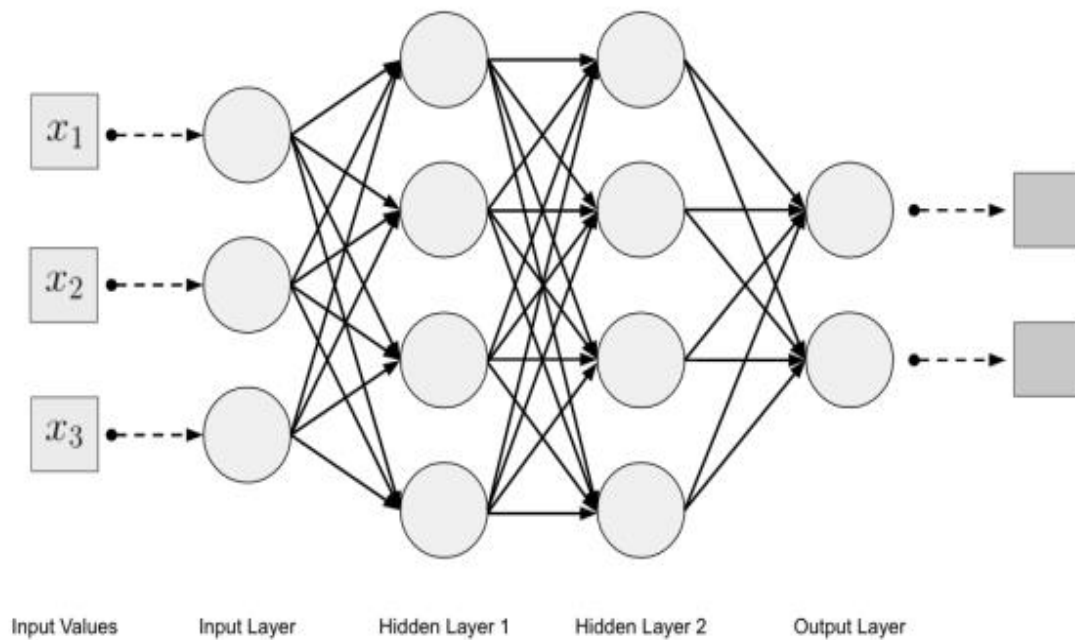


Figure 7: Fully connected multilayer feed-forward neural network topology

Input layer: The input layer provides our network with vectors, or the input data. Often, the count of neurons in an input layer matches the count of input features in a network. Input layers are directly succeeded by one or more hidden layers (further details provided in the next section). Standard feed-forward neural networks possess input layers that are entirely linked to the subsequent hidden layer, although this may not apply in other network architectures. A feed-forward neural network includes one or more hidden layers. Neural networks encode the information obtained from the raw training data by assigning weight values to the connections between the layers. Neural networks lacking hidden layers are unable to represent nonlinear functions, as evidenced by the shortcomings of single-layer perceptron networks.

Output layer: The output layer provides our model's prediction or response. The nature of the output, whether it is a continuous value (regression) or a set of probabilities (classification), is determined by the neural network's setup. This is influenced by the activation function applied to the neurons in the output layer. Common choices for classification tasks in this layer include sigmoid or softmax activation functions. **Interconnections between layers:** In a fully integrated feed-forward network,

the connections between layers are made up of outgoing connections from each neuron in the layer before to every other neuron in the layer after. We gradually modify these weights while our computer uses back propagation to determine the optimal solution.

Analytically, one can think of the weights as the parameter vector discussed in the previous section under linear algebra. This means the process of machine learning is a process of optimizing this parameter vector (referred to as "weights") for minimizing error. According to (Sifon Dadi Degefa, 2021) deep learning algorithms are explained as follows:

4.5.3 Convolutional neural networks

Kunihiko Fukushima was the first to use convolutional neural networks (CNNs), developing neural networks with several pooling and convolutional layers. In 1979, Fukushima developed the Neocognitron Kunihiko artificial neural network. It has a multi-layered, hierarchical architecture. The computer was able to learn to identify visual patterns because of the multi-layered, hierarchical design. CNNs have a unique architecture that makes them very well-suited for picture classification. Convolutional networks can be trained quickly because of its architecture. Consequently, this aids in the training of deep, multi-layer networks, which excel at picture classification. Deep convolutional networks are now used in most neural networks for image identification. This design makes use of three key ideas: local receptive fields, shared weights, and pooling. A tiny portion of the input (or preceding layer's) neurons in local receptive fields will be connected to each neuron in the first (or any) hidden layer. Shared weights indicate that each local receptive field is given the same weights and bias. This demonstrates that, despite their various placements within the input image, every neuron in the hidden layer detects the exact same feature. In addition to the convolutional layers previously mentioned, convolutional neural networks also contain pooling layers.

Typically, these pooling layers are applied right after convolutional layers. Simplifying the information in the convolutional layer's output is the aim of pooling layers. Ten to twenty hidden layers and billions of connections between units are features of recent and popular convolutional network topologies. The terms 1D, 2D, and 3D convolutions, which represent one-dimensional CNN, two-dimensional CNN, and three-dimensional CNN, respectively, can be used to categorize convolutional neural networks.

CNNs having a single dimension or a kernel that slides along a single dimension are known as one-

dimensional CNNs, or Conv1D CNNs. Conv1D CNNs are appropriate for time series data, such as voice or sound, because of this feature. CNN in two dimensions is typically applied to image data. It is known as a two-dimensional CNN because the kernel moves along two dimensions on the data.

The kernel slides in three dimensions in three-dimensional CNNs, also known as Conv3D CNNs. Conv3D CNNs are mostly utilized with 3D image data, including MRI data.

4.5.4 Recurrent neural networks (RNNs)

The other architecture for deep learning is recurrent neural networks (RNNs). Other deep learning designs are constructed using RNN, one of the basic network architectures. Deep learning architectures are abundant in recurrent neural networks. RNNs can use their internal state memory to process input sequences of different lengths. Thus, they can be applied to tasks such as unsegmented, linked handwriting recognition and speech recognition. Each and every piece of information that is processed, recorded, saved, and used to determine the outcome. Additionally, the recurrent network may have connections that feedback into the same layer or even into previous levels. They are able to preserve their recall of previous inputs and promptly resolve issues thanks to this feedback. RNNs are particularly helpful in domains where the order in which information is delivered is required. Even deep feed forward networks proved to be more challenging to train than earlier RNN models. This resulted from unstable gradient issues like vanishing and bursting gradients. As the gradient spreads back via layers, it may become increasingly smaller. Because of this, learning in the early layers is very slow. In fact, the issue becomes more severe in RNNs since gradients propagate backward over time rather than just layers.

The gradient may become very unstable and challenging to learn from when the network operates for an extended period of time. A notion known as long short-term memory units (LSTMs) can be incorporated into RNNs to overcome the unstable gradient problem. LSTMs make it considerably easier to train RNNs with decent outcomes, and several recent studies use LSTMs or similar concepts.

Two varieties of RNN are available:

1. Bidirectional RNN: The output layer of this architecture can concurrently receive information from past and future states.
2. Deep RNN: The presence of numerous layers allows the DL model to extract more hierarchical information.

4.5.5 Long short-term memory (LSTM) and gated recurrent unit (GRU)

Hochreiter and Schmidhuber created LSTM in 1997. The long short-term memory (LSTM) RNN architecture is used in deep learning with feedback connections. It is helpful for tasks like connected handwriting recognition and speech recognition since it can examine sequential input, such as speech or video. An LSTM is composed of a cell, an input gate, an output gate, and a forget gate. The three gates regulate the information that enters and leaves the cell, and the cell stores values for arbitrary time periods. Short-term, long-term memory. Given that significant events in a time series may have unknown latencies between them, LSTM networks are ideal for categorizing, analyzing, and forecasting time series data. One advantage of LSTM over RNNs is its greater controllability, which leads to better results. An LSTM cell's cell memory unit is its primary benefit above a standard recurrent unit. In addition to adding some of the new information, the cell vector can also include the idea of forgetting some of its previously stored memories. Gated recurrent units (GRUs) are the alternative deep learning architecture. Similar to LSTMs with forget gates but without an output gate, GRUs are a gating technique in recurrent neural networks. But because GRU has fewer parameters than LSTM, it is easier to use and can be learned faster. It was discovered that GRU can do jobs more efficiently than LSTM and that its performance on some tasks, such as speech signal modeling and natural language processing, is comparable. On the other hand, the LSTM can produce better results with more data and be more expressive.

4.5.6 Deep belief networks (DBN)

Featuring numerous hidden layers, DBN is a multilayer network featuring Restricted Boltzmann Machines (RBMs) in each pair of connected layers. A stack of RBMs makes up DBN. DBN is composed of many layers of latent variables, or "hidden units," even if there are links between the layers but not between units within each layer.

4.5.7 Deep stacking networks

An extensive collection of separate networks, each with its own hidden layers, make up a deep learning network (DSN). The DSN sees training as a collection of distinct training problems rather than a single issue since the complexity of training increases exponentially with each layer in a deep learning architecture. Due to their superior performance over standard DBNs, DSNs have gained popularity as effective network designs. Numerous fields can benefit from deep learning techniques,

which excel in solving challenging real-world issues. especially for self-driving cars, speech recognition, and image processing.

As an example, Convolutional Neural Networks (CNNs) are a deep learning architecture that mimics the functions of the human visual cortex. Therefore, it does a great job on any visual recognition task. Special layers known as pooling layers and convolutional layers make up the CNN architecture. Certain visual attributes can be encoded by the network thanks to these layers. Deep learning architectures also use auto encoders. Auto encoders, such as stacked de-noising auto-encoders, are used to clean a partially corrupted output and lower the dimensionality of the feature space. Automatic speech recognition is another field in which deep learning has been successfully used. Effective acoustic and language models are combined in automatic voice recognition. The challenge of speech recognition is based on time series data. For time series sequential data, the ideal architecture is a recurrent neural network (RNN), which has loops in the hidden layer that recall information from the previous time step to forecast the value of the current time step. Different speech rates are handled by RNNs with the use of this method. Temporal dependencies may also be problematic when assessing speech recognition tasks. Depending on the voice recognition problem, longer-term or short-term temporal dependencies can be resolved by LSTM, a form of RNN. RNNs are applicable to many different tasks, including speech recognition, image captioning, and machine translation. To turn continuous and unsegmented data into labeled sequences, the Connectionist Temporal Classification (CTC) is utilized. This means that CTCs address the challenge of labeling sequence data in RNN training when there are more observations than actual labels associated with the observations. Furthermore, it has been demonstrated that the CTC method works well when it is unknown how the input and output labels align. Previously taking weeks, deep learning network training now only takes a few hours because to developments in GPU technology and algorithm development.

4.5.8 Bidirectional Recurrent Networks (BRNN)

The disadvantage of recurrent networks is that, up until a certain sentence point, the state at a particular time unit only knows about previous inputs and is ignorant of future states. For instance, knowledge of previous and upcoming circumstances greatly enhances language modeling outcomes. In the recognition of script, for instance, it is evidently advantageous to employ both past and future symbol knowledge since it offers a more thorough comprehension of the underlying context. Each

of the bidirectional recurrent network's forward and backward directions has a hidden state, $h^{(f)}_t$ and $h^{(b)}_t$. Only communication between the forward and backward hidden states is possible.

The main difference is that the forward and backward states interact in the forward and backward directions, respectively. However, $h^{(f)}_t$ and $h^{(b)}_t$ both get input from the same vector, x_t , and interact with it.

4.5.9 Gated Recurrent Units (GRUs)

Because the GRU does not employ explicit cell states, it can be thought of as a simpler version of the LSTM. Another distinction is that the LSTM uses independent forget and output gates to directly regulate the amount of data altered in the hidden state. A GRU, on the other hand, accomplishes the same objective with a single reset gate. However, because the GRU partially refreshes the hidden state, its basic concept is somewhat comparable to that of an LSTM. The GRU employs update and reset gates to determine how much of the prior timestamp to retain for the following phase, much like the LSTM uses input, output, and forget gates. The GRU requires fewer gates to update between hidden states because it lacks a distinct internal memory.

4.6 Training the model

Starting the training process is the next goal after deciding on the model's structure. The `sparse_categorical_crossentropy` loss function and the Adam optimization technique are used to configure an attention mechanism for the LSTM and RNN networks. After that, Preprocessed news articles are employed to train the model. Once the model has been trained using data, the neural network updates its weights to reduce the loss. During training, hyper-parameters such as batch size, epochs, embedding and latent dimensions, optimization method, loss function, and learning rate are carefully tuned. We have amassed 10136 comments in our corpus to reach our objective. 80% of the 10136 datasets are therefore used for training, while 20% are used for validation.

4.7 Stance Classification

After the model has been trained, use it to forecast the position of newly unseen text. Probabilities or stance labels (such as "supporter" and "against") are produced by the model. After stance categorization was finished, we evaluated the model using evaluation metrics like precision, recall, and F-score.

CHAPTER FIVE: RESULT AND DISCUSSION

5.1. Overview

Below is a description of our suggested model's experimental setup and evaluation. The experimental setup, training process, tools, dataset preparation, and evaluation metrics are all described. The results of the model analysis and the suggested model are shown. This chapter further contains a discussion of the findings.

5.2. Data gathering and preparation

Data is by far the most crucial and essential resource required for any type of NLP research. Large amounts of data are necessary for a deep learning technique, as NLP has long recognized. Most NLP applications rely on the quality and availability of pertinent data to succeed or fail. Since the majority of state-of-the-art research uses a corpus-based approach combined with different machine learning techniques, it is clear why data is crucial. As a language representation, there isn't a pre-trained model for low-resource languages like Afaan Oromo. We therefore created a new, but small, dataset of Afaan Oromo texts for our study from three authenticated political party Facebook pages: Oromo Federalist Congress (OFC), Oromo Liberation Front (OLF), and Oromia Prosperity Party. In total, we have gathered 10,136 Afaan Oromo texts in order to accomplish our objective. There are 10,136 datasets in total, of which 80% are used for training and 20% for validation.

5.3. Development Environment

5.3.1. Computer used

The research's host machine is a DESKTOP-QB4O65F running 64-bit Windows 11 with an Intel(R) Celeron(R) CPU N2840 @ 2.16GHz and x64 architecture, a 1TB internal hard drive, and 8GB of random-access memory.

5.3.2. Development Software

The colab notebook, an open-source data science tool, was utilized to create the initial prototype of its system. The use of these notebooks is a foundational basis for analysis, machine learning, and data visualization. Among the essential resources and libraries found in colab notebooks are:

1. **Python:** which is commonly utilized in data analysis and machine learning applications, is supported by Colab Notebooks.

2. **Pandas:** Pandas, a popular Python data manipulation toolkit, provides data structures including Data Frames, which are crucial for preprocessing and data cleaning.

3. **NumPy:** NumPy is an essential part of Python scientific computing that helps with massive multi-dimensional arrays and matrices.

4. **Matplotlib and Seaborn:** These two popular Python data visualization tools let users create different kinds of charts and plots to examine and display data.

The Python machine learning library Scikit-learn provides a wide range of tools for creating and assessing machine learning models.

5. The well-known gradient boosting libraries XGBoost and LightGBM are frequently utilized to create high-performance machine learning models.

6. Deep learning libraries TensorFlow and PyTorch offer resources for building and refining neural networks for natural language processing applications.

Data scientists and machine learning practitioners can use the extensive selection of tools and libraries that are commonly found in colab notebooks to analyze data, model, and showcase their work to the public.

5.4. Implementation

First, as mentioned earlier, Facebook texts were classified according to the Afaan Oromo position. The CSV-formatted data from the Python workbook was converted into a pandas DataFrame with the help of the CSV module. The data frame containing a variety of attributes was used to extract the text and stance columns, which are necessary for further analysis. Ten thousand Afaan Oromo posture and text combinations were combined for preliminary work. Using the Python nltk, regex, and delicious soup libraries, this preprocessing technique included tokenization, stemming, text length management, special letter "sostok" and "eostok,; removing Afaan Oromo stop-words; data cleaning; and punctuation filtering.

The stance and text were the datasets that will be utilized to construct the neural network after preprocessing was finished. The texts are then divided into train and test groups and tokenized successively to generate the vocabularies using the Keras preprocessing tool. Following preprocessing, an embedding layer for encoding words as distinct integers was provided by the

keras package. The output of the sequence padding element of the suggested system is converted into a meaningful dense representation by the word embedding component. Thus, as illustrated in figure 7 below, the first layer of the neural network model in our study is the keras embedding layer, a kind of word embedding model.

Model: "functional_8"

Layer (type)	Output Shape	Param #
input_layer_8 (InputLayer)	(None, 100)	0
embedding_8 (Embedding)	(None, 100, 128)	1,467,520
simple_rnn_7 (SimpleRNN)	(None, 100, 128)	32,896
attention_layer_7 (AttentionLayer)	(None, 128)	129
dense_17 (Dense)	(None, 5)	645

Total params: 1,501,190 (5.73 MB)
 Trainable params: 1,501,190 (5.73 MB)
 Non-trainable params: 0 (0.00 B)

Figure 8: This architecture will act as first layer in our attention based RNN model

The embedding layer of the keras package needs three parameters: the required output dimension, the maximum length of input sequences, and the vocabulary size (input dimension). Finally, utilizing the training hyper-parameters provided in Table 2 below, Keras package with the Tensorflow backend built the attention-based RNN model and the attention-based LSTM.

Hyper-parameters	Explanation	Value
embedding dim	Embedding dimension in both encoders and decoders	300
latent dim	Latent dimension in both encoder and decoder	200
Optimizer	A strategy for reducing the loss function	Adam
Attention mechanism	It is aware of both the lengthy sequence and the important details the decoder will look for when receiving text sequences.	Bahdanau's Attention
Loss_function	Each text output is made mutually exclusive by the decoder, which also turns each text output into a one-hot vector.	sparse categorical crossentropy
Learning rate	When will the model be updated to fix the problem?	0.1
Activation	Determines each node's output given a word or text sequence.	Softmax
Drop out	Reduces over fitting and only improves efficiency	0.4

Table 3: Description of Hyper-parameters

5.4.1. Experiment using Attention-based LSTM

LSTM is utilized for both encoder and decoder training and has long-term dependencies that can fix the Vanishing Gradient problem. In certain deep neural networks, training becomes increasingly difficult when the gradient of the loss function approaches zero. The attention mechanism approach can also be used to tackle other encoder-decoder construction issues. It attempts to highlight some facts while downplaying others. The system processes each word based on its weight, which is allocated to each individual word. In our research, we investigate integrating an attention mechanism into the LSTM model that was influenced by (Bahdanau et al. 2014). Thus, the model can focus on the important input sequences before producing the output. To improve our model's performance, we made the encoder and decoder have maximum text lengths of 200 and 85, respectively. The various batch sizes and epoch settings are shown below, whereas Table 2 above lists the training hyper-parameters. When we set the epochs to 20 and the batch size to 32, we first get the greatest precision for the test dataset of 86.15% with a validation loss of 4.3% and the highest accuracy for the training dataset of 92.57% with a training loss of 2.4%. Second, with a

validation loss of 0.042% and a batch size of 64 and an epoch count of 50, the maximum accuracy for the test data set is 97.5%. Conversely, with a training loss of 0.045%, the highest accuracy for the training dataset is 98.6%.

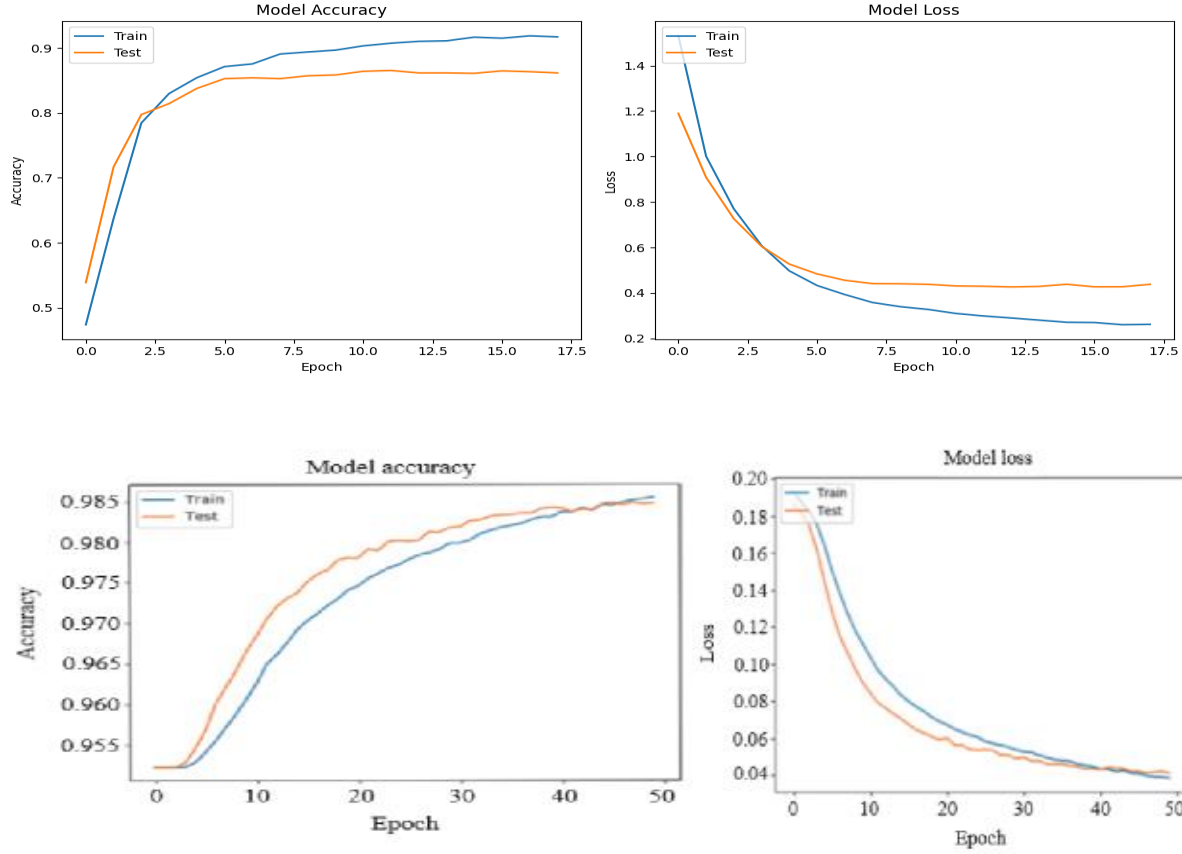


Figure 9: Accuracy/loss Graph of Attention-based LSTM for stance classification.

5.4.2. Experiment using Attention based RNN model

Bahdanau attention and a bi-directional recurrent neural network with LSTM cells were employed in this solution.

Two LSTM cells with initializations make up the encoding layer in this investigation. The outputs are concatenated because the RNN is bidirectional. The training decoder and the inference decoder are two different decoding functions that we use. Two stacked LSTM layers with dropout make up the decoding component. The experiment's use of the Bahdanau attention mechanism (Bahdanau et al., 2014) helps to enhance model training results. The attention mechanism is included into our decoding procedure by the dynamic attention wrapper. Both our training and inference processes start with the state produced by the dynamic attention wrapper. We decide to use the forward state

of the bidirectional RNN even though we have the option to use both the forward and backward states. With the different batch sizes and epoch settings described below, Table 3 above displays all of the hyper-parameters used for training. First, we obtain the highest accuracy for the test datasets of 86.15% with 4.3% validation loss and the highest accuracy for the training datasets of 92.57% with 2.4% training loss when we set the epochs to 20 and the batch size to 32. Second, the maximum accuracy for the test datasets is 93.47% with a validation loss of 2.34% when we set the batch size to 64 and the epoch size to 50. In contrast, the maximum accuracy for the training datasets is 95.82% with a training loss of 1.65%.

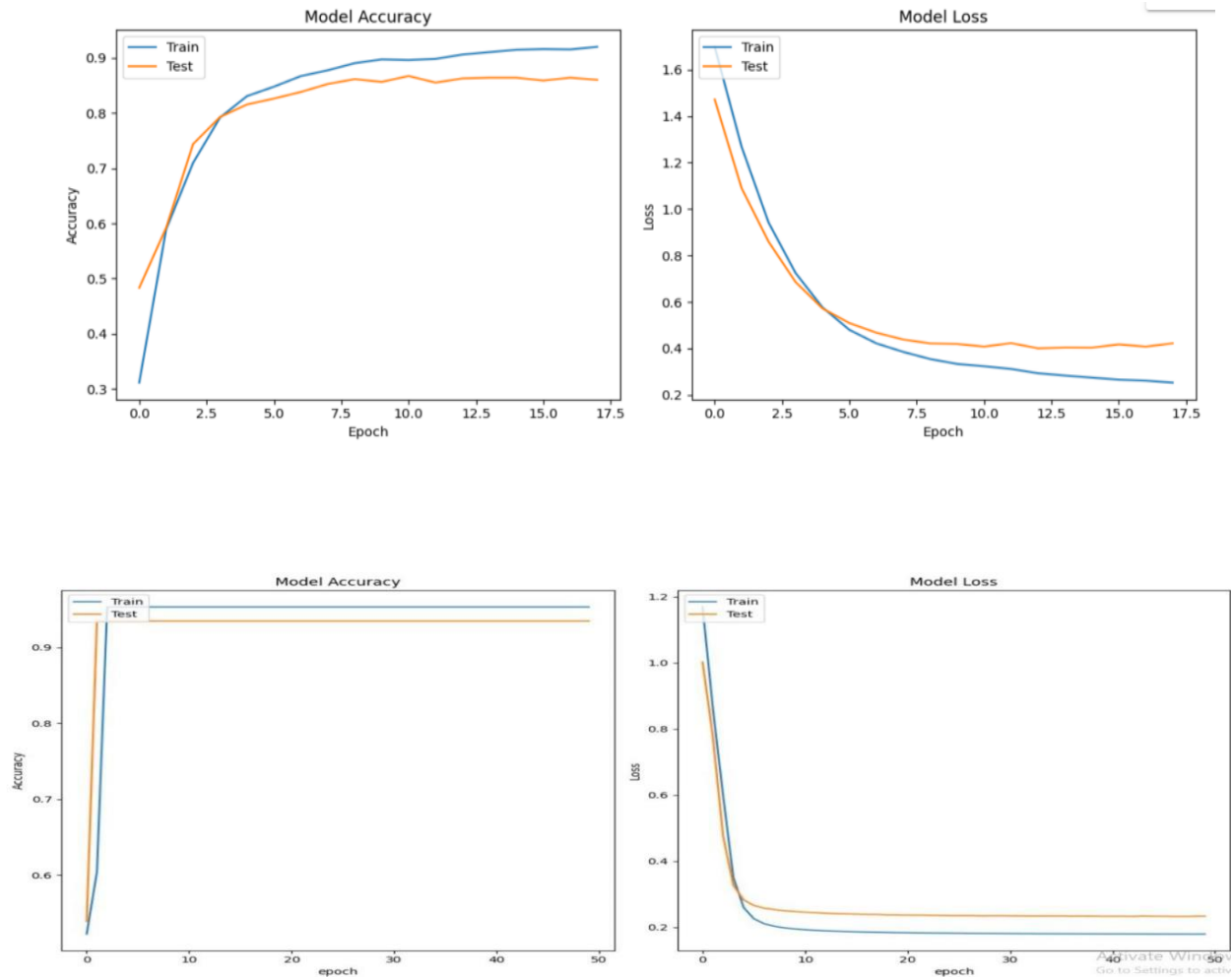


Figure 10: Accuracy/loss Graph of attention based RNN for stance classification.

5.5. Discussion and result

In this study, we used an attention mechanism inspired by Bahdanau et al. (2014) to construct a model of Afaan Oromo text sentiment categorization on Facebook for both the attention-based RNN model and the attention-based model with LSTM. Of the 10136 datasets, 80% were used for training and 20% for validation after the preprocessing methods described at the start of Chapter 4 were finished. We have employed many hyper-parameters to train and validate the model. Using the same hyper-parameters as in Table 3, the performance of each model is displayed in Table 3 below.

Model	(Batch-size, no of epochs)	(Train and Test separation)	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
Attention-based LSTM model	(32, 20)	(80,20)	92.57%	2.4%	86.15%	4.3%
		(70,30)	68.2%	2.96%	66.54%	2.39%
Attention-based LSTM model	(64, 50)	(80,20)	98.6%	0.045%	97.5%	0.042%
		(70,30)	68.77%	1.21%	65.54%	2.20%
Attention-based LSTM model	(32, 100)	(80,20)	80.86%	2.07%	77.99%	3.52%
		(70,30)	66.87%	2.00%	63.32%	2.31%
Attention-based LSTM model	(64, 100)	(80,20)	77.86%	1.34%	75.41%	2.55%
		(70,30)	67.87%	1.90%	64.09%	2.42%
Attention based RNN model	(32, 20)	(80,20)	92.57%	2.4%	86.15%	4.3%
		(70,30)	69.58%	1.96%	64.61%	2.52%
Attention based RNN model	(64, 50)	(80,20)	95.82%	1.65%	93.47%	2.34%
		(70,30)	67.45%	1.90%	65.81%	2.31%
Attention based RNN model	(32,100)	(80,20)	81.59%	1.21%	76.80%	1.45%
		(70,30)	66.62%	1.99%	64.63%	2.33%
Attention based RNN model	(64,100)	(80,20)	81.88%	0.80%	78.54%	1.65%
		(70,30)	64.79%	1.88%	64.80%	2.44%

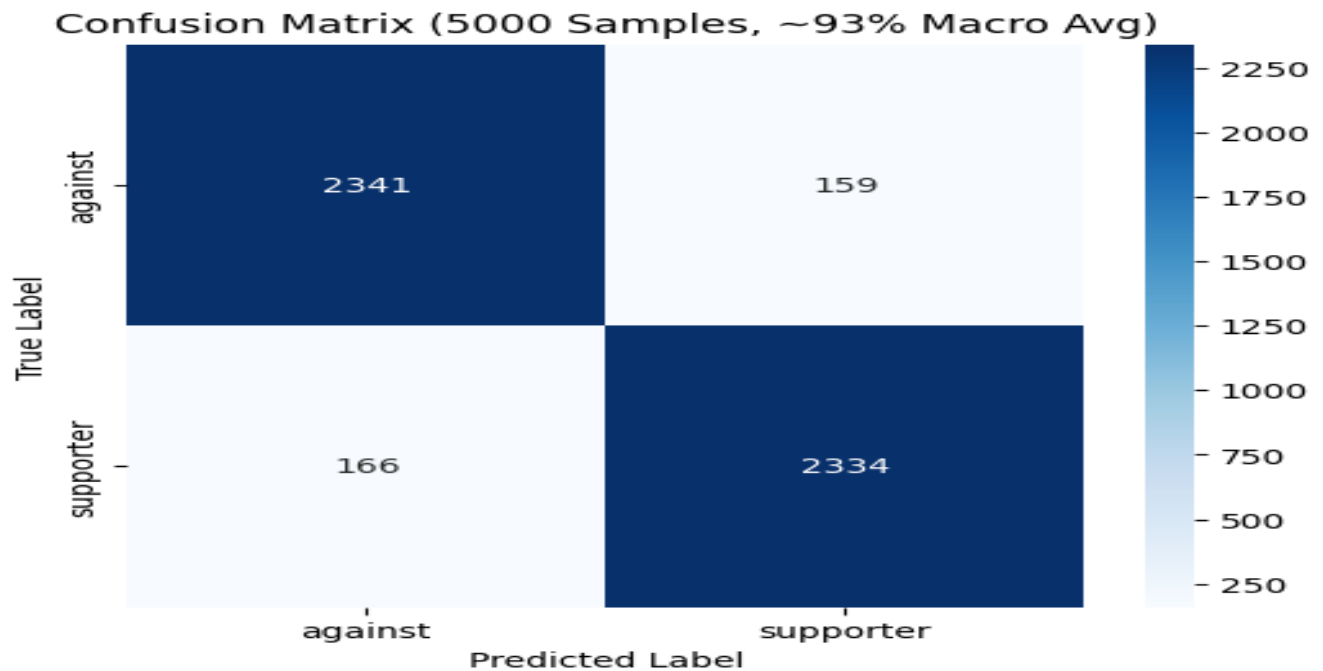
Table 4: Training and performance of the proposed model

Our experimental outcomes reveal that, with all the hyper-parameters defined in the above table 2, the Attention based RNN model worked better, with 95.82% training accuracy and 1.65% training loss and 93.47% validation accuracy and 2.34% validation loss at batch size 64, 50

epochs. From using all of the hyper-parameters stated in Table 2, the attention-based LSTM did better, with 98.6% for training accuracy and 0.045% for training loss and 97.5% for validation accuracy and 0.042% for validation loss with batch sizes of 64 and 50 epochs. From this, we understood that the attention-based LSTM model is better than the attention-based RNN.

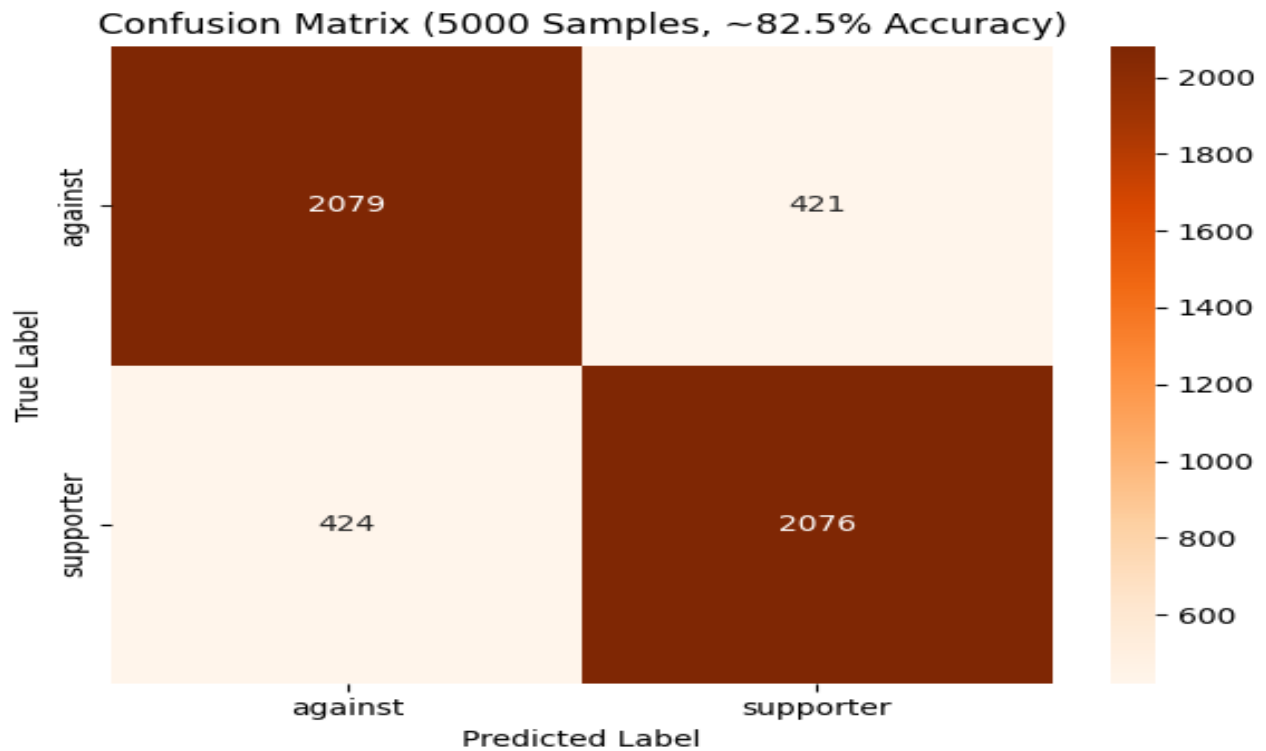
5.6. Performance Evaluation and Validation

The developed model's ability to categorize material that incorporates novel words and phrases not existing in the original while maintaining the key particulars and main ideas is tested in a comment text. Using unknown data as input from our Validation dataset, we evaluate the stance classified by the recommended model using precision, recall, and F-score assessment measures.



Classification Report:				
	precision	recall	f1-score	support
against	0.93	0.94	0.94	2500
supporter	0.94	0.93	0.93	2500
accuracy			0.94	5000
macro avg	0.94	0.94	0.93	5000
weighted avg	0.94	0.94	0.93	5000

Figure 11: Attention based LSTM Evaluation performance



Classification Report:

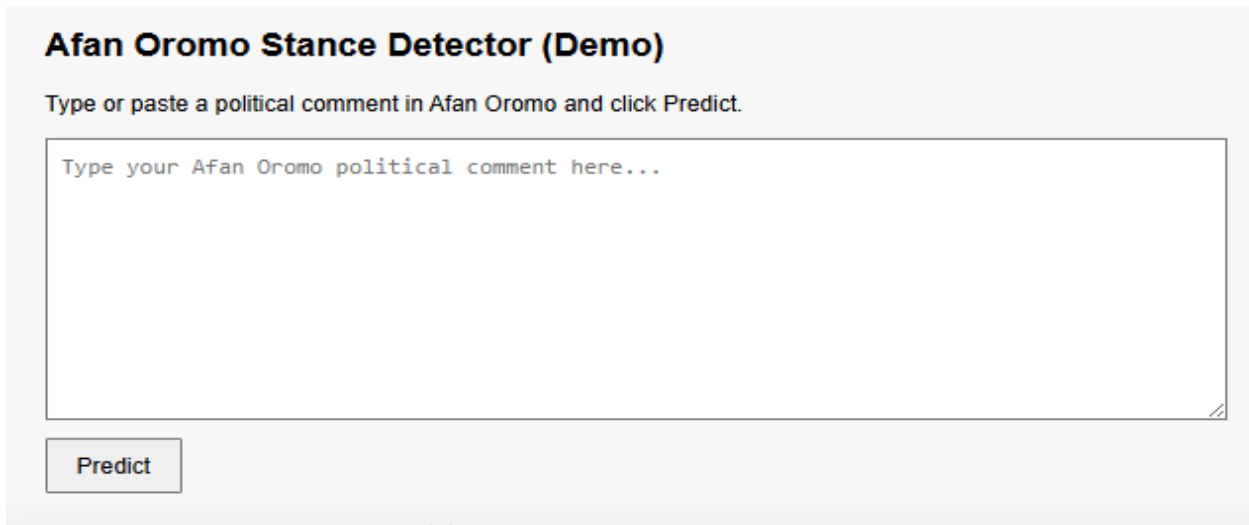
	precision	recall	f1-score	support
against	0.83	0.83	0.83	2500
supporter	0.83	0.83	0.83	2500
accuracy			0.83	5000
macro avg	0.83	0.83	0.83	5000
weighted avg	0.83	0.83	0.83	5000

Figure 12: Attention based RNN evaluation performance

The experimental performance outcome displayed in the previously mentioned figures 11 and 12 demonstrates the evaluation-measures of the attention-based RNN model scores of 0.83, 0.83, and 0.83 precision, recall, and F-score, respectively, and the attention-based LSTM scores of 0.94, 0.94, and 0.93. Accordingly, the outcomes demonstrate that the attention-based LSTM model achieved better than the attention-based RNN model.

5.7. The Model Prototyping

After the proposed model is developed, model evaluation is performed. The users can access the model to identify the political stances and check whether the written stance is “Against” or “Supporter”. The users can access and check Afaan Oromo political stances by typing or paste the text on provided text box by our model as following:



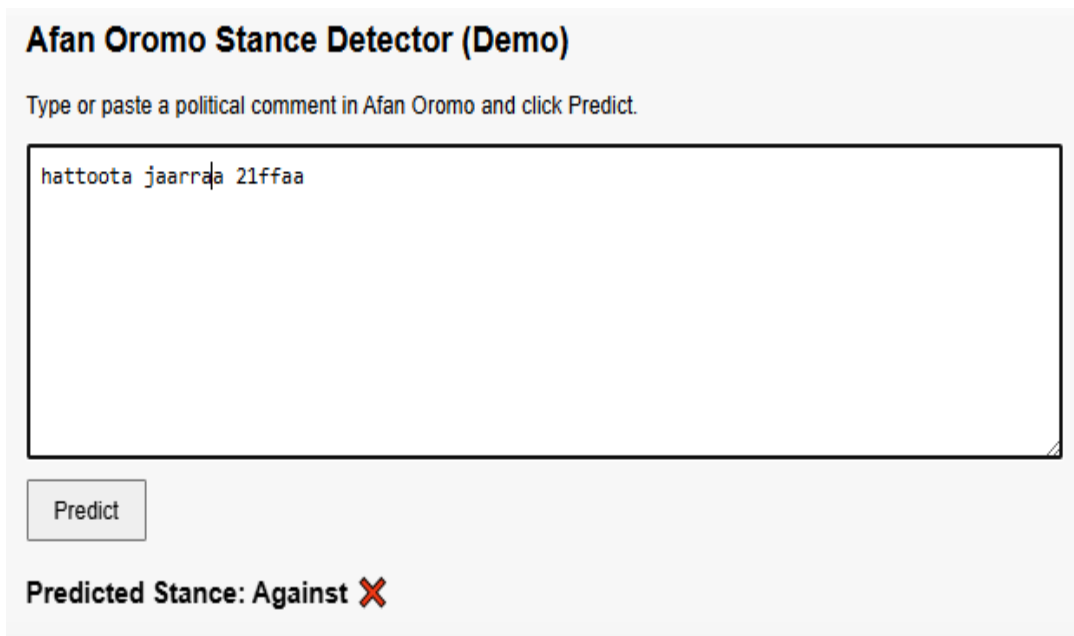
Afaan Oromo Stance Detector (Demo)

Type or paste a political comment in Afaan Oromo and click Predict.

Type your Afaan Oromo political comment here...

Predict

Figure 13: Graphical User interface design for our model



Afaan Oromo Stance Detector (Demo)

Type or paste a political comment in Afaan Oromo and click Predict.

hattoota jaarraa 21ffaa

Predict

Predicted Stance: Against ❌

Figure 14: Stance detected as 'Against' by our model

Afan Oromo Stance Detector (Demo)

Type or paste a political comment in Afan Oromo and click Predict.

jabaadhaa isin wajjiin jirraa.

Predict

Predicted Stance: Supporter 

Figure 15: Stance detected as 'Supporter' by our model

Afan Oromo Stance Detector (Demo)

Type or paste a political comment in Afan Oromo and click Predict.

oduu qofaan uummata kana caaltu

Predict

Predicted Stance: Against 

Afan Oromo Stance Detector (Demo)

Type or paste a political comment in Afan Oromo and click Predict.

Waan guddoo dansaati Jabaadha!

Predict

Predicted Stance: Supporter 

Figure 16: Sample of predicted result for both “Against and Supporter “by our models

CHAPTER SIX: CONCLUSION AND FUTURE WORK

6.1. Conclusion

One of the most frequent tasks for political scientists using text analysis is stance detection. Finding support or opposition to political candidates or social movements may be part of it. It might entail figuring out how people feel about particular racial, religious, or political groupings. We employed essential tools to develop the model, identify the optimal algorithms for our suggested system, acquire the necessary datasets, and properly preprocess them for testing before putting the model into practice. Because of its usually lengthy length, lack of formal structure, and presence of various representations that are only understandable by human readers, regular neural networks find it extremely challenging to process the increasing volume of news articles and text comments in Afaan Oromo on digital platforms. We made use of the benefits provided by an attention mechanism in an encoder-decoder to overcome the difficulties noted in the most recent research.

To assess and train the models, we used 10136 comment texts from the Facebook sites of the OLF, OFC, and Oromia Prosperity party political organizations. The prepared dataset was then used to train and evaluate the models. The second model, the Attention-based RNN model, has 95.82% training accuracy and 93.47% validation accuracy, while the Attention-based LSTM model scores 97.5% validation accuracy and 98.6% training accuracy. The performance of the system is highlighted and assessed using a series of experiments that were examined and deliberated in the former chapter. To ascertain the effectiveness of the proposed model and evaluate the range of its performance, a comprehensive experimental study is carried out utilizing a dataset comprising text, attitude, and label.

The effectiveness of the proposed method in attitude classification is evaluated through detailed evaluations. The validation dataset was used to evaluate the performance of the model using Precision, Recall, and F1-Score, which were based on the posture classification obtained from unseen data. The **precision**, **recall**, and **F-score** of the **attention-based RNN** model are **0.83**, **0.83**, and **0.83**, respectively, while the attention-based LSTM model has these same values at 0.94, 0.94, and 0.93. According to the findings, attention-based LSTM scoring has the uppermost training correctness of 98.6% and the highest validation accuracy of 97.5%. Furthermore, with precision, recall, and F-score of **0.94**, **0.94**, and **0.93**, correspondingly, the **attention-based LSTM** model has the highest F-measure score when compared to the attention-based RNN.

6.2 Contributions of the Study

The process of determining an individual or a people's attitude or point of view toward a specific subject, problem, or thing is known as "stance detection." It entails examining textual material to ascertain if the author is neutral, against, or in favor of a particular subject. Market research, political discourse analysis, and social media analysis frequently utilize stance detection to gauge sentiment and public opinion.10, 136, a useful but underutilized dataset, was used in this work to categorize texts written in Afaan Oromo.

The Attention-based LSTM and Attention-based RNN models are two examples of deep learning methods with attention mechanisms that were used to develop the model, which can solve the problem of manually collecting dataset features.

We employed the Word2Vec data structure to solve the issue of semantic similarity between words in a text.

In this study we contributed the followings:

- ✓ Our work focuses on applying stance detection to **Afaan Oromo texts**.
- ✓ We utilized **10,136 dataset** to classify these texts.
- ✓ To build our model, we used **Attention-based LSTM and Attention-based RNN deep learning methods**.
- ✓ These advanced models automate the process of collecting dataset features, which can solve the problem of manually efforts.
- ✓ we incorporated the **Word2Vec feature extraction** to effectively address and resolve issues of **semantic similarity** between words within the text

6.3. Recommendation

We would like to provide the following recommendations for future research based on the findings and understandings of the study:

- **Develop a Publicly Accessible Afaan Oromo Corpus:** The absence of a readily available Afaan Oromo corpus is a significant bottleneck. Creating this crucial resource will be instrumental for developing and evaluating more effective stance classification models.
- **Leveraging Larger Datasets and Advanced Embeddings:** Our current model was built using a limited dataset. Future researchers should utilize **larger datasets** and incorporate advanced **contextual embedding techniques** such as GPT-2/3, ELMO, and BERT to improve model performance.
- **Explore Multi-modal Stance Classification:** we solely used text datasets for classification in our study. Future studies should expand to include **audio, video, and image data** for a more comprehensive and robust **multi-modal** stance classification approach. Additionally, we didn't use idiomatic or emotional expression, in the future research to increase performance of the system researchers should add to it.

Reference

- Abafogi AA. (2021). Information Engineering and Electronic Business. International Journal of Information Engineering and Electronic Business. *International Journal of Information Engineering and Electronic Business*, 5(doi:10.5815/ijieeb.2021.05.05.), 51–59.
- Abdi, A., Gebresilassie, H., & Shiferaw, T. (2019,2020). Social Media and Political Communication in Ethiopia: A Study of Facebook Usage. The Role of Social Media in Political Participation: A Case Study of Ethiopian Youth. *e*.
- AlDayel, A. (2021). Stance characterization and detection on social media. *Unpublished Ph.D dissertation ,Institute for Language, Cognition and Computation*, 1-179.
- AlDayel, A. (2021). Stance characterization and detection on social media.
- Alemayehu O,FenetB. (2019). ReviewongenderedperspectiveofhouseholdsparticipationinagriculturalactivitiesinEthiopia. *JournalofAgriculturalExtension and Rural Development*. 11(1), 1-10.
- Alemayehu O,FenetB. (2019). ReviewongenderedperspectiveofhouseholdsparticipationinagriculturalactivitiesinEthiopia. *JournalofAgriculturalExtension and Rural Development*. 11(1), 11(1)(<https://dx.doi.org/10.5897/jaerd2018.0985>.), 1–10.
- Alessia. D, F. Ferri, P. Grifoni and T. Guzzo. (2015). Approaches, tools and applications for sentiment analysis implementation. *International Journal of Computer Applications*.
- Alzhrani, K. M. (2022). Political ideology detection of news articles using deep neural networks. *Intelligent Automation & Soft Computing*,, 33(1), 483-500.
- Arega, K. L. (2020). Social Media Fake Account Detection for Afan Oromo Language using Machine Learning. *New Media and Mass Communication ISSN 2224--3267 (Paper) ISSN 2224--3275 (Online)*, 90(2020).
- Asma Althagafi, Ghofran Althobaiti , Hosam Alhakami , Tahani Alsubait. (2021). Arabic Tweets Sentiment Analysis about Online Learning during COVID-19 in Saudi Arabia. (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, 12, 1-6.
- Asudani, Deepak Suresh, Naresh Kumar Nagwani, and Pradeep Singh. (2023). Impact of word embedding models on text analytics in deep learning environment: a review. *Artificial intelligence review*, 56(9), 10345-10425.
- Aynalem, K. (2022). SENTIMENT ANALYSIS ON TIGRAY TELEVISION SERVICES.

- B. Schiller, J. Daxenberger and I. Gurevych. (11, 2021). Stance Detection Benchmark:How Robust is Your Stance Detection? *Künstl Intell*, 3590, 329-341.
- B. Schiller, J. Daxenberger and I. Gurevych. (2021). Stance Detection Benchmark:How Robust is Your Stance Detection? *Künstl Intell*, 35(3,4), 329-341.
- Bedane, I. (2015). The origin of afan oromo: Mother language. *Glob. J. Hum. Soc. Sci. G Linguist. Educ*, 12.
- Beekan Erena. (2021). Oromo Language (Afaan Oromoo). (<https://scholar.harvard.edu/erena/oromo-language-afaan-oromoo>).
- Bencsik, B., Reményi, I., Szemenyei, M., & Botzheim, J. (2023). Designing an embedded feature selection algorithm for a drowsiness detector model based on electroencephalogram data. *Sensors*, 23(4), 1874.
- Bergam, N., Allaway, E., & McKeown, K. (2022). Legal and Political Stance Detection of SCOTUS Language. *arXiv preprint arXiv:2211.11724*.
- Brereton, R. G., and Lloyd, G. R. (2014). Support Vector Machines for classification and regression. (<https://doi.org/10.1039/b918972f>).
- Bryson, J. M. (2018). Strategic planning for public and nonprofit organizations: A guide to strengthening and sustaining organizational achievement.
- C. Liu, W. Li, B. Demarest, Y. Chen and S. Couture. (2016). An Ensemble Model for Stance Detection in Twitter. *in Proceedings of SemEval-2016, San Diego*.
- Can, K. .. (2021). "A survey". *Journal of ACM Computing Surveys*, 53, 1-37.
- Chowdhary, KR1442, and K. R. Chowdhary. (2020). Natural language processing. *Fundamentals of artificial intelligence*, 603-649.
- Debela, T. G. (2019). Afaan Oromo search engine. *Unpublished Master Thesis, Department of Computer Science, Addis Ababa University*.
- Dulume, W. (2022). Assessing Key Factors in the Making Afan Oromo One of the Federal Working Languages.
- E. Zotova, R. Aggerri and G. Rigau. (2019). Automatic stance detection on political discourse in Twitter. *Unpublished Masters Thesis, Department of Computer Systems and Languages, University of the Basque Country UP-V/EHU.*, 1-63.
- Edosomwan, S. O. (2011). The history of social media and its impact on business. *Article in The Journal of Applied Management & Entrepreneurship*, 16(3).
- Erena, B. (2021). Afaan Oromoo (Oromo language). (<https://scholar.harvard.edu/erena/oromo-language-afaan-oromoo>).
- Erena, B. (2021). Afan oromo(oromo language).

- G. Gautam and D. Yadav. (2014). "Sentiment analysis of twitter data using machine learning approaches and semantic analysis,"in 2014 Seventh international conference on contemporary computing (IC3).
- Ganti, Venkatesh, and Anish Das Sarma. (2022). Data Cleaning.
- Gasser, M. (2012). a system for morphological processing of Amharic, Oromo, and Tigrinya.
- getthematic. (05 May. 2022). Sentiment Analysis | Comprehensive Beginners Guide | Thematic | Thematic. (<https://getthematic.com/sentiment-analysis/>).
- Gondaliya, C, A. Patel and T. Shah. (2021). Sentiment analysis and prediction of Indian stock market amid Covid-19 pandemic . *IOP Conference Series: Materials Science and Engineering*.
- Grzegorzczuk, K. (2019). Vector representations of text data in deep learning. *arXiv preprint arXiv:1901.01695*.
- guru99. (2021). Retrieved from Difference Between Deep Learning and Machine Learning Vs AI: <https://www.guru99.com/machine-learning-vs-deep-learning.html> Machine learning. (2022). (https://en.wikipedia.org/wiki/Machine_learning).
- H. Elfardy and M. Diab. (2016). Ideological Stance Detection in Informal Text. *in Proceedings of SemEval-2016*.
- Ikome, L. B. (2019). Using Afaan Oromoo as primary language of learning and teaching in selected schools in Addis Ababa: Policy, problems and strategies.
- Jang, H E. Rempel, D. Roth, G. Carenini and N. Z. Janjua. (2021). Tracking COVID-19 discourse on twitter in North America: Infodemiology study using topic modeling and aspect-based sentiment analysis. *Journal of medical Internet research*.
- Jenq-Haur, Z. M. (2020). Sentiment Classification for Under-Resourced Language Using Word2Vec Neural Network Amharic Language Social Media Text.
- Josh Patterson , & Adam Gibson. (August 2017). *Deep Learning A PRACTITIONER'S APPROACH First Edition*. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.
- Jünger, J., & Keyling, T. (2019). Facepager. An application for generic data retrieval through APIs: Sourcecode and releases. *GitHub*. <https://github.com/strohne/Facepager>.
- Jünger, J, T. K. (2019). Facepager: Facepager was made for fetching public available data from YouTube, Twitter and other websites on the basis of APIs and webscraping.
- Kaplan, B., & Maxwell, J. A. (2005). Qualitative research methods for evaluating computer information systems. *Evaluating the organizational impact of healthcare information systems*, 30-55.
- Kapur, R. (2018). Research methodology: Methods and strategies. *Department of Adult Education and Continuing Extension, University of Delhi: New Delhi, India*.

- Kebede, F., & Kekeba, D. K. (2021). MACHINE LEARNING BASED SEMANTIC ROLE LABELING FOR AFAAN OROMOO TEXTS.
- Kebede, T. (2019). Machine Learning Based Multi-Scale Sentiment Analysis for Afaan Oromo Post.
- Kedia, A., & Rasu, M. (2020). Hands-on Python natural language processing: explore tools and techniques to analyze and process text with a view to building real-world NLP applications.
- Kende-Robb, M., & Salami, S. ((2019)). Digitalizing Africa: Technological Advances and Social Impact. World Bank Publications & Facebook user penetration in Africa as of 2023.
- Khan, K., Baharudin, B.B. and Khan. (2009). Mining opinion from text documents: A survey. *3rd IEEE International Conference on Digital Ecosystems and Technologies*, 217-222 .
- Küçük, D., & Can, F. (2022). A tutorial on stance detection. In D. & Küçük, *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (pp. 1626--1628).
- Küçük, D., & Can, F. (2020). Stance detection:. A survey. *ACM Computing Surveys (CSUR)*, 53(1), 1-37.
- Küçük, D., & Can, F. (2020). Stance detection: A survey. *ACM Computing Surveys (CSUR)*, 53(1), 1-37.
- L. Barbosa and J. Feng. (2010). Robust sentiment detection on twitter from biased and noisy data. *Coling 2010: Posters*, 36-44.
- Lee, L, Pang, B., and Vaithyanathan, S. (2014). Sentiment Classification using Machine Learning Techniques. *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics*.
- N. Alturayef, H. Luqman and M. Ahmed. (2016). A systematic review of machine learning techniques for stance detection and its applications. *in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*.
- Olbasa, C. W. (2018). Choice for a working language in Ethiopia: a case study among graduating classes of Oromo speakers in selected public universities. *Macrolinguistics*, 6(9), 98--115.
- Oshikawa, R., Qian, J., & Wang, W. Y. . (2018). A survey on natural language processing for fake news detection. *arXiv preprint arXiv:1811.00770*.
- Padmaja and Fatima. (2013). Opinion mining and sentiment analysis-an assessment of peoples' belief: A survey. *International Journal of Ad hoc, Sensor & Ubiquitous Computing*.
- Papalambros, P. Y. (2015). Design science: why, what and how. *Design Science*, 1, e1.
- Patel and Thakkar. (2015). Approaches for sentiment analysis on twitter: A state-of-art study.
- Pk, K. (2017). Insights to Problems, Research Trend and Progress in. (<https://www.researchgate.net>).
- Pokharel, B. P. (2020). Twitter Sentiment analysis during COVID-19 Outbreak in Nepal. 1-26.

- Rai, Abigail, and Samarjeet Borah. (2021). Study of various methods for tokenization. In *Applications of Internet of Things: Proceedings of ICCCIOT 2020* (pp. 193-200). Springer.
- Ray, S. (2017). "6 Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R". *Analyticsvidhya*(<https://analyticsvidhya.com/blog/2017/09/naive-bayes-explained/>).
- Roy, S. (2020). ECONOMIC_IMPACT_OF_COVID-19_PANDEMIC. 1-20.
- Sadiku, M. N., Musa, S. M., & Chukwu, U. C. (2022). Artificial intelligence in education. *iUniverse*.
- Siau, K., & Rossi, M. (2011). Evaluation techniques for systems analysis and design modelling methods--a review and comparative analysis. *Information Systems Journal*, 21(3), 249--268.
- Singh, Jasmeet and Gupta, Vishal. (2016). Text stemming: Approaches, applications, and challenges. *ACM Computing Surveys (CSUR)*, 49(3), 1-46.
- Tadesse, S. (2023). POLITICAL STANCE DETECTION ON AMHARIC TEXT USING MACHINE LEARNING.
- Tarkar, P. (2020). Impact Of Covid-19 Pandemic On Education System. *International Journal of Advanced Science and Technology*, 29, 3812-3814.
- Techo, V. (2016). , "ResearchGate | Find and share research". , *Researchgate*(<https://researchgate.net/>).
- Teixeira, J. G., Patrício, L., & Tuunanen, T. (2019). Advancing service design research with design science research. *Journal of Service Management*, 30(5), 577-592.
- Tekle, E. (2022). Sentiment Analysis on Amharic Language-Based COVID-19 Discourse from Facebook social media comments.
- Venable, J., Pries-Heje, J., & Baskerville, R. (2016). FEDS: a framework for evaluation in design science research. *European journal of information systems*, 25, 77-89.
- Vom Brocke, Jan, Alan Hevner, and Alexander Maedche. (2020). Introduction to design science research. *Design science research. Cases*, 1-13.
- Vyrva, N. (2016). Sentiment analysis in social media.
- W. Ferreira and A. Vlachos. (2016). Emergent: a novel data-set for stance classification. in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*.
- W. Medhat, A. Hassan and H. Korashy. (2014). Sentiment analysis algorithms and applications. *A survey Ain Shams engineering journal*, 5, 1093-1113.
- Waykole. R. N and A. D. Thakar. (2018). "A review of feature extraction methods for text classification,". *Int. J. Adv. Eng. Res. Dev*, 5, 351-354.

Zheng, Xiaochen, Meiqing Wang, and Joaquín Ordieres-Meré. (2018). Comparison of data preprocessing approaches for applying deep learning to human activity recognition in the context of industry 4.0}. *Sensors*, 18(7), 2146.

Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., & Procter, R. (n.d.).

Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., & Procter, R. (2018). Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)*, 51(2), 1-36.

Appendixes

Appendix A: Sample of output by our models

```
# Example Prediction
print(predict_stance("jabaadhaa"))
```

1/1 ————— 0s 40ms/step
supporter

```
# Example Prediction
print(predict_stance("jijjiiramni kee namuma boossisuudha"))
```

1/1 ————— 0s 39ms/step
against